www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 03 Issue 07 July, 2014 Page No. 6985-6989

Security Assessment Automation Framework: Web Applications

Gopal R. Choudhari[1], Prof Madhav V. Vaidya[2],

[1]Department of Information Technology, SGGS IE & T,
Nanded, Maharashtra, India
chaudharigopal88@gmail.com

[2]Department of Information Technology, SGGS IE & T,
Nanded, Maharashtra, India
mvvaidya@gmail.com

Abstract: ***The security of the Web applications has increased rapidly over the last years. At a same time, the quantity and impact of the vulnerabilities in the Web applications have grown as well. Since the manual code reviews are time-consuming, costly and error-prone, the need for the automated solutions has become evident.***
In this paper, we proposed an automated security assessment framework for Web applications. The purpose of this system is to improve the security standard of software products and applications. The end-end framework gathers information from potential clients; helps determine the scope of assessments, tools to use and the methodology for conducting assessments.
It also generates a report showing graphs and provides actionable intelligence about identified vulnerabilities. It can also be integrated with build systems used for developing and deploying applications so that security issues are caught in early phases of the SDLC. The system works as a single point of control for running security tools and scripts and managing information about security projects. By providing a single point of control, the system automates delivery of security solutions.

Keywords: Software Security, Automated testing, Web application vulnerabilities, secure coding.

1. Introduction

Now days the Web applications are developed using several different and interacting technologies. Developing a Web application using these different technologies causes vast body of security vulnerabilities. Often searching for these vulnerabilities there are many ways including manual test, automated test, pen test, using automated security assessment tools, etc.
In this paper, we proposed an automated security assessment framework for Web applications. The purpose of this system is to improve the security standard of software products and applications using automated the software security assessment process with cost-effective methods for detecting a range of important vulnerabilities.

The system has a central single point of control that automates delivery of the security solutions. The system has built-in-scripting engine that confirms the positivity of the vulnerabilities. The system has functionality to manually exploit the identified vulnerabilities and enables the tester to see the real time impact of the attack.

2. Background Research

There is lot of research is done over the web application security and vulnerabilities. The Web Application assessment process is a comprehensive application audit performed through different means including manual penetration testing, code review, automated security assessment tools, and application specific architecture review.

The ultimate goal of the security audit is to identify the application logic, configuration, and software coding flaws jeopardize the security of the application.

There are many automated security tools are available for web application assessment. Mainly the automation process is the use of tools and strategies that reduce human involvement or interaction in unskilled repetitive or redundant tasks.
There are many security assessment tools are available for software assessment. The security assessment tools are designed to discover the vulnerabilities present in your applications. Automated tools also perform the scanning and perform the penetration testing against the identified issues.

3. Proposed System

The Security Assessment Automation framework is designed to discover security holes in the web application that an attacker would likely abuse to gain illicit access to your systems and data. The System looks for multiple vulnerabilities through different types of assessments including SQL injection cross site scripting and weak password and many more.
Application can be used to perform scanning for web and application vulnerabilities and to perform penetration testing

against the identified issues. The Impact and Mitigation suggestions are then provided for each weakness and can be used to increase the security of the web server or application being tested.

The proposed system will overcome the different disadvantages of the existing systems. Here we are listing the main key features of the system:

- Proposed system performs the Web Application Security Assessment based on the process and methodologies from OWASP Web application Testing Guide and the OWASP ASVS 2009 standard.
- Proposed solution will store the issues that are false positive and will ask the user to include or exclude them.
- It will provide the automation of reports after security assessment and penetration testing.
- Proposed system provides a system with exclusion functionality for certain directory's files those are not required for security assessment. We can exclude the directories such as libraries, jars and tools directories from the data to be scanned. We can customize the directories as per requirement.
- The proposed system generates the issues that are severe as well as minor so that all the issues will be dealt with the same importance and it will remove the issues that are false positive.
- Proposed systems will have a GUI for adding the new vulnerabilities and will be provided with the functionality of auto updating of issues.
- System reduces the labor work that was needed in case of existing systems. The proposed system is planned to be a web application so that it can be accessed from local machine also. Proposed system will contain simplified GUI and user friendly environment so we can easily operate it.
- The proposed system will not leak the data or source code to any third party as it has designed its security mechanism with user authentication.
- Proposed system will not be much expensive as we have designed the system using the freeware and inbuilt tools.
- The tools that we used are validated by OWASP organization so that are not banned by any agency.
- The different tools are used for testing the system for different types of assessment and we are going to use it as a standard for the testing the applications.
- Also the proposed solution will contain the all types of assessments like White Box, black box and network based in the same system. Also it adds some more strength to the system as we have added the reporting part in the proposed system.

4. System Description

The system comprises of different vulnerability assessment tool, reporting tools and data management tools.
System contains following components:

1. Dashboard
2. White-Box Security Assessment System
3. Black-Box Security Assessment System
4. Reporting Sub-System
5. Support Sub-System
6. Knowledge Base

Dashboard: It contains information of all applications with their respective subsystems, and subsystem wise vulnerability assessment details like what tools are used for particular assessment and who are the team members. It will help managerial persons to keep track of system. It also contains the client interactions and web application inputs to the assessing system.
It has following components:

Client Interaction:
This will contain the different agreements and policies. Also there will be data collection required for carrying out the assessment securely.

Web Application input
In this component the data shared for web application is checked for validity and is processed to give it as an input to dash board.

Project Management
Project data for all those projects whose entry is done in the system will be displayed on the dash board. Also the reports and assessment details are shown in this section of the system.

Statistics Data
This tab will notify the users regarding the number of assessments, issues found in assessment, false positives found in the system, etc.

The dashboard systems is mainly a control systems that checks the all the activities during vulnerability assessments of application till delivery of the reports. Access control functionality is provided while providing access to the Dashboard system, so depending upon the access user will get the functionality. E.g. Admin user gets all rights to-do but the client or other person such as developer, tester are have limited access to the root functionality.

White-Box Security Assessment System: This sub-system is responsible for white box code assessment and to Identify vulnerabilities in very early stage of SDLC. As stated earlier, it will identify vulnerabilities in early stage of SDLC; it will help in improving development standards and practices and will make system less vulnerable at development stage itself. This may have static or dynamic code analysis according to the tools used.

Automated White-Box Security Assessment system
Particularly the Automated assessment system for white-box security assessment will integrate the different tools and vulnerability assessment of application will be carried out. Depending on functionality the tools determines the vulnerable points in the application.

Manual White-Box Security Assessment System
This will take manual input for the issues that need to be considered for particular assessment of the system.
The issues that are missing from the assessment due to their false positive nature will get added in the system through this way.

Onwards after analyzing the report, report parsing is done in Repository subsystem.

Black-Box Security Assessment System: In this system there will be integration of penetration testing tools. These tools will test against security issues to find different vulnerabilities. The identified vulnerabilities are categorized into OWASP top vulnerabilities with that tool's also provide testing of Databases, operating systems, websites, networks assessment etc.

Automated Black-box Security Assessment System
Automated Black-box security assessment works as a combination of different units such as scanning tools and penetration testing tools. Those tools will check for the vulnerabilities and will try penetrating into the system in order to show the flaws that will breach the system.

Manual Black-box Security Assessment System
This component will manually test for penetration in different ways into the system and will add those issues that occurred as true positive. This process is efficient in checking for new issues that are found while penetration testing.

Same process is takes place & vulnerability report's information gets provided to the Repository system.

Reporting Sub-System:

Repository System: The report from different vulnerability scanner systems are gets parse before the repository system and all vulnerability information details are gets store to repository system. Depending on the requirements all information or selected records are fetching from the repository system and used throughout the system.

The Repository system contains the information gathered from different tools & collectively store in database repository.

*Automated Report Generation System***:** Final vulnerability assessment report is generated automatically as the user request for it using some key attributes to the assessment. The regenerated report is again store into a repository system & all controlling functionality is then again forwarded to the Dashboard system. Generated report contains whole analysis about the vulnerability assessment like,

- No of vulnerability with respect to severity
- No. of vulnerability with respect to category
- No. of vulnerability with respect to user role

Details description about vulnerable point
After the Report generation activity, final report is deliverable part of the Dashboard.
*FP Removal System***:** A static code analysis tool will often produce false positive results where the tool reports a possible vulnerability that in fact is not. This often occurs because the tool cannot be sure of the integrity and security of data as it flows through the application from input to output.

False positive results might be reported when analyzing an application that interacts with closed source components or external systems because without the source code it is impossible to trace the flow of data in the external system and hence ensure the integrity and security of the data.

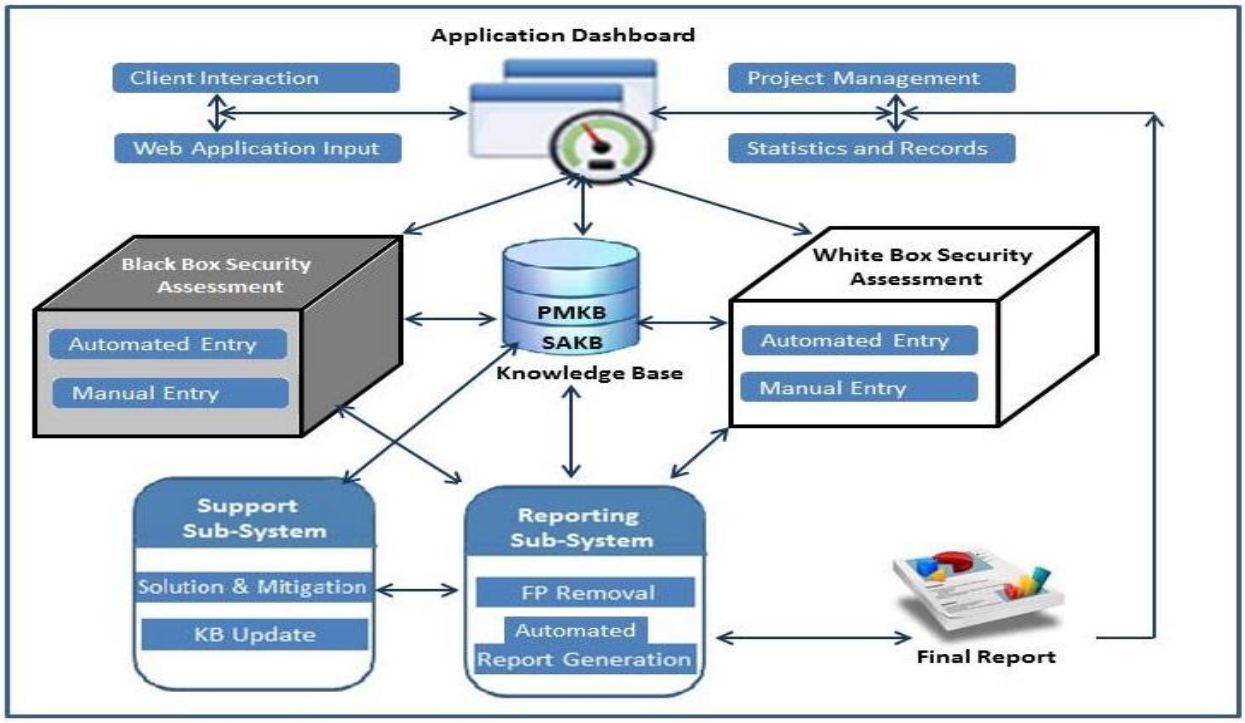

Fig.1: Security Assessment Automation Framework: Block diagram

Support Sub-System:

This system will deal with the solution and mitigation and support for the scanned web application. Support subsystem helps the user to maintain the data and find the solution on the issues that are raised during the security assessment.
It has following Components:
Solution & Mitigation:

This component will deal with the occurred issues and will mitigate with them to find the solution. This system is used to deal with the issues that are not resolvable by any tool or are false positive need to be removed from the final report.

KB Update:
Knowledge base update is done for newly occurred issues. The cause of occurrence and remedy on that issue will be done in knowledge base update. KB update helps the system to be updated with newly occurring issues and will help to mitigate those issues.

Knowledge Base (KB):
It is the central database where all the data related to the different issues in terms of name, severity, tool, description and status is stored. It also contains the details of the data that are stored for particular project as a part of product info and its assessment info.
There are two parts of KB:

Project Management KB:
Project related data, project requirements, application data, types of users and number of users, different policies and rules those are used in the proposed system are stored in the project management knowledge base. This knowledge base is updated whenever we add new project or new product under same project.

Security Assessment KB:
The issues, assessment details, reports and different statistics those are stored in the security assessment knowledge base. Security assessment KB will help the user to formulate its report according the different aspects like severity, tool used, description and code found vulnerable.

5. System Control flow diagram

While interacting with the proposed system following steps are carried out:

- Determine Scope
  Firstly we have to check for scope of the application that you are going to scan. The application is on which OS, the language of the application, the tools needed to access the code of application.

- Information gathering
  We need to get the information of the application such as design environment, target system.

- Vulnerability Detection
  This step identifies the vulnerabilities present in the given web application.

- Root Cause Analysis
  What are the causes for vulnerability to occur will be decided in this step.

- Target Penetration Testing
  Future impact on the system affected by the found vulnerabilities will be estimated by target penetration testing.

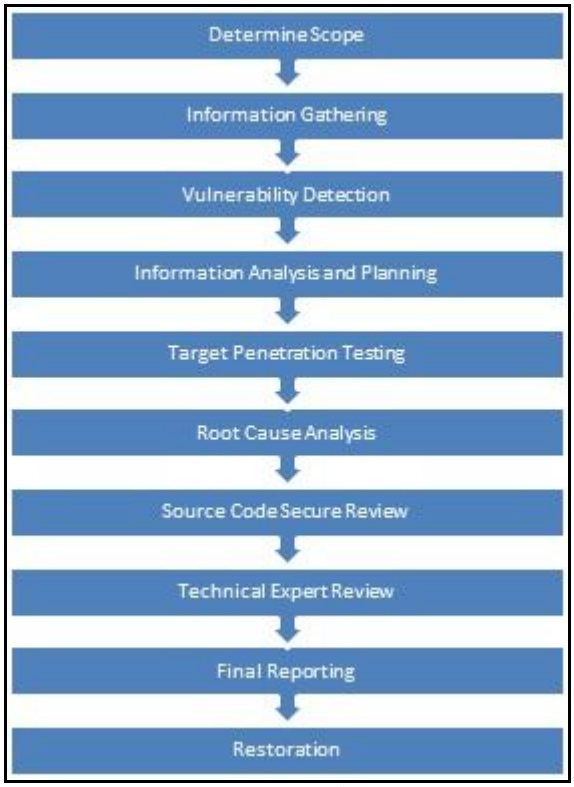

Fig.2: System operation flow diagram

- Information analysis and Planning
  Analyze the information gathered after scan and plan accordingly to remove the vulnerabilities and deal with the other issues .We are using different data structures to store the generated result. The System requirements are analyzed accordingly the system is needed.

- Source Code Secure Review
  Source code will be tested in secure environment for occurring vulnerabilities or issues and issues are found. Then define them according to the severity of the code as high to low vulnerable. Also we need to set some vulnerabilities and their description.

- Technical Expert Review
  The issues found during an assessment are analyzed by using the false positive removal system. The vulnerabilities that are less vulnerable must be set to Low severity. In this step we will work removal of the false positive issues and rearranging the report.

- Final Reporting
  Final report will consist of the issues that we actually want in the report. Other issues are just removed from the report or they are just disabled for the scan.
  The reporting subsystem works on the report generation part.

- Restoration
  The errors and other findings are removed in this step. Also the data which is not producing any report is reanalyzed and checked for the issues.

6. Advantages and limitations

- Identifies exactly where vulnerabilities exist and why/how they occurred
- Easier to begin remediation because the exact location of the vulnerabilities has been identified
- It will not generate large number of false positives and have the mitigation on the vulnerability with root cause of it in source code.
- Provides feedback on environmental components that affect the security of an application and will provide the remedial solution on it.
- The growing knowledge base of vulnerabilities makes system more accurate for identifying the false positives.
- Likely user friendly system –security staff as well as developers can easily understand the system.
- Currently this system is a single threaded model.

7. Conclusion

The proposed system will scan the given code and will generate some set of issues. The issues generated will be reviewed. The review will consist of checking the correct issues for which all the fields are generated. Then it will check for the issues which are redundant and will make them as redundant.

The system will assess for the issues which are false positive and will remove or will change the status as disabled. Also the system will be having a template for report generation and will generate the report. In report automation module the report to be generated is passed with the data and will connect with the different dataset and will allow it to generate the report.

The system is having immense importance in assessment for security issues. The system will generate the issues that are of security concern so will be dealt all times. The system will of course get information for generation report according to security perspective.

References

[1] LaShanda Dukes, Xiaohong Yuan, Francis Akowuah, "A Case Study on Web Application Security Testing with Tools and Manual Testing", Southeastcon, 2013 Proceedings of IEEE,4-7 April 2013, ISBN: 978-1-4799-0052-7

[2] Nuno Teodoro, Carlos Serrao, "Web Application Security", Information Society (i-Society), 2011 International Conference on 27-29 June 2011, ISBN: 978-1-61284-148-9.

[3] Shahriar H. and Zulkernine M., " Automatic Testing of Program Security Vulnerabilities", proceeding of 33rd International Conference of Computer Software and Applications (COMPSAC), 2009 IEEE, pages 550-555.

[4] Dhanya Pramod, "A Study of Various Approaches to Assess and Provide Web based Application Security", IJIMT, Vol. 2, No. 1, February, 2011, ISSN: 2010-0248.

[5] R. Kumar, S. K. Pandey, S. I. Ahson, "Security in Coding Phase of SDLC", Wireless Communication and Sensor Networks, 2007. WCSN '07. Third International Conference on 13-15 Dec. 2007, IEEE, ISBN: 978-1-4244-1877-0

[6] N. Jovanovic, C. Kruegel, and E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities (short paper)," in 2006 IEEE Symposium on Security and Privacy, 2006, pp. 258-263

[7] Open Source Vulnerability Database, http://osvdb.org.

[8] https://www.owasp.org/index.php/Appendix_A:_Testing_Tools

[9] http://www.exforsys.com/tutorials/testing/automated-testing-advantages-disadvantages-and-guidelines.html

[10] https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

Author Profile

Gopal R. Choudhari received the Bachelor of Engineering in Computer Science from University of Pune, Pune- Maharashtra State of India and pursuing Masters of Technology in Information Technology from Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded- Maharashtra State of India. His research interest lies in Network and Information Security. He is currently working as a research scholar in IT department at SGGS IE & T, Nanded-Maharashtra state of India.

Prof Madhav V. Vaidya is currently working as Assistant Professor in Information Technology department at Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded- Maharashtra State of India. Currently he is pursuing his Ph.D. from SRTMU, Nanded-Maharashtra State of India. His research interest lies in computer network, digital image processing and pattern recognition.