SQL Injection Attack Prevention Based on Decision Tree Classification

B.Hanmanthu
Assistant Professor, Dept. of CSE
Kakatiya Institute of Technology & Science, Warangal, Telangana, India.
bh@cse.kitsw.ac.in

B.Raghu Ram
Assistant Professor, Dept. of CSE
Kakatiya Institute of Technology & Science, Warangal, Telangana, India.
brg@cse.kitsw.ac.in

Dr.P.Niranjan
Professor, Dept.of CSE
Kakatiya Institute of Technology & Science, Warangal, Telangana, India.
pnr@cse.kitsw.ac.in

***Abstract*— In real world as dependence on World Wide Web applications increasing day by day they transformed vulnerable to security attacks. Out of all the different attacks the SQL Injection Attacks are the most common. In this paper we propose SQL injection vulnerability prevention by decision tree classification technique. The proposed model make use famous decision tree classification model to prevent the SQL injection attacks. The proposed model will filter the sent HTTP request by using a decision tree classification based attack signatures. We test our proposed model on synthetic data which given satisfactory results.**

***Keywords*— *SQL Injection Attack, Web Security, Decision Tree, Data Mining*.**

I. INTRODUCTION

In the present scenario Web applications such as websites, face book, twitter, mail servers, and financial applications have become part of our life. Due to this effectiveness, web applications have made them a most significant objective for web attackers. Web applications are known to a number of vulnerabilities which can be due to an improper design or a ruthless implementation. Among the top ten web application vulnerabilities published by Open Web Application Security Project [1], SQL Injection Attack is the most vulnerable. According to OWASP, SQL injection vulnerabilities were reported in 2008, making up 25% of all reported vulnerabilities for web applications.

An SQL Injection Attack occurs when an unlawful user access to database by altering the planned effect of an SQL query by inserting new SQL keywords or operators into the query by which he acquires the informal access to a database in order to view or manipulate data base. The origin of SQL Injection Attack is inappropriate user input validation. Although there is a great awareness about security, there are number of important reasons that make securing web applications complicated. At one end web applications are increasing day by day at a extraordinary speed and largely fuelled by the unfussiness with which even a common people also can develop such applications due to the easiest tools available in the market. At another end the developers and administrators do not have the adequate information and experience in the area of security in SQL Injection Attack.

A practical approach to handle the problem of SQL Injection Attack is to scan the each query coming or inputted to the web database and allow the queries only the proper SQL queries to access the databases. In spite of their massive number of proposals, these existing SQL Injection Attack prevention applications still have some problem with regard to the high number of undetected vulnerabilities and high percentage of false positives. The detection of Web SQL injection attack is not a great application but it is a useful tool to access the security of web applications. The technique proposed in this paper is to first to use decision tree classification algorithm to get common rules in different SQL attacks and use these rules for defend the other attacks. According to the proposed model first it will scans a webpage in a controlled environment and discover the vulnerabilities using a decision tree classification rules. Next we provide a framework to detect and defeat SQL Injection Attack using decision trees.

A decision tree is a data mining technique which uses a tree data structure for decisions and their possible cost, including chance event outcomes, resource costs, and utility. It is one way to show an algorithm. For decision analysis the Decision trees are commonly used in the data mining is regularly, to help identify a strategy most likely to reach a target. A decision tree is a graph and tree combinational data structure in which internal node represents a preparation on an attribute each branch represents the outcome of the test and each leaf node represents a class label. The paths from origin to leaf represent classification rules.

Where the predictable values of challenging alternatives are calculated the decision trees are used for as a visual and analytical mining tool, Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach an objective. If in practice decisions have to be taken World Wide Web with no recall under incomplete knowledge, a decision tree should be paralleled by a statistic model as a best choice model or online selection model algorithm. Another use of decision trees is as a descriptive means for calculating conditional statistics.

The decision tree can be converted into decision rules, where the outcome is the results of the leaf node, and the conditions along the path form an intersection in the decision

978-1-4799-6480-2/15/$31.00 © 2015 IEEE

clause. Decision rules can also be calculated by building associative classification rules.

By considering the importance of the SQL Injection Attack s and the decision trees we propose a SQL Injection Attack prevention model based on decision tree classification technique. Remaining paper is organized by section 2 with related work, section 3 with proposed model, section 4 with evaluation procedure and finally paper end with conclusion and references.

II. Related Work

A successful SQL injection attack is one which interfere privacy, Integrity and availability of the information in the database. Based on the statistical researches this type of attack had a high impact on the business. To stop or mitigate the SQL injection attacks is required to discover an appropriate solution [2]. In this paper [3], a look at different approaches to noticed and defend against SQL Injection Attacks, each with their strengths and weaknesses, and then proposed a combined approach of SQLIA prevention techniques (the fine grained Role Based Access Control [RBAC] and static and dynamic analysis of SQL parse trees) in order to maximize the advantages of each method and to ensure that a second line of protection is provided, if in case of the first method fails. The mechanism of SQL injection attack is introduced in the paper [4]. Differing from the works of the predecessors, the authors categorize the injection attacks according to the characteristics of the injection codes. For the type of web databases with SQL Server as the backend, a DDL (Detection-Defense-Log) Model against SQL injection is fashioned. For both the client's computer and the server are included in the model. In this paper [5], we propose a novel technique to defend against the attacks targeted at stored procedures. This technique combines static application code analysis with runtime validation to eliminate the occurrence of such attacks. In the static part, we design a stored procedure parser, and for any SQL statement which depends on user inputs, we use this parser to instrument the necessary statements in order to compare the original SQL statement structure to that including user inputs. In this paper [6] they have proposed a novel query transformation scheme and hashing technique. This technique is a lightweight approach to prevent SQL Injection attacks. We implemented it on a prototype e-commerce application and the results of our experiments show that it can successfully and efficiently block a variety of SQL Injection attempts. This approach can also be easily implemented on any language or database platform with little modification. Big Web applications have hundreds of places where users can input data, each of which can provide a SQL injection opportunity. Attacker can steal confidential data of the organization with these attacks resulting loss of market value of the organization. This paper [7] The SQL injection attack discovery and avoidance techniques are presented in an effective survey of SQL Injection attack. A successful SQL injection attack is one which reveals dangerous secret information to the attacker. In this paper [8] firstly we provided the background information on the vulnerability. A comprehensive review of different types of SQL injection attack. For every attack we had given an example that shows how the attack is launches. Lastly we propose the best solution at development phase to defeat SQL injection and conclusion.

Diagnostic Decision Trees [9] are one which built based on the fault trees as static trees that can serve as the fundamental diagnostic trees, and the dynamic DDTs are built over time from vehicle telemetry data. The dynamic DDT will add the functionalities of guessing, and it will be able to detect the unidentified faults. Crew or maintenance engineers can use the decision tree system without having previous information or experience about the diagnosed system. To show through mapping and ISS examples that the approach is feasible and successful. We also present future work and development. An improved random decision trees algorithm [10] with application to land cover remote sensing classification was proposed in this paper. Firstly, in accordance with the low operation competence of random decision trees algorithm, an improved random decision trees algorithm was presented by adding tree balance factor, setting node contamination and distinguishing sample types. Decision trees have been widely and successfully used in machine learning. However, they have suffered from over fitting in noisy domains. This problem has been remedied, in C4.5 for example, by tree pruning, resulting in better performance. More recently, decision trees have been combined with fuzzy representations [11]. To generate a decision tree the RBDT-1 method [12] uses a set of declarative rules as an input. The objective of method is to create on-demand a short and accurate decision tree from a stable or dynamically changing set of rules.

For credit card fraud detection and biometrics related applications the Data mining can also being applied [13]. While some development has been made on topics such as stream data mining, there is still a lot of work to be done here. To mine multimedia data including surveillance video is another challenge. Finally, we need to maintain the privacy of individuals. Firewalls [14] are used as a security check point in a network environment; nevertheless there are still various types of security issues which are on the rise. In order to strengthen the network from illegal access the concept of IDS (Intrusion Detection System) is gaining popularity around the world. The constantly growing scale and enriching genres of network data always demand higher levels of efficiency, robustness and generalizability where existing approaches with successes on small, homogeneous network data are likely to fall short. We introduce MultiAspectForensics, [15] a handy tool to automatically detect and visualize novel sub graph patterns within a local community of nodes in a heterogeneous network, such as a set of vertices that form a dense bipartite graph whose edges share exactly the same set of attributes. Instrumenting components such as data transformations, model deployment, and cooperative distributed detection remain a labor intensive and complex engineering Endeavour. This paper [16] describes a database centric architecture that leverages data mining with .NET to address these challenges. It also offers numerous advantages in terms of alert infrastructure, security, scalability, reliability and also has data analysis tools. With a Data mining Based Intrusion detection system application prototype using .NET the database centric architecture is illustrated.

III. PROPOSED MODEL

The proposed model work by sending different specially planed attack request, the proposed SQL injection decision tree model where the vulnerabilities chances are get and the final SQL injection database will be created for using as classification data. For checking and finding the SQL injection we have defined a database that database is stored all the database error related to different injection attack. The proposed model uses the satisfied analysis technique for finding the SQL injection attack which makes use of the SQL decision tree. We create the attack appeal in such a way that, if the web pages are not disinfected the query URL will always provide database problem. After putting the attack request our tool robotically checks the response if there exist any database problem then we can inform to the end user if there is no problem then the end user can access the database without any problem. This database attack is stated in classification rules which can be updated further by the administrator if it is needed. If any database attack is found in the web server of the response page then we can say that vulnerability exists in the given database query. We are also sanitizing the single input points with several injection attacks and then we assure that this point is goodness.

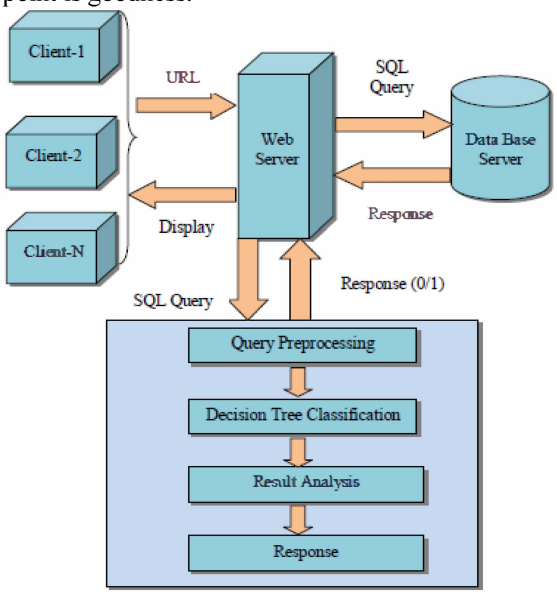

Fig.1: Three tier architecture for defending SQL Injection attack using Decision trees.

As per the architecture proposed in this paper the client machines can be n-number and any client can access the web server and he can get the services of the web server. In order to get that service he will be using URL request using post and get methods. After user posting the query if the information is require from the database the web server will make use the database server where the query transformation can be occur.

Software engineers mostly rely on dynamic query building with string concatenation to construct SQL statements. In real time applications at the runtime, the system forms queries with inputs directly received from external sources. The proposed method makes it possible to engineer different queries based on varying conditions set by users. However, as this is the cause of many web applications, some developers opt to use parameterized queries or stored procedures. While these methods are more secure, their inappropriate use can still result in vulnerable code. In HTML code there is lack of checks for SQL attacks which make the system more attractive to attacks. The most common and serious mistake developers make is using inputs in SQL statements without any checks.

Mostly manual defensive coding practices are the best way to defeat SQL injection, such application is need more manual interactions and error-prone. To alleviate these problems, the method proposes input validation automation system which will be eradicating the any kind of user interaction with the system form validation of input. In our proposed model the engineers can optional to provide their own database schema and construct SQL statements using its APIs. The proposed model is especially useful when developers need to use dynamic queries instead of parameterized queries for greater flexibility. In most cases they can only use it with new software projects, and they must learn a new query development process.

In the proposed model the client URL request to server will be filtered using decision tree. Which will be classifies the SQL query as attack class and non-attack class based up on previous information. According to our model first the data regarding the SQL injection attack will be acquired from the sources and using the decision tree classification algorithm for SQL injection classification (Algo.A) the class rules for classifying the SQL queries as attacks and non-attacks will be constructed.

When an request come to Web server the server will send the request to SQL injection attack detection algorithm (Algo.B) which make use the Decision tree Analyzer (Algo.C) and find the whether the request safer or not if the request is safe then it will send it to data base are it send failure message to user.

A. Aalgorithm for Decision tree construction on SQL Injection Database

Step1: Check the SQL injection database
Step2: For each SQL query I
Step3: Find the information gain ratio
Step4: Based up on information gain
Step5: Create a decision node that splits with information gain
Step6: Based up on nodes construct the tree for classifying SQL queries as attack and non attacks.

B. Aalgorithm for SQL Injection attack detection procedure

Step1: Create a Queue Data Structure to store incoming URL requests
Step2: Decide whether given URL is to access database or not
Step3: If the URL is to target query then send it to Decision tree Analyzer and record response.
Step4: If the response is positive direct to database.
Step5: Else report failure message to user.
Step6: Finish

C. Aalgorithm for Decision tree Analyzer

Step1: Get the SQL request from web server
Step2: Map to the SQL injection Decision tree and find the class of SQL query
Step3: if the resulted class is attack class give the negative result
Step4: else give the positive response

IV. EVALUATION

The experiments conducted using P4-2.0Ghz processor and 512MB RAM based Pc's and we used synthesized SQL injection attack data. To provide heterogeneous environment the collected data is stored on different pc's with different type of data bases. There are several commonly used evaluation metrics for evaluating SQL injection drifting algorithms. They include accuracy measure precision and recall, and the percentage of recommendable items in the system known as coverage. The choice of these performance metrics is dependent upon the recommendation strategies used.

Experiment conducted in a fold cross validation, and the results of all trials were averaged to obtain the result. According to model first using the synthesis data set decision tree form SQL attack classification generated the this tree used with false attack and good cases and accuracy measured where the experiment shown above the 82% accuracy shown which prove the greatness of the proposed model.

The various attack types that we used for evaluation models include Tautology-based attack, Illegal or incorrect queries,

UNION command attack, Piggy bag attack, Blind SQL attack and Timing attack. In performance evaluation the model showed the consistence in attack detection and elimination. In experimentation the model showed consistency performance at average of 82% at all type of attacks. The continuous consistency appraisal of proposed model shown in Fig.2.

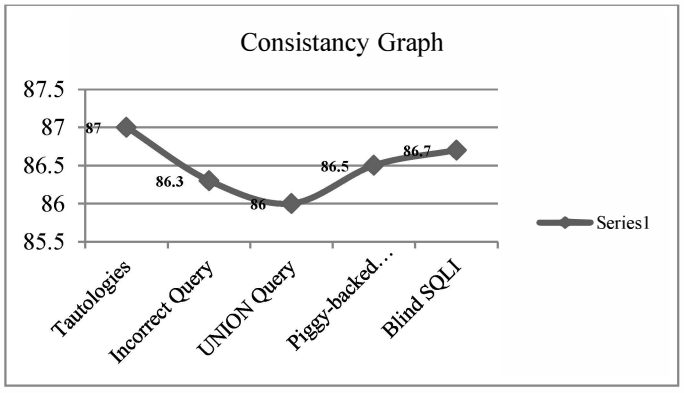

Fig.2: Consistency graph for different evaluation model

In order to do perform comparative evaluation of the proposed model we compare our proposed model to the other SQL scanning model which include Acuneits, Netsparker, Web cruiser and results of the proposed model shown good accuracy in compare to other models. The comparison done with respect to detection of vulnerability as well as false positive comparison out of all comparative models the proposed model shown effective performance. It shown better than web cruiser and equality to Netsparker and it slightly degraded to Acuneits model. Out of 25 attacks tested the proposed model able to detect 20 model and 5 models are considered as false positives at an average we can conclude that our proposed model is well appreciated with respect to accuracy in compare to the other popular models.

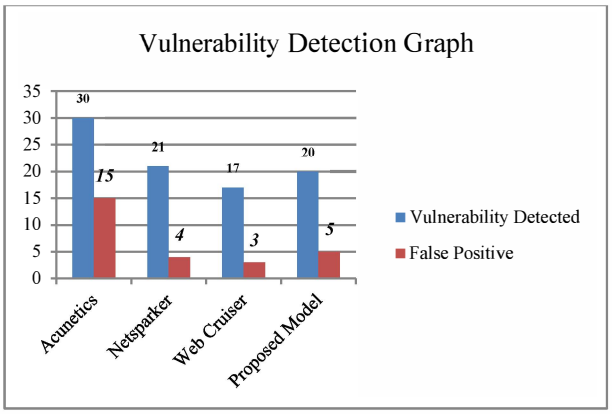

Fig.3: accuracy comparison of proposed model

In order to do evaluate the proposed model with respect to time of accuracy models we compare our proposed model to the other SQL scanning model which include Acuneits, Netsparker, Web cruiser and results of the proposed model shown greater accuracy in compare to other models. The model shown 1.48 sec of time to detect 25 attacks which is better than Acunetics but slower than Netsparker and WebCruiser so we can claim our model is optimized with respect to time and accuracy. The time taken of various model is shown in Table.1.

TABLE.1: TIME OF EVALUATION FOR DIFFERENT MODELS

Model Name	Acunetics	Netsparker	Web Cruiser	Proposed Model
Time in Sec.	2.24	1.2	0.15	1.48

V. CONCLUSION

In this paper we proposed SQL injection vulnerability prevention by the decision tree classification technique. The proposed model makes use famous decision tree classification model to prevent the SQL injection attacks. The proposed model will filter the HTTP request send by client using a decision tree based classified attack signatures. We tested our proposed model on synthetic data which given satisfactory results.

REFERENCES

[1] OWASP (Open Web Application Security Project) https://www.owasp.org/index.php/Category: OWASP_Top_Ten_Project visited on January 2011.

[2] Sadeghian, A; Zamani, M.; Manaf, AA, "A Taxonomy of SQL Injection Detection and Prevention Techniques," Informatics and Creative Multimedia (ICICM), 2013 International Conference on, vol., no., pp.53,56, 4-6 Sept. 2013 doi: 10.1109/ICICM.2013.18

[3] Dogbe, E.; Millham, R.; Singh, P., "A combined approach to prevent SQL Injection Attacks," Science and Information Conference (SAI), 2013 , vol., no., pp.406,410, 7-9 Oct. 2013

[4] Qian Xue; Peng He, "On Defense and Detection of SQL SERVER Injection Attack," Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on , vol., no., pp.1,4, 23-25 Sept. 2011 doi: 10.1109/wicom.2011.6040534

[5] Ke Wei; Muthuprasanna, M.; Kothari, S., "Preventing SQL injection attacks in stored procedures," Software Engineering Conference, 2006. Australian , vol., no., pp.8 pp.,, 18-21 April 2006 doi: 10.1109/ASWEC.2006.40

[6] Kar, D.; Panigrahi, S., "Prevention of SQL Injection attack using query transformation and hashing," Advance Computing Conference (IACC), 2013 IEEE 3rd International , vol., no., pp.1317,1323, 22-23 Feb. 2013 doi: 10.1109/IAdCC.2013.6514419

[7] Kumar, P.; Pateriya, R.K., "A survey on SQL injection attacks, detection and prevention techniques," Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on , vol., no., pp.1,5, 26-28 July 2012 doi: 10.1109/ICCCNT.2012.6396096

[8] Sadeghian, A; Zamani, M.; Abdullah, S.M., "A Taxonomy of SQL Injection Attacks," Informatics and Creative Multimedia (ICICM), 2013 International Conference on , vol., no., pp.269,273, 4-6 Sept. 2013 doi: 10.1109/ICICM.2013.53

[9] Lee, C.; Alena, R.L.; Robinson, P., "Migrating Fault Trees To Decision Trees For Real Time Fault Detection On International Space Station," Aerospace Conference, 2005 IEEE , vol., no., pp.1,6, 5-12 March 2005 doi: 10.1109/AERO.2005.1559584

[10] Haiwei Xu; Minhua Yang; Liang Liang, "An improved random decision trees algorithm with application to land cover classification," Geoinformatics, 2010 18th International Conference on , vol., no., pp.1,4, 18-20 June 2010 doi: 10.1109/GEOINFORMATICS.2010.5567531

[11] Benbrahim, H.; Bensaid, A, "A comparative study of pruned decision trees and fuzzy decision trees," Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American, vol., no., pp.227,231, 2000 doi: 10.1109/NAFIPS.2000.877426

[12] Abdelhalim, A; Traore, I, "A New Method for Learning Decision Trees from Rules," Machine Learning and Applications, 2009. ICMLA '09. International Conference on , vol., no., pp.693,698, 13-15 Dec. 2009 doi: 10.1109/ICMLA.2009.25

[13] Thuraisingham, Bhavani, "Data Mining for Malicious Code Detection and Security Applications," Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on , vol.2, no., pp.6,7, 15-18 Sept. 2009 doi: 10.1109/WI-IAT.2009.379

[14] Shwetha Nayak, B., "Research on application of intrusion detection system in data mining," Research & Technology in the Coming Decades (CRT 2013), National Conference on Challenges in , vol., no., pp.1,8, 27-28 Sept. 2013 doi: 10.1049/cp.2013.2486

[15] Maruhashi, K.; Fan Guo; Faloutsos, C., "MultiAspectForensics: Pattern Mining on Large-Scale Heterogeneous Networks with Tensor Analysis," Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on , vol., no., pp.203,210, 25-27 July 2011 doi: 10.1109/ASONAM.2011.80

[16] Chetan, R.; Ashoka, D.V., "Data mining based network intrusion detection system: A database centric approach," Computer Communication and Informatics (ICCCI), 2012 International Conference on , vol., no., pp.1,6, 10-12 Jan. 2012 doi: 10.1109/ICCCI.2012.6158816