An Approach To Automate Security Assessment For Web Applications

Authors

Sushilkumar Yadav\textsuperscript{1}, Chandrakant Navdeti\textsuperscript{2}

\textsuperscript{1,2}Department of Information Technology, SGGSIET, Nanded, Maharashtra, India

\textsuperscript{1}sushilyadav1234@gmail.com
\textsuperscript{2}navdetichandrakant@gmail.com

ABSTRACT-

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 scope of assessments, tools to use and the methodology for conducting assessments. It also generates 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.

INTRODUCTION

Test Automation

Manual testing is carried out by individual person who is working on by checking with each and every aspect of the application. The programmer or tester will try with different methodologies and different techniques, checking with expected result and observed scenarios. We have to do the manual tests for every change so need to do them again and again.

Automation security assessment tool will check for each and every feature and will store them in database for future reference so that we can compare the system behavior for different inputs or changes to the system. Once we are done with the design of the automation tool it is the only requirement for user to carry out the assessment and check with the found vulnerabilities. So it is the need of today to have the system to be automated to have tests as well as reporting accordingly.
Security Assessment
There are systems with different layers and different configurations like client-server, server-server configurations and single server and multiple client configuration. So there will be so many complex networks for to hackers or criminal organizations to breach into it. It is not the leman who is checking with your system. So we need to concentrate on the security of the application.

Automated Assessment Tools
These tools are used to generate the report of different vulnerabilities occurring in the system through the automation process for scanning the application.

We need to test the application manually or by using the testing automation toll in order to check the application secure to use. We check the security of the application by checking with certain parameters which are responsible for your application’s security. We can check the security of the application by performing penetration testing on the application or by checking the application using the tools that we use for security tests. Now a day there is high quality security testing tools that do test in a quick succession and give the result to you in the format you want like pdf, doc, etc.

Limitations of Previous Automated Tools
– The automation tools were only capable of finding the technical issues, vulnerabilities.
- The issues that were not the system vulnerability or tool vulnerability cannot be found as they were not easily detectable.
– The application is vulnerable or not to security is not decided by the tools.
– They require the skilled people to automate or operate them.

Security Testing & Risk Assessment
We require checking for the security of the application through the severity of the issues that are found in certain assessment for security of an application. We need to update the knowledge base and backend database in order to check with the newly found issues so that it will be efficient tool to use. We also need to take into consideration the risk that we are going to be in if the assessment is not performed correctly.

Functional Security Requirement
We need to concentrate on the security needed by certain functional section of the application such as if it is a financial calculation application we need to take care of the debit as well as credit functions of those applications. These are firstly check in order to offer safer way to use the application. We can check the functional requirement by checking them with existing conditions and future impact on it. the system is tested for various validation norms and rules. We do have security requirement for certain financial or online transaction that are doing through that application then we need to concentrate on the security of that application.

Secure confidential data while reading and while storing it in database.
Encrypt the important and no to be disclosed information from the users. (e.g., authentication keys and recovery paths.)

Provide recovery system for the users in order to protect the account from hackers.

Do not give the user info about the login attempt that is done by a specific user.

Give permission to the user for certain regular expression passwords that are not easily guessable.

Allow a user to change the password by using his recovery email address or the phone number registered or through the security check and questions to guess the password.

The password for a user will be changed or recovered only after certain security checks like recovery email address and valid and registered user name. The password that is assigned first time should be one time password and should be changed after first login. The user will be provided with a link in the authenticated email account as well as certain security checks such as security questions and recovery unique checks so that it will lead the user to set his password and the application will be able to track on the security of the certain user’s account as well as whole application security.

Why Vulnerability Assessment

The vulnerability assessment is carried out to assess the given application for certain norms and security procedures in order to attain the security objectives.

Find the vulnerable point where the attacker can easily breach into.

Give acknowledge of the issues in the application.

Check with the security framework as well as configuration that application is in.

PROPOSED SOLUTION

The proposed system will minimize as well as remove the different shortcomings of the existing systems.

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 system with exclusion functionality for certain directories 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 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 updation of issues.
System reduces the labor work that was needed in case of existing systems. 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 proposed solution will contain different combination of tools that will be used for assessment of different applications.

Also the proposed solution will contain the all types of assessments like WhiteBox 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.

**SYSTEM DESCRIPTION**

The system comprises of different vulnerability assessment tool, reporting tools and data management tools.

System contains following components:

- **Dashboard**
- **White-Box Security Assessment System**
- **Black-Box Security Assessment System**
- **Reporting Sub-System**
- **Support Sub-System**
- **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.

![Security Assessment Automation Framework: Block diagram](Fig.1)

*Sushil kumar Yadav, Chandrakant Navdeti IJSRE Volume 2 Issue 7 July 2014*
**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 that we are in real not considering as vulnerability. This will occur as the tool will not be able to check for the logical flaws.
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.

**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.

**SYSTEM FLOW DIAGRAM AND CONTROL FLOW STEPS**

**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 will identify the vulnerabilities that are there 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.

ADVANTAGES AND LIMITATIONS

Advantages
– detects the occurrence of the certain issues correctly and states the reason for the vulnerability occurrence.
– gives the information of the code whether it is from source code or any other place
– this will make it easy to start with the remediation for issues.
– It will not generate large number of false positives and will try to mitigate those issues in early stages.

– Provides feedback on environmental components that affect the security of an application and will provide the remedial solution on it.
– Likely user friendly system – security staff as well as developers can easily understand the system.

Limitations
– It takes time for black box security assessment in highly complex applications.
– Currently this system is a single threaded model.

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


