www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

A Flexible Vulnerability Reduction System Using Machine Learning

Dr. Shaji. N. Raj[1]
[1]Assistant Professor, SAS SNDP Yogam College, Konni Mahathma Gandhi University, Kerala, India.
[1]ammashajinraj@gmail.com

Abstract
The human system have an ability to adapt dynamically and protect against biological viruses is amazing. Computer security faces an ever-increasing threat and a system which can prevent any viruses coming in, is an open research problem. We propose a new model, called (RI Secure-Web), which can be resilient and immune to web application vulnerability for injection and URL manipulation for injection methods using an agent based machine learning system The ability of human immune system to survive and maintain body from different damages and its self-curing capability inspires the development of a resilient and adaptive cyber security system. Such system functions proactive and defends itself against viruses as human immune system does.

In this paper, an architectural view of a system for reducing application level vulnerabilities to protect cyber attacks, particularly injection method is proposed.

Key-words: RI Secure-Web, Intrusion Detection Systems (IDS), Vulnerability Analysis System.

1. Introduction

One of the major challenges of Intrusion Detection Systems (IDS) is to continuously watch network traffic, virus attacks and log files in real time. Also the delay in attack identification and to apply prevention is a major drawback of many of IDS. Furthermore, many existing systems are not built with learning capabilities. Here we propose a resilient system architecture, to develop a secure cyber security system. This system has subsystems of Vulnerability Analysis System, Intrusion Detection System, Intrusion Response System and Security Management System. The system uses secure mobile agents using negative and clonally selection algorithm. The work also talks about the system to learn and adapt to dynamic environments, but not to the new and unknown virus attacking using learning methods.

The paper is structured as follows: After introducing the background information on immune system and related concepts, literature review follows. Section 3 presents the proposed methodology and the new algorithm. The section 4 presents the results and evaluations. The paper concludes in Section 5, with future work.

2. Background Literature

There are some earlier works in cyber security that can adapt to dynamic environments like human immune system. Jamie paul's [18], thesis "Integrated Innate and Adaptive Artificial Immune Systems for Anomaly Detection" has been modelled on the biological adaptive immune system to an immune system for network security.

Muhammad Awais et al. [19] proposed an immune system for network security, using mobile agents and sensors for a comprehensive security system.

A survey paper written by Doyen Sahoo et.al [20] on malicious URL detection using machine learning provides better insights into URL manipulation with its various feature selection in lexical, host based, blacklisting and heuristic approach and also by using the classifiers SVM, Linear classifier, perceptrons with first order and second order algorithms, Naive bayes algorithms etc. The practical difficulties and issues in implementation in live environment are also discussed in detail.

In "Malicious sequential pattern mining for automatic malware detection" by Yujie Fan et al. [21] is showcased in a three step procedure: Intrusion sequence extractor, Malicious sequential pattern miner, and an ANN with K-nearest neighbour classifier to detect and predict different spams. Other than signature based detection, feature based detection is also applied exhaustively with preprocessing, relevant feature extraction and selection, and thereby prediction is made.

Anjali B. Sayamber, Arati M. Dixit, "On URL Classification" [22], described various types of attacks in URL and also used machine learning classifiers such as SVM, Linear regression, Naive bayes etc. The features considered are host based, lexical features, content based, DNS features, Link popularity etc. and its feature properties are also described in detail.

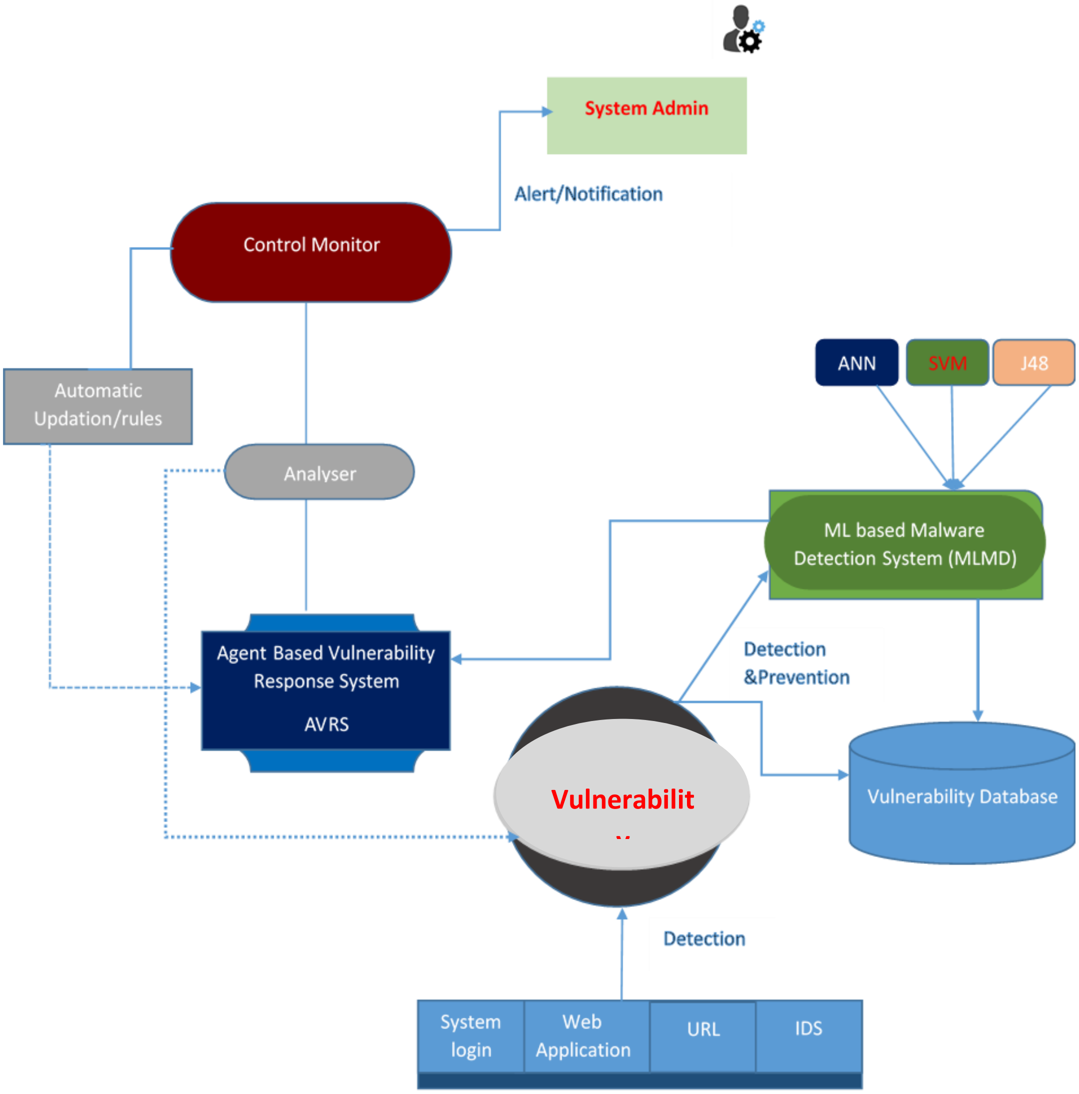
3. Proposed Model

We propose a new model, called (RI Secure-Web), which can be resilient and immune to web application vulnerability for SQL injection and URL manipulation for injection methods using an agent based machine learning system. The architecture of the system is depicted in Figure 1. The system also facilitates to connect to any existing Intrusion Detection system.

RI Secure-Web consists of mainly three components:

a. Agent based Vulnerability Response System (AVRS)
b. Vulnerability Identification for Web (VI-Web)
c. ML based Malware Detection System (MLMD)

Figure 1 - Architecture of RI Secure-Web



3.1. Agent Based Vulnerability Response System (AVRS)

An 'Agent' is a software that can act independently to perform certain tasks towards a goal. Mostly agents have dynamic behaviour and can be configured or performed at run time without

Received: 17.06.2021 – Accepted: 18.07.2021

human intervention. The characteristics of ‘Agents’ are Adaptation, Autonomy and Cooperation. Agent based IDS collects information about the intrusion and intruders, responds dynamically and reconfigures or blocks sites automatically. Attackers also try to disable firewalls, IDS and other anti-virus as a precaution to inject viruses that can also be blocked by agents to avoid their attacks and save the computer.

An effective IDS system is an elusive goal due to several reasons and each virus has its own behaviour. The complex behaviour and the newly penetrating viruses and attacks make security developers and researchers sleepless. A resilient and innate system for cyber security has to go a long way because a single mechanism will not fulfil the dream. The research is continuous in these areas to keep up with the increasing complexity of attacks.

The advantage of agent based system is its mobility and autonomy that can overcome certain intrinsic limitations of existing IDS. When applying mobile agents to this application domain, careful design choices are still required to take advantage of their traits. Developing multiple agent based Immune IDS is extremely challenging and complex.

The AVRS approach is explained as follows.

Agent based response system is connected to the vulnerability scanner and ML system. The vulnerability data is stored in a vulnerability database. Whenever any intrusion or attack is identified, it is notified and analysed by the Analyser through Central Manager. The cases are well studied by the control manager and actions will be prompted to prevent the system, where the attack is encountered. For example, if an attack is reported from a web application, ARS blocks the cookies, or for severe attacks, shuts down the system immediately. ARS will also notify the responses and changes to the system administrator. AVRS framework is only described here and its implementation is left as future work.

3.2. Vulnerability Identification for Web (VIWeb)

For VIWeb, vulnerability scanner defined in Figure 1 takes inputs from Reverse Resemblance Algorithm and Malicious String Matching Algorithm. The data will be stored in a vulnerability database. It can also accept other IDS and their output, stored in database. The database include the data from above algorithm. Reverse Resemblance Algorithm, the input is reversed and a special character is inserted and then it is compared with the reversed data in the database. In malicious string matching algorithm first process involves reading the URL and finding number of variables in the URL. Then it finds the user input values, and checks for malicious string.

3.3. Machine Learning based Malware Detection System (MLMD)

In MLMD, we have implemented three different machine learning algorithms from the dataset taken from vulnerability database stored in vulnerability scanner. In the following section, brief information about three machine learning techniques such as SVM, ANN and Decision Tree are described with its implementation details.

4. Machine Learning and Classification

Classification is one of the most widely used techniques in many machine learning applications, including cyber security for spam detection, Intrusion Detection Systems, URL Manipulations etc. Classification models allow to predict malicious and legitimate sites into two classes and predict when a new virus attacks the system.

Though there are a number of machine learning techniques, we have taken here three models, Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Decision tree with J48, which are proven in many real world applications to get state-of-the art performance.

Generally, many malicious sites are blacklisted using manual intervention, but effective only for known malicious URLs. As the number of malicious sites are increasing day by day, it is almost impossible to maintain an exhaustive list and also attackers are smart enough to create techniques to evade blacklisting. This method also fails to recognize new and unnoticed malicious URLs. The proposed Machine Learning technique identifies discriminative features of URL, that could distinguish malicious and benign URL over a large training set. The classification technique identifies the category of legitimate pages and malicious ones for different types of input mechanism vulnerabilities in Injection Method.

4.1. Experimentation with ANN, SVM and Decision Tree

In this experiment, we have considered URL lexical features for injection method. Since the URL properties vary for different types of attacks, choosing the parameters is very crucial for accurate classification. For example, phishing URLs and domains are known to exhibit characteristics that are different from other benign URLs and domains on the Web. Therefore, in this experiment, the URL properties considered are only relevant to injection method, that to categorise malicious or benign URL. We have taken extracted strings from Malicious String Matching Algorithm and the rest of the lexical features are extracted by a URL parser, which extracts tokens from a string delimited by

(‘/’, ‘?’, ‘.’, ‘-’, ‘_’, and ‘=’). Experiments were carried out on 89 instances with nine attributes namely name/phrase identified, encoded form of special character/operator, equality of LHS and RHS, repeated numbers, SQL keywords, logical operators, special characters apostrophe, and parenthesis. 10 fold cross validation is used for testing the performance.

The dataset is trained in ANN, SVM and Decision tree with J48 in Matlab. The performance is compared by analyzing algorithm generalization accuracy on the test set. The percentage of correctly classified instances, mean absolute error, root mean square error and kappa statistics are computed. Out of three classifiers, ANN outperforms with an accuracy of 86.50%. The architecture of ANN resulted as 9x3x2 with a learning rate of 0.3 and momentum 0.2. Learning rate is selected 0.3 because training will be progressed not slowly.

The Tables 1, 2, 3 and 4 show the performance comparison results.

Table 1 - Performance of Artificial Neural Network, Support Vector Machine and Decision Tree Methods

Algorithm	**Correctly Classified Instances**	**Incorrectly Classified Instances**	**Accuracy**
ANN	77	12	86.50%
SVM	73	16	82.00%
J48	72	17	80.90%

Table 2 - Performance of Artificial Neural Network, Support Vector Machine, Decision Tree Method in Terms of Mean Absolute Error, RMS Error and Kappa Statistics

Algorithm	**Mean absolute error**	**Root mean squared error**	**Kappa statistic**
J48	0.2143	0.3709	0.5696
SVM	0.1798	0.424	0.5749
ANN	0.1402	0.2971	0.6748

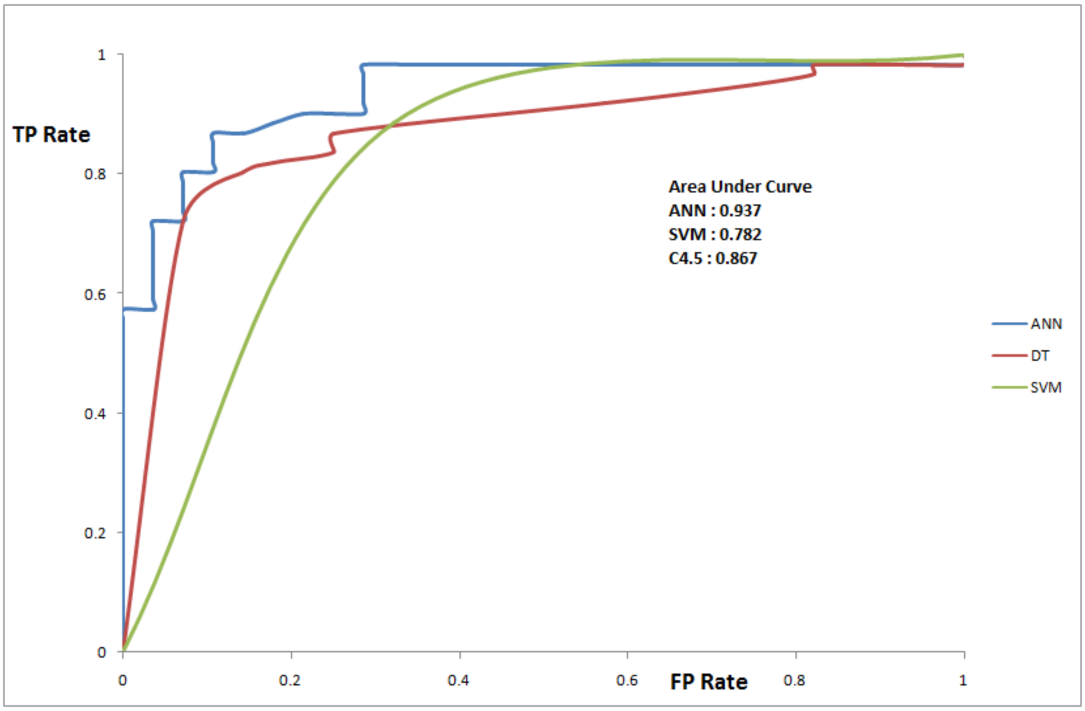
Table 3 - Comparison of TP Rate, FP Rate, Precision, Recall, f Measure of the Classifiers

Algorithm	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F Measure**	**Class**
ANN	0.934	0.286	0.877	0.934	0.905	SQL Injection
	0.714	0.066	0.833	0.714	0.769	Valid
SVM	0.885	0.321	0.857	0.885	0.871	SQL Injection
	0.679	0.115	0.731	0.679	0.704	Valid
J48	0.836	0.250	0.879	0.836	0.857	SQL Injection
	0.750	0.164	0.677	0.750	0.712	Valid

Table 4 - Area Under Curve (AUC) of the Classifiers

Class	**Algorithm**		
	J48	**SVM**	**ANN**
SQL injection	0.867	0.782	0.937
Valid	0.867	0.782	0.937

Figure 2 - ROC Curve for ANN, SVM and J48 Algorithm



5. Summary

In this paper two works have been presented on URL Scanner for injection methods. The proposed Malicious String Matching algorithm applied on URL scanning for injection method showed 95.54% of accuracy. A resilient immune vulnerability reduction system using ML techniques is also applied to protect against ever increasing threats. The lexical features for injection method is taken for training that learn to adapt to dynamic changes. An architectural view of an Agent based Vulnerability Response System is also proposed, which can act as an alert, notification and quick response for modification or reconfiguration automatically. The work here has concentrated on implementation of a resilient security system using three ML techniques such as ANN, SVM and J48, out of which ANN performed the best with 86.50% accuracy.

References

D. Avresky, J. Arlat, J. Laprie and Y. Crouzet, "Fault injection for formal testing of fault tolerance", *IEEE Transactions on Reliability*, vol. 45, no. 3, pp. 443-455, 1996.

Advanced Automated SQL Injection Attacks and Defensive Mechanisms", *IEEE Trans. on Security*, 456-463, 2016.

W. Halfond, J. Viegas and A. Orso, *"A Classification of SQL Injection Attacks and Countermeasures"*, 2006, pp. 145-153.

P.V. Vilasini, "Eliminate SQL Injection Using LINQ", *IJARCST*, vol. 2, no. 1, pp. 361-369, 2004.

J. Atoum and A. Qaralleh, "A Hybrid Technique for SQL Injection Attacks Detection and Prevention", *International Journal of Database Management Systems*, vol. 6, no. 1, pp. 21-28, 2014.

W. Stallings, *Cryptography and network security*, 2nd ed. Upper Saddle River, N.J.: Pearson/Prentice Hall, 2006, p. 345.

A.N, M. Varun Kumar and V. Vaidhyanathan. G, "Preventing SQL Injection Attacks", *International Journal of Computer Applications*, vol. 52, no. 13, pp. 28-32, 2012.

G. Yiğit and M. Arnavutoğlu, "SQL Injection Attacks Detection & Prevention Techniques", *International Journal of Computer Theory and Engineering*, vol. 9, no. 5, pp. 351-356, 2017.

B.B. Hanmanthu, B. Eaghu Ram and Dr.P. Niranjan, "SQL Injection Prevention Based on Decision Tree Classification", in *IEEE International Conference on Intelligent System and Control*, 2015.

H. Kaur and M. Dhingra, "A Practical Approach for SQL Injection Prevention Attacks Using IPS", *IJARCCE*, pp. 8118-8123, 2014.

Indrani Balasundaram and E. Ramraj, "An authentication Mechanism to Prevent SQL Injection Attacks", *International Journal of Computer Application*, vol. 19, no. 1, pp. 30-36, 2011.

I. Lee, S. Jeong, S. Yeo and J. Moon, "A novel method for SQL injection attack detection based on removing SQL query attribute values", *Mathematical and Computer Modelling*, vol. 55, no. 1-2, pp. 58-68, 2012.

Srinivas Avireddy and Varalakshmi Perumal, "Random 4: An Application Specific Randomized Encryption Algorithm to Prevent SQL Injection", in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, 1327-1338.

S. Anjugam and A. Murugan, "Efficient methods for preventing SQL injection attack on a web application using encryption and tokenization", *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(3), 173-177, 2014.

Ramya Dharam and Sajjan G. Shiva, "Runtime Monitoring Technique to handle Tautology based SQL Injection Attacks", *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, vol. 2, pp. 189-192, 2012.

J. Fonseca, M. Vieira and H. Madeira, "Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection", *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 440-453, 2014.

"SQL Injection", *W3schools.com*, 2017. https://www.w3schools.com/sql/sql_injection.asp.

Jamie paul, *"Integrated Innate and Adaptive Artificial Immune Systems for Anomaly Detection"*, University of Nottingham, 2007.

Muhammad Awais Shibli, "MagicNET: The Human Immune System and Network Security System", *Paper.ijcsns.org*, 2009. http://paper.ijcsns.org/07_book/200901/20090113.pdf.

Doyen Sahoo, Steven C.H. Hoi and Chenghao Liu, "Malicious URL Detection using Machine Learning: A Survey", *https:/ /arxiv.org /abs/ 1701.07179*, 2017.

Y. Fan, Y. Ye and L. Chen, "Malicious sequential pattern mining for automatic malware detection", *Expert Systems with Applications*, vol. 52, pp. 16-25, 2016.

Anjali B. Sayamber and Arati M. Dixit, "On URL Classification", *Ijcttjournal.org*, 2018. http:// www. IJCTT journal. org/Volume12/number-5/IJCTT-V12P148.pdf

Feroz Zahid, *"Network Optimization for High-Performance Cloud Computing"*, University of Oslo, 2017.