ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Intelligent Server Based Data Security Enhanced With Attack Avoidance Scheme

S. Muthumeenakshi[1], S. Aarthi[2], N. Krishnaveni[3], G. Megala[4]

B.E. Computer Science and Engineering, Department of CSE, MIET Engineering College, Trichy - Pudukottai Road, Tiruchirapalli , Tamil Nadu, India.

B.E. Computer Science and Engineering, Department of CSE, MIET Engineering College, Trichy - Pudukottai Road, Tiruchirapalli, Tamil Nadu, India.

B.E. Computer Science and Engineering, Department of CSE, MIET Engineering College, Trichy - Pudukottai Road, Tiruchirapalli, Tamil Nadu, India.

Assistant Professor, Department of CSE, MIET Engineering College, Trichy - Pudukottai Road, Tiruchirapalli, Tamil Nadu, India.

ABSTRACT: The main objective of this system is to shows that the injection of vulnerabilities and attacks over the server and implementing an effective way to evaluate security mechanisms as well as to point out not only their weaknesses but also ways for their improvement. Attacks and vulnerabilities are the major threats over data security domain. Data can be affected by the attackers via various modes such as: (i) Monitoring the data in hidden manner, (ii) Modify the data into the server and (iii) Erase the data from the server. Web based application platforms are the major target to the attackers to attack the data. Two new algorithms are introduced to identify the attack possibilities and overcome it. (i) Generic Cryptographic Algorithm and (ii) Fault Injection Reduction Algorithm these two innovative algorithms are combine worked together and efficiently safeguard our data from attacks. Combination of these two algorithms is named as Vulnerability and Attack Injector Tool (VAIT).

KEYWORDS: Parameterization Approach, SQL Injection, Data security, Generic Cryptographic Algorithm, Fault Injection Reduction Algorithm.

I. INTRODUCTION

The selected domain for implementation of the above mentioned title is Web and Information Security with Data Protection logics. This domain is most important now-a-days because all the users maintain the records in unknown server, so all should have a reputation like how the server is secured and how it is reliable.

Web Information Security deals with protecting digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. This Security measure also protects data from corruption. It is an essential aspect of Information Technology for organizations of every size and type. Information Security is also known as Information System Security or Computer Security. The current industry of web based application platform suffered by various types of attacks and vulnerabilities, the major thing which affects them by means of injecting realistic vulnerabilities in a web application and attacking the application to make them to perform with low efficiency and cost expensive manner. In this approach we implement two new algorithms which efficiently identifying the attack possibilities and overcome the problem of injection and vulnerabilities in the web application portal, that is called Generic Cryptographic Algorithm and Fault Injection Reduction Algorithm.

These two algorithms work in combined manner to identify the possibility of attacks and prove this proposed tool called the Vulnerability & Attack Injector Tool (VAIT) provides efficient results compare to all the other methods like IDS and so on. Conversely, the valid web requests (SQLIA negative) would take the form of generating all

DOI: 10.15680/IJIRCCE.2018.0603087

possible member strings. We apply Non-Deterministic Finite Automata (NFA) implemented in Regular Expression (RegEx) to define the constraint patterns, and employing Symbolic Finite Automata (SFA) with a constraint solver termed Satisfiability Modulo Theories (SMT-Z3) to generate member strings from the defined RegEx patterns. We trained a supervised learning model with this pattern driven learning data in demonstrating a proof of concept by applying predictive analytics to a test web application expecting dictionary word list as input data.

The learning data are obtained by automata states walk to derive as many member strings in a given pattern. The trained models are evaluated in ROC curve with the TC SVM having the best performance metrics in Area Under Curve (AUC) value of 0.986 deployed as a Web Service (WS). The WS is consumed in a Fiddler proxy Application Programming Interface (API) for the ongoing SQLIA prediction as to reject intercepted requests that are positive.

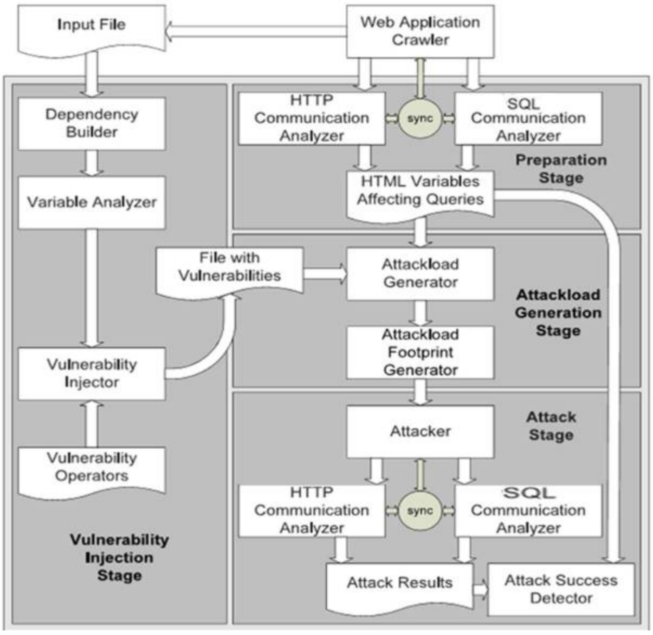

Input File
Web Application Crawler
Dependency Builder
Variable Analyzer
HTTP Communication Analyzer
sync
SQL Communication Analyzer
Preparation Stage
HTML Variables Affecting Queries
File with Vulnerabilities
Attackload Generator
Attackload Generation Stage
Attackload Footprint Generator
Vulnerability Injector
Attacker
Attack Stage
Vulnerability Operators
HTTP Communication Analyzer
sync
SQL Communication Analyzer
Vulnerability Injection Stage
Attack Results
Attack Success Detector

Fig.1 Proposed System Archietctural Design

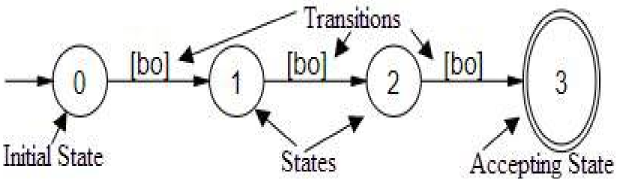


Fig.2 States walk from expected input string to generate member strings

II. EXISTING APPROACHES – A SUMMARY

To handle web application security, new tools need to be developed, and procedures and regulations must be improved, redesigned or invented. Moreover, everyone involved in the development process should be trained properly. All web applications should be thoroughly evaluated, verified and validated before going into production. However, these best practices are unfeasible to apply to the hundreds of millions of existing legacy web applications, so they should be constantly audited and protected by security tools during their lifetime. This is particularly relevant due to the extreme dynamicity of the security scenario, with new vulnerabilities and ways of exploitation being discovered every day. The existing system has several disadvantages, some of them are listed below: (i) Clearly, data security technology is not good enough to stop attacks and practitioners should be concerned with the evaluation and the assurance of their success, (ii) Web based security mediums are more complicated compare to regular approaches and (iii) In practice, there is a need for new ways to effectively test existing web application security mechanisms in order to evaluate and improve them.

III. PROPOSED SYSTEM

The methodology proposed was implemented in a concrete Vulnerability & Attack Injector Tool (VAIT) for web applications. The tool was tested on top of widely used applications in two scenarios. The first to evaluate the effectiveness of the VAIT in generating a large number of realistic vulnerabilities for the offline assessment of security tools, in particular web application vulnerability scanners. The second to show how it can exploit injected vulnerabilities to launch attacks, allowing the online evaluation of the effectiveness of the counter measure mechanisms installed in the target system, in particular an intrusion detection system. Generic Cryptographic Algorithm and Fault Injection Reduction Algorithm are proposed to evaluate the web application security mechanisms. The proposed system has several advantages, some of them are listed below: (i) Both static and dynamic data analysis are used, which is a key feature of the proposed methodology, (ii) Overall performance and effectiveness are improved, (iii) The proposed methodology provides a practical environment that can be used to test countermeasure mechanisms (such as intrusion detection systems (IDSs), web application vulnerability scanners, web application fire-walls, static code analyzers, etc.), train and evaluate security teams, help estimate security measures (like the number of vulnerabilities present in the code), among others and (iv) More fast and accurate in results.

IV. SYSTEM IMPLEMENTATION

The proposed system is implemented with the help of following modules; all of them are described in detail below:

A. Security Counter Measurement

The Security Counter Measurement module is relevant to the extreme dynamicity of the security in web oriented applications, with new vulnerabilities and ways of exploitation being discovered every day. Clearly, security technology is not good enough to stop web application attacks and practitioners should be concerned with the evaluation and the assurance of their success. To handle web application security, new tools need to be developed, and procedures and regulations must be improved, redesigned or invented. Moreover, everyone involved in the development process should be trained properly. All web applications should be thoroughly evaluated, verified and validated before going into production. However, these best practices are unfeasible to apply to the hundreds of millions of existing legacy web applications, so they should be constantly audited and protected by security tools during their lifetime.

B. Vulnerability Injection over Web

The past approaches like validation prevention and master data correction methods fails to propose a methodology and a tool for identifying the intruders and prevent them to inject vulnerabilities and attacks in web applications. The proposed methodology is based on the new idea of Fault Injection Detection Method, from that we can assess different attributes of existing web application security mechanisms by injecting realistic vulnerabilities in a web application and attacking them automatically. This follows a procedure inspired on the fault injection technique that has been used for decades in the dependability area. The attack injection methodology is based on the dynamic analysis of information obtained from the runtime monitoring of the web application behavior and of the interaction with external resources, such as the backend database. This information, complemented with the static analysis of the source code of the application, allows the effective injection of vulnerabilities that are similar to those found in the real world. This module provides the overall scenario of web based application affection and its remedies to overcome from those attacks.

C. Database Injection Evaluation

The proposed methodology of database injection evaluation provides a practical environment that can be used to test countermeasure mechanisms (such as intrusion detection systems (IDSs), web application vulnerability scanners, web application firewalls, static code analyzers, etc.), train and evaluate security teams, help estimate security measures (like the number of vulnerabilities present in the code), among others. The attack injection methodology is based on the dynamic analysis of information obtained from the runtime monitoring of the web application behavior and of the interaction with external resources, such as the backend database. This assessment of security tools can be done online by executing the attack injector while the security tool is also running; or offline by injecting a representative set of vulnerabilities that can be used as a testbed for evaluating a security tool.

D. Attack Injection Exploitation Circumstances

The attack injection circumstances is summarized by means of the following work scenarios, imagine the two various scenarios as the most relevant exploitations of the proposed attack injection methodology and VAIT:
(a) Inline Circumstances: In the inline scenario, the VAIT can be used to evaluate tools and security assurance mechanisms, like IDS for databases, web application IDS, web application firewalls and reverse proxies.
(b) Offline Circumstances: In the offline scenario, the VAIT injects vulnerabilities into the web application and attacks them to check if they can be exploited or not. The outcome is the set of vulnerabilities that can, effectively, be attacked. They can then be used in a variety of situations, such as: to provide a test bed to train and evaluate security teams that are going to perform code review or penetration testing, to test static code analyzers, to estimate the number of vulnerabilities still present in the code, to evaluate web application vulnerability scanners, etc.

V. LITERATURE SURVEY

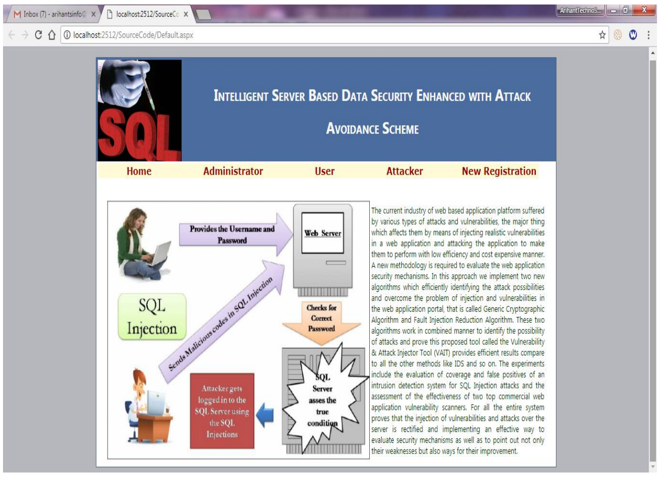
In the year of 2016, the authors "I. Medeiros, N. Neves, and M. Correia" proposed a paper titled “Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining", in that they described such as: although a large research effort on web application security has been going on for more than a decade, the security of web applications continues to be a challenging problem. An important part of that problem derives from vulnerable source code, often written in unsafe languages like PHP. Source code static analysis tools are a solution to find vulnerabilities, but they tend to generate false positives, and require considerable effort for programmers to manually fix the code. We explore the use of a combination of methods to discover vulnerabilities in source code with fewer false positives. We combine taint analysis, which finds candidate vulnerabilities, with data mining, to predict the existence of false positives. This approach brings together two approaches that are apparently orthogonal: humans coding the knowledge about vulnerabilities (for taint analysis), joined with the seemingly orthogonal approach of automatically obtaining that knowledge (with machine learning, for data mining). Given this enhanced form of detection, we propose doing automatic code correction by inserting fixes in the source code. Our approach was implemented in the WAP tool, and an experimental evaluation was performed with a large set of PHP applications. Our tool found 388 vulnerabilities in 1.4 million lines of code. Its accuracy and precision were approximately 5% better than PhpMinerII's and 45% better than Pixy's.

In the year of 2016, the authors "S. Uwagbole, W. Buchanan, and L. Fan" proposed a paper titled “Numerical Encoding to Tame SQL Injection Attacks", in that they described such as: recent years have seen an astronomical rise in SQL Injection Attacks (SQLIAs) used to compromise the confidentiality, authentication and integrity of organisations' databases. Intruders becoming smarter in obfuscating web requests to evade detection combined with increasing volumes of web traffic from the Internet of Things (IoT), cloud-hosted and on-premise business applications have made it evident that the existing approaches of mostly static signature lack the ability to cope with novel signatures. A SQLIA detection and prevention solution can be achieved through exploring an alternative bio-inspired supervised learning approach that uses input of labelled dataset of numerical attributes in classifying true positives and negatives. We present in this paper a Numerical Encoding to Tame SQLIA (NETsQlIA) that implements a proof of concept for scalable numerical encoding of features to a dataset attributes with labelled class obtained from deep web traffic analysis. In the numerical attributes encoding: the model leverages proxy in the interception and decryption of web traffic. The intercepted web requests are then assembled for front-end SQL parsing and pattern matching by applying traditional Non-Deterministic Finite Automaton (NFA). This paper is intended for a technique of numerical attributes extraction of any size primed as an input dataset to an Artificial Neural Network (ANN) and statistical Machine Learning (ML) algorithms implemented using Two-Class Averaged Perceptron (TCAP) and Two-Class Logistic Regression (TCLR) respectively. This methodology then forms the subject of the empirical evaluation of the suitability of this model in the accurate classification of both legitimate web requests and SQLIA payloads.

In the year of 2017, the authors "S. O. Uwagbole, W. J. Buchanan, and L. Fan" proposed a paper titled “Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention", in that they described such as: the back-end database is pivotal to the storage of the massive size of big data Internet exchanges stemming from cloud-hosted web applications to Internet of Things (IoT) smart devices. Structured Query Language (SQL) Injection Attack (SQLIA) remains an intruder's exploit of choice on vulnerable web applications to pilfer confidential data from the database with potentially damaging consequences. The existing solutions of mostly signature approaches were all before the recent challenges of big data mining and at such lacks the functionality and ability to cope with new signatures concealed in web requests. An alternative Machine Learning (ML) predictive analytics provides a functional and scalable mining to big data in detection and prevention of SQLIA. Unfortunately, lack of availability of readymade robust corpus or data set with patterns and historical data items to train a classifier are issues well known in SQLIA research. In this paper, we explore the generation of data set containing extraction from known attack patterns including SQL tokens and symbols present at injection points. Also, as a test case, we build a web application that expects dictionary word list as vector variables to demonstrate massive quantities of learning data. The data set is pre-processed, labelled and feature hashing for supervised learning. The trained classifier to be deployed as a web service that is consumed in a custom dot NET application implementing a web proxy Application Programming Interface (API) to intercept and accurately predict SQLIA in web requests thereby preventing malicious web requests from reaching the protected back-end database. This paper demonstrates a full proof of concept implementation of an ML predictive analytics and deployment of resultant web service that accurately predicts and prevents SQLIA with empirical evaluations presented in Confusion Matrix (CM) and Receiver Operating Curve (ROC).

VI. EXPERIMENTAL RESULTS

The following figure illustrates the home page of the proposed system.

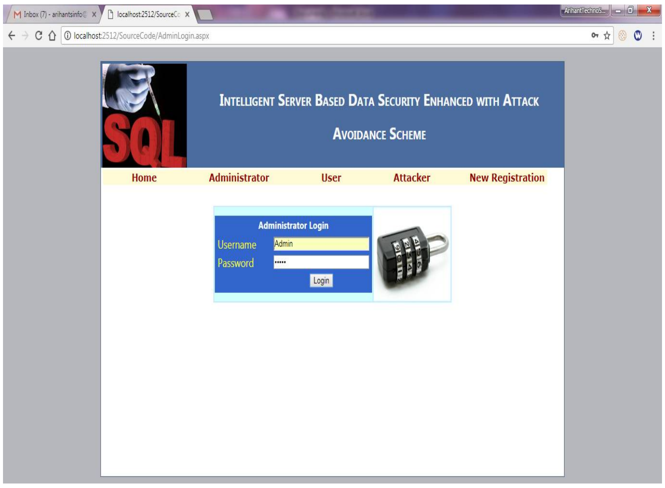


Inbox (7) - arihantsinfo@
localhost:2512/SourceCo
ArihantTechnoS...
localhost:2512/SourceCode/Default.aspx
SQL
INTELLIGENT SERVER BASED DATA SECURITY ENHANCED WITH ATTACK AVOIDANCE SCHEME
Home
Administrator
User
Attacker
New Registration
Provides the Username and Password
Web Server
SQL Injection
Sends Malicious codes in SQL Injection
Checks for Correct Password
SQL Server asses the true condition
Attacker gets logged in to the SQL Server using the SQL Injections

The current industry of web based application platform suffered by various types of attacks and vulnerabilities, the major thing which affects them by means of injecting realistic vulnerabilities in a web application and attacking the application to make them to perform with low efficiency and cost expensive manner. A new methodology is required to evaluate the web application security mechanisms. In this approach we implement two new algorithms which efficiently identifying the attack possibilities and overcome the problem of injection and vulnerabilities in the web application portal, that is called Generic Cryptographic Algorithm and Fault Injection Reduction Algorithm. These two algorithms work in combined manner to identify the possibility of attacks and prove this proposed tool called the Vulnerability & Attack Injector Tool (VAIT) provides efficient results compare to all the other methods like IDS and so on. The experiments include the evaluation of coverage and false positives of an intrusion detection system for SQL Injection attacks and the assessment of the effectiveness of two top commercial web application vulnerability scanners. For all the entire system proves that the injection of vulnerabilities and attacks over the server is rectified and implementing an effective way to evaluate security mechanisms as well as to point out not only their weaknesses but also ways for their improvement.

Fig.3 Home Page Design

The following figure illustrates the Administrator Authentication page of the proposed system.



INTELLIGENT SERVER BASED DATA SECURITY ENHANCED WITH ATTACK AVOIDANCE SCHEME

Administrator Login

Username Admin

Password •••••

Fig.4 Administrator Authentication

VII. CONCLUSION

This approach proposes a novel methodology to automatically inject realistic attacks over data in web applications and real-time environments. This methodology consists of analyzing the web application and generating a set of potential vulnerabilities. Each vulnerability is then injected and various attacks are mounted over each one. The success of each attack is automatically assessed and reported.

REFERENCES

[1] USA, “Sarbanes-Oxley Act,” 2002.
[2] Payment Card Industry (PCI) Data Security Standard, PCI Security Standards Council, 2010.
[3] S. Christey and R. Martin, “Vulnerability Type Distributions in CVE,” Mitre Report, May 2007.
[4] S. Zanero, L. Carettoni, and M. Zanchetta, “Automatic Detection of Web Application Security Flaws,” Black Hat Briefings, 2005.
[5] N. Jovanovic, C. Kruegel, and E. Kirda, “Precise Alias Analysis for Static Detection of Web Application Vulnerabilities,” Proc. IEEE Symp. Security Privacy, 2006.
[6] J. Williams and D. Wichers, “OWASP Top 10,” OWASP Foundation, Feb. 2013.
[7] IBM Global Technology Services “IBM Internet Security Systems X-Force 2012 Trend & Risk Report,” IBM Corp., Mar. 2013.
[8] Verizon “2011 Data Breach Investigations Report,” 2011.
[9] The Privacy Rights Clearinghousewww.privacyrights.org/databreach, Accessed 1 May 2013, Apr. 2012.
[10] M. Fossi, et al., “Symantec Internet Security Threat Report: Trends for 2010,” Symantec Enterprise Security, 2011.
[11] W. G. J. Halfond, A. Orso, and P. Manolios, “WASP: Protecting web applications using positive tainting and syntax-aware evaluation,” IEEE Trans. Softw. Eng., vol. 34, no. 1, pp. 65–81, 2008.
[12] A. Kie, P. J. Guo, and M. D. Ernst, “Automatic Creation of SQL Injection and Cross-Site Scripting Attacks,” 2009.
[13] C. Bockermann, M. Apel, and M. Meier, “Learning SQL for database intrusion detection using context-sensitive modelling (extended abstract),” in Lecture Notes in Computer Science, 2009, vol. 5587 LNCS, pp. 196–205.
[14] R. Komiya, I. Paik, and M. Hisada, “Classification of malicious web code by machine learning,” in Proceedings of 2011 3rd International Conference on Awarene ss Science and Technology, iCAST 2011, 2011, pp. 406–411.
[15] J. Choi, C. Choi, H. Kim, and P. Kim, “Efficient malicious code detection using N-gram analysis and SVM,” in Proceedings - 2011 International Conference on Network-Based Information Systems, NBiS 2011, 2011, pp. 618–621.
[16] Y. Wang and Z. Li, “SQL Injection Detection via Program Tracing and Machine Learning,” LNCS, vol. 7646, pp. 264–274, 2012.
[17] C. I. Pinzón, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo, J. M. Corchado, C. I. Pinz??n, J. F. De Paz, ??lvaro Herrero, E. Corchado, J. Bajo, and J. M. Corchado, “IdMAS-SQL: Intrusion Detection Based on MAS to Detect and Block SQL injection through data mining,” Inf. Sci. (Ny)., vol. 231, pp. 15–31, 2013.
[18] M.-Y. Y. Kim and D. H. Lee, “Data-mining based SQL injection attack detection using internal query trees,” Expert Syst. Appl., vol. 41, no. 11, pp. 5416–5430, 2014.
[19] S. O. Uwagbole, W. J. Buchanan, and L. Fan, “Applied Machine Learning Predictive Analytics to SQL Injection Attack Detection and Prevention,” in 3rd IEEE/IFIP Workshop on Security for Emerging Distributed Network Technologies (DISSECT), 2017.
[20] S. O. Uwagbole, W. Buchanan, and L. Fan, “Applied web traffic analysis for numerical encoding of SQL injection attack features,” in European Conference on Information Warfare and Security, ECCWS, 2016, vol. 2016–Janua.