Volume 2, Issue 1, January 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

URLAD (URL attack detection) - using SVM

Romil Rawat
Depart. of IT,SATI

Megha Zodape
Department of IT,SGSITS

Praveen kataria
Department of SS,SATI

chandrapal singh dangi
Deparrment of CSE,SSSIT

Abstract— **As the technological view came in existence, human prospectus is greatly enhanced and completely made dependent on it, internet technology is great and beneficial approach for human, internet has various functionalities as banking ,railway reservation etc, various secure algorithms has been designed for protecting web applications, certain approach keeps updating as needs required, but as the security increased ,vulnerabilities also increased for getting illegal accessing in secure prohibited area.web application has various input functions for entering user inputs, which might be vulnerable, malicious web-sites ,which are difficult to predict by only accessing, they could only be predicted by their content, signature, pattern and procedure. Website phishing, disclosure of personal information and data theft. it would seems difficult to check malicious URL from existing algorithms, In this paper, technique of SVM is used for classification ,detection and prediction of blacklisted URL's. The proposed algorithms provide accuracy of 96.97% and which is the best among the existing approaches.**

Keywords— **SQL Injection, Database Security, Authentication, blacklisted, SVM.**

I. INTRODUCTION

This Various techniques are available for detecting and eliminating suspicious and blacklisted URL's. Algorithm and tool works by detecting and checking attack signatures and attack procedure and patterns. They provide bypassing mechanism to access secure connections and designs. every attack have their predefined pattern but they grows as the security implication increases, as security increases ,attack potential also increases. Every web-browser has their own deign pattern and algorithms like internet explorer, Netscape navigator, Mozilla Firefox, Opera, googlecrome, some them works on HTTP and some them works on HTTPS. https provides secure channel for web application and are less susceptible to attack but http is susceptible to attack.

Ex: - www.staff.com/info.php/id/123/address (**Trusted URL**)

User input is supplied through web application interface, which then further executed through available modules or codes of databases. If proper input validation, syntax validation, secure coding framework, secure guideline for web designing and, URL verification mode is not followed malicious code could be injected in database.

Ex:- www.staff.com/@#$%^&*+?:{}|<>/id (**Blacklisted URL**)

The above example shows the URL based attack signatures,

A. Web application Interaction is as follows

-Web application is requested through a web browser by a user.
-The HTTP or HTTPs protocol accepts a request of user and sent to the targeted web server.
-Request received is executed by Server.
-Output is generated by Application program and sent back to the user via HTTP.
-Cookies maintain Current states of User, Web server and their execution report.

B. SVM(Support Vector Machine)

The term SVM[13] is typically used to describe classification with support vector methods and support vector regression is used to describe regression with support vector methods. SVM (Support Vector Machine) is a useful technique for data classification.

The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, i.e. it generalizes well. Consider the example in figure 1. Here there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin (maximizes the distance between it and the nearest data point of each class). These linear classifiers termed the optimal separating hyper plane. Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries.

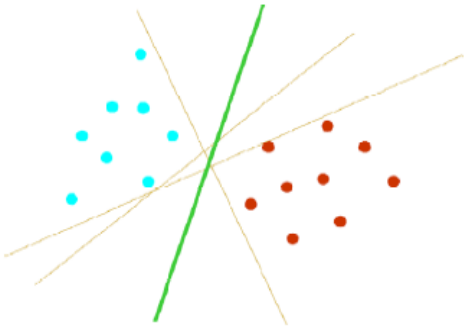


Fig. 1 Optimal Separating Hyper Plane

A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one "target value" (class labels) and several "attributes" (features). The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes.

In this proposed work linear kernel function is used. Which is shown below:

$$\text{Linear: } K(x_i, x_j) = x_i^T x_j$$

Other kernel functions are Polynomial, RBF (Gaussian kernel), sigmoid function

The RBF kernel nonlinearly maps samples into a higher dimensional space, so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF show that the linear kernel with a penalty parameter C has the same performance as the RBF kernel with some parameters (*C, r*). In addition, the sigmoid kernel behaves like RBF for certain parameters.

In proposed work a unique concept of determining Blacklisted URL's is presented using SVM. Which classifies Trusted URL and Blacklisted URL's?

II. RELATED WORK

The mechanism to keep track of the positive taints and negative taints is proposed by William G.J. Halfond, Alessandro Orso, PanagiotisManolios [10],Defensive Programming [11][12] has given a approach for Programmers by which they can implement their own input filters or use existing safe API s that prevent malicious input or that convert malicious input in to safer input. Vulnerability pattern approach is used by Livshits and Lam [8], they propose static analysis approach for finding the SQL injection attack. . The main issues of this method, is that it cannot detect the SQL injection attacks patterns that are not known beforehand. Vulnerability patterns are described here in this approach.

AMNESIA mechanism to prevent SQL injection at run time is proposed by William G.J. Halfond and Alessandro Orso [9].It uses a model based approach to detect illegal queries before it sends for execution to database.Static analysis framework (called SAFELI) has been proposed by Xiang Fu et al [5], for identifying SIA (SQL Injection attacks) vulnerabilities at compile time.. the source code information can be analyzed by SAFELI and will be able to identify very delicate vulnerabilities that cannot be discovered by black-box vulnerability scanners.Scott and Sharp Proxy filter [1] [2] , this technique can be effective against SQLIA; they used a proxy to filter input data and output data streams for a web application ,although correctly specify filtering rules for each application is required by the developers to input. The mechanism which filters the SQL Injection in a static manner is proposed by Buehrer et al [7]. The SQL statements by comparing the parse tree of a SQL statement before and after input and only allowing to SQL statements to execute if the parse trees match. Novel-specification based methodology proposed by Konstantinos et al [6], they given a mechanism to detect SQL injection with. This approach utilizes specifications that define the intended syntactic structure of SQL queries that are produced and executed by the web-application.

Instruction–Set Randomization [1][3] defined a framework that allows developers to create SQLqueries using randomized keywords instead of the normal SQL keywords.A study of drive-by exploit URLs had been performed by Provos et al. perform and they use a patented machine learning algorithm as a pre-filter for VM-based analysis [14]. They extract content-based features from the page, including whether IFrames are "out of place,"the presence of obfuscated javascript, and whether IFrames point to known exploit sites.

.

III. PROPOSE TECHNIQUE

All Applied techniques contains special and unique feature and concept for detecting blacklisted(malicious)URL, which provides bypassing mechanism to attacker, here a dataset is created ,which contains lists of blacklisted URL's and trusted URL's. our unique feature detects and blocks blacklisted URL's for securing application.

Ex-
Trusted URL= www.staff.com/info.php/id/123/address

Blacklisted URL= www.staff.com/@#$%^&*+?:{}|<>/id

Here, blacklisted URL are detected and blocked and trusted URL are passed for processing.

Dataset contains labels for detecting and blocking blacklisted URL.

'B'-Blacklisted URL
'T'-Trusted URL

Ex- ('T') www.staff.com/info.php/id/123/address
('B') www.staff.com/*.info/id/#$@/address

('B') www.staff.com/@#$%^&*+?:{}|<>/id

The above showed URL contains different combinations for accessing application by an authorized manner. Here only Trusted URL is passed by checking and detecting safe combinations.

A. *Propose Algorithm*

Step 1. Enter URL in the checking textbox.

Step 2. Train the system from provided datasets of URL(using SVM).

(i) URL 's marks as B, shows blacklisted URL.

(ii) URL 's marks as T, shows trusted URL.

Step 3.Predict the attack.

(i) Classify the attack using labels B(blacklisted) and T(trusted).

Step 4.Calculate performance and efficiency of system using labels (B and T).

Step 5.Repeat steps 1 to 3 till the correct classification precision is achieved.

IV. TESTING AND RESULTS

The applied approach has been tested on a URL's dataset. For testing dummy dataset has been created. The dataset has been populated with the records of blacklisted URL's (B) and Trusted URL's (T) and was tested, whose results are shown in fig 2

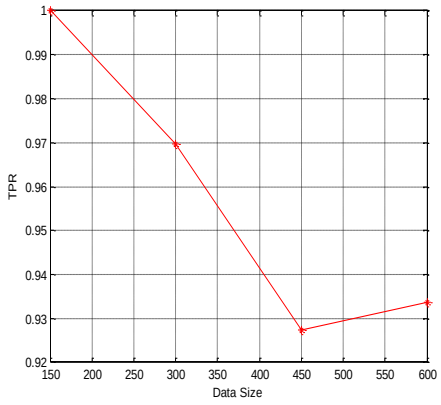
Detection time (in seconds) is calculated by taking average of 100 blacklisted URL's (B) and Trusted URL's (T).

TABLE I

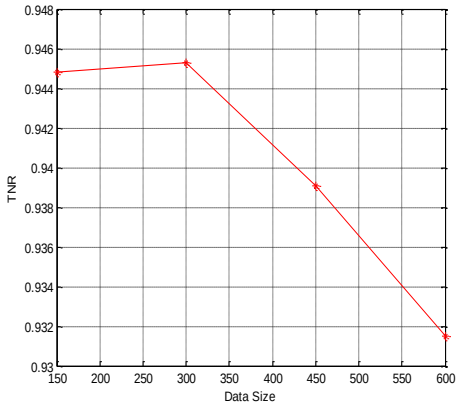
Trusted URL's	0.01298
Blacklisted URL's	0.01302

Fig. 2 Detection Time

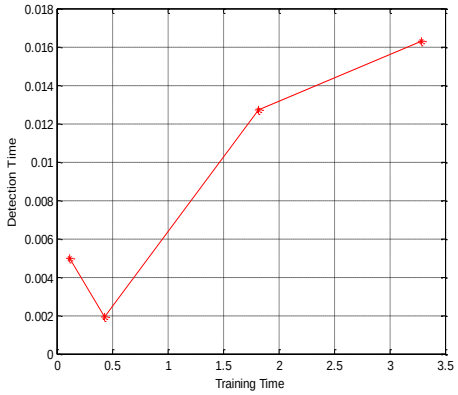
(TPR)True positive rate is calculated as the data size changes. As shown in fig 3



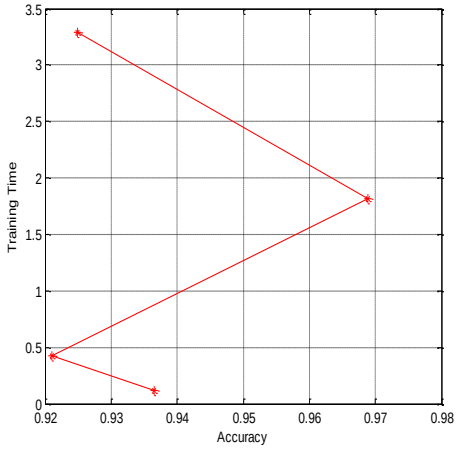
(TNR)True negative rate is calculated as the data size changes. As shown in fig 4



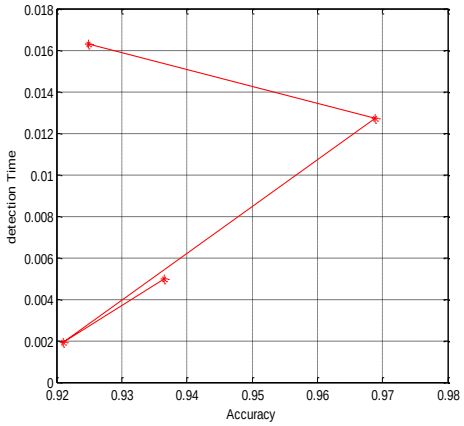
Detection time and training time is compared as shown in fig 5



Training time and accuracy is compared as shown in fig 6



Detection time and Accuracy is compared as shown in fig 7



As from the result shown above, the it is found that, when data size increase detection time also increases but accuracy is increased .TPR,TNR,FPR and FNR also shows the accuracy and efficiency of our system**.**

Accuracy	96.97%

Accuracy of our system is 96.97 % as shown in fig. 10 and which is the highest among the existing Sql-Injection detection techniques

V. CONCLUSIONS

Proposed work provides a secure application, based classification of Trusted URL and suspicious Blacklisted query strings using SVM. Here dataset of different size is used for training and classification .different parameters like accuracy, detection time ,training time ,tpr, tnr, fpr, fnr and the graphical description shows the performance of proposed system. This research work shows the best performance result in accuracy which is 96.97% and best among the existing systems...

.

REFERENCES

[1]. William G.J. Hal Fond and Alessandro Orso ,A Classification of SQL Inject ion At tacks and Countermeasures:, College of Computing, Georgia Institute of Technology.Gatech.edu.

[2]. D. Scott and R. Sharp, "Abstracting Application-level Web Security", In Proceedings of the 11th International Conference on the World Wide Web (WWW 2002), pp. 396–407, 2002.Y.

[3]. Huang, F. Yu, C. Hang, C. H. Tsai, D. T. Lee, and S. Y. Kuo. "Securing Web Application Code by Static Analysis and Runtime Protection", *Proc* 12th International World Wide Web Conference (WWW 04), May 2004.

[4]. SQL Injection Attack Examples based on the Taxonomy of Orso et al.

[5]. Xiang Fu, Xin Lu, Boris Peltsverger, Shijun Chen, "A Static Analysis Framework For Detecting SQL Injection Vulnerabilities", IEEE Transaction of computer software and application conference, 2007.

[6]. Konstantinos Kemalis and Theodoros Tzouramanis, "Specification based approach on SQL Injection detection", ACM, 2008.

[7]. G.T. Buehrer, B.W.Weide and P.A..G.Sivilotti, "Using Parse tree validation to prevent SQL Injection attacks", In proc. Of the 5th International Workshop on Software Engineering and Middleware (SEM '056), pp.106-113, Sep. 2005.

[8]. V.B. Livshits and M.S. Lam, "Finding Security vulnerability in java applications with static analysis", In proceedings of the 14th Usenix Security Symposium, Aug 2005.

[9]. William G.J. Halfond, Alessandro Orso, Panagiotis Manolios, "WASP: Protecting Web Applications Using Positive Tainting and Syntax-Aware Evaluation", IEEE Transaction of Software Engineering Vol34Nol, January/February 2008.

[10]. W.G. J. Halfond and A. Orso, "Combining Static Analysis and Run time monitoring to counter SQL Injection attacks", 3rd International workshop on Dynamic Analysis, St. Louis, Missouri, 2005, pp.1.

[11]. Marco Cova, DavideBalzarotti, ViktoriaFelmetsger, and Giovanni vigna, " Swaddler: An approach for the anamoly based character distribution models in the detection of SQL Injection attacks", Recent Advances in Intrusion Detection System, pp.63-86, Springerlink, 2007.

[12]. ntagw abira Lambert and kang Song Lin ," Use of Query Tokenization to detect and prevent SQL Injection Attacks", IEEE,2010.

[13]. Vipin Das 1, Vijaya Pathak2, Sattvik Sharma3,Sreevathsan4,MVVNS.Srikanth5,Gireesh Kumar T," Network Intrusion Detection System Based On Machine Learning Algorithms", IJCSIT, Vol 2, No 6, December 2010.

[14]. Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker ," Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs", *KDD'09,* June 28–July 1, 2009, Paris, France.