Detecting and Correcting Web Vulnerabilities With Taint Analysis

Pruthvi H M [1], DR Sreepathi .B [2], T.Sumarani[3], Anusha H[4], Divya Bharathi[5]
[1, 2] Dept of ISE
[1, 2] RYMEC, Ballari

Abstract- *The security of web applications continues to be challenging problem.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 less false positives. We combine taint analysis, which finds candidate vulnerabilities, with data mining, in order 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) versus 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.*

Keywords- Automatic, data mining, false positives, input validation vulnerabilities, software security, source code static analysis, web applications.

I. INTRODUCTION

SINCE its appearance in the early 1990s, the Web evolved from a platform to access text and other media to a framework for running complex web applications. The main contributions of the paper are: (1) an approach for improving the security of web applications by combining detection and automatic correction of vulnerabilities in web applications; (2) a combination of taint analysis and data mining techniques to identify vulnerabilities with low false positives; (3) a tool that implements that approach for web applications written in PHP with several database management systems; (4) a study of the configuration of the data mining component and an experimental evaluation of the tool with a considerable number of open source PHP applications. Some of the web security vulnerabilities: SQL injections, Cross site scripting(XSS) ,Broken Authentication and Session Management, Insecure Direct references, Security Misconfiguration, Cross Site Request Forgery(CSRF).

1.1 EXISTING SYSTEM

In the existing system, the system begins by giving a survey of web application attacks and vulnerabilities, also approaches to improve the web application security using intrusion detection systems and scanners based on machine learning and artificial intelligence. When it comes to vulnerability, it is also an attack which exploits this vulnerability; therefore the existing system presents web intrusion detection system based on detection of web vulnerabilities. Experimental results have been acquired from HTTP simulations in our network and from responses of HTTP requests sent to a bunch of websites and applications to test the efficiency of our intrusion detection system. This efficiency can be noticed from a High detection rate which is greater than 90%.

1.2 PROPOSED SYSTEM

In the proposed system, the system explores the use of a novel combination of methods to detect this type of vulnerabilities: static analysis and data mining. Static analysis is an effective mechanism to find vulnerabilities in source code, but tends to report many false positives (non-vulnerabilities) due to its undesirability. This problem particularly difficult with languages such as PHP that are weakly typed and not formally specified. Therefore, the system complements a form of static analysis, taint analysis, with the use of data mining to predict the existence of false positives. This solution combines two apparently opposite approaches: humans coding the knowledge about vulnerabilities (for taint analysis) versus automatically obtaining that knowledge (with supervised machine learning supporting data mining).

II. MODULES

Admin In this module, admin has to login with valid username and password. After login successful he can do some operations such as view all user, their details and authorize them , view all owners, their details and authorize them, view all attackers details view all sql injection vulnerabilities and block them, view all file access vulnerabilities of users , view all blocked data owners , view unblock requests and unblock them, view all secret key requests and generate, view all users download history, view SQL Injection Vulnerabilities in chart, View number of remote vulnerabilities in chart. User In this module, there are n numbers of users are present. User should register before doing some operations. After registration successful he can login by using valid user name and password. Login successful he will do some operations like view profile details, request secret key and view response, search documents and download by entering secret key. Data Owner In this module, there are n numbers of owners are present. Owner should register before doing some operations. After registration successful he can login by using valid user name and password. Login successful he will do some operations like view profile details, Upload documents and generate digital sign, view uploaded documents, verify his documents and recover if it is attacked, view all on his documents like, Execute SQL queries if query is incomplete the it is SQL Injection Vulnerabilities.

III. SYSTEM ARCHITECTURE

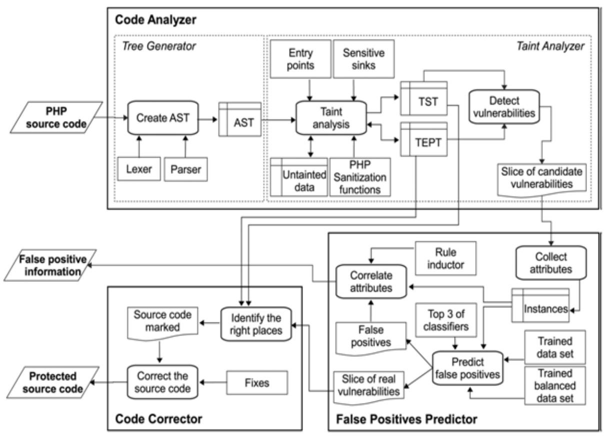

FIG-1: Architecture including main modules and data structures.

Figure shows system architecture which is composed of 3 modules: Code analyzer, False Positive Predictor, Code Corrector. The code analyzer first parses the PHP source code and generates AST then it use free walkers to do taint analysis, i.e to track data supplied by users through entry points reaches sensitive sinks without sanitization .The false positive predictor continues where the code analyzer stops. For every sensitive sinks that was found to be reached by tainted input, it tracks the path from that sink to the entry point using tables and trees .Code Corrector picks the path classified as true positives to signal the tainted inputs to be sanitized using the tables and trees. This architecture also represents the architecture of WAP tool.

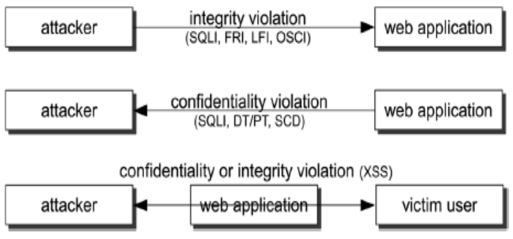

FIG-2: Information flows that exploit web vulnerabilities

Figure shows the information flows the exploit each of the vulnerabilities. The information flows are labeled with vulnerabilities that usually permit them.XSS is a different from other vulnerabilities because the victim is not the web application itself but a user.Our approach is a way of enforcing information flow security at language level,The tool detects the possibility of exisiting the information flow represented in the figure and modifies the source code to prevent them.

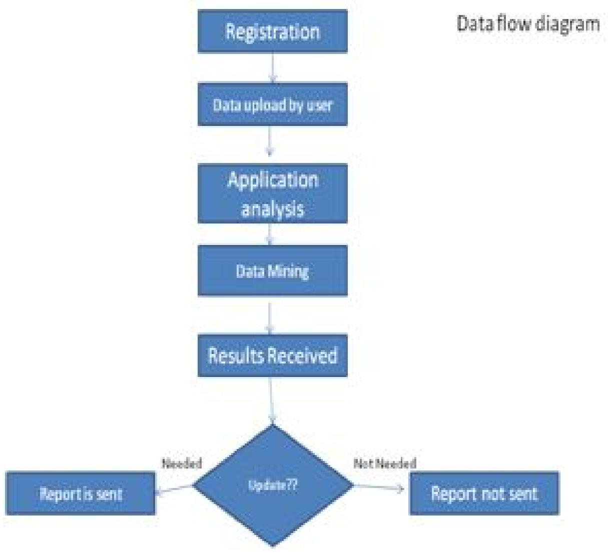

FIG-3: Data Flow diagram.

IV. FEASIBILITY ANALYSIS

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

Operational Feasibility
Economic Feasibility
Technical Feasibility

Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

V. SYSTEM DESIGN AND DEVELOPMENT

INPUT DESIGN

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rests with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer in used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

VI. CONCLUSION

The paper presents an approach for finding and correcting vulnerabilities in web applications and a tool that implements the approach for PHP programs and input validation vulnerabilities. The approach and the tool search for vulnerabilities using a combination of two techniques: static source code analysis and data mining. Data mining is used to identify false positives using a top 3 of machine learning classifiers and to justify their presence using an induction rule classifier .All classifiers were selected after a thorough comparison of several alternatives. It is important to note that this combination of detection techniques cannot provide entirely correct results. The static analysis problem is un decidable and the resort to data mining cannot circumvent this un decidability, only provide probabilistic results. The tool corrects the code by inserting fixes, i.e., sanitization and validation functions. Testing is used to verify if the fixes actually remove the vulnerabilities and do not compromise the (correct) behavior of the applications. The tool was experimented with synthetic code with vulnerabilities inserted on purpose and with a considerable number of open source PHP application. It was also compared with two source code analysis tools, Pixy and PhpMinerII. This evaluation suggests that the tool can detect and correct the vulnerabilities of the classes it is programmed to handle. It was able to find 388 vulnerabilities in 1.4 million lines of code. Its accuracy and precision were approximately 5% better than PHP Miner's and 45% better than Pixy's.

REFERENCES

[1] WAP tool website. http://awap.sourceforge.net/.

[2] J. Antunes, N. F. Neves, M. Correia, P. Verissimo, a nd R. Neves. Vulnerability removal with attack injection. IEEE Transactions on Software Engineering, 36(3):357–370, 2010.

[3] E. Arisholm, L. C. Briand, and E. B. Johannessen. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. Journal of Systems and Software, 83(1):2–17, 2010.

[4] R. Banabic and G. Candea. Fast black-box testing of system recovery code. In Proceedings of the 7th ACM European Conference on Computer Systems, pages 281–294, 2012.