Web Application Penetration Testing Tool

Yugabdh Pashte, Yash Patel, Ruthvik Shetty, Asst. Prof.Aju Palleri
Information Technology Department
Pillai College of Engineering, Mumbai University
New Panvel, Raigad, Maharashtra, India

***Abstract* - In the current era, Digitization has taken day-to-day utilities starting from a cab to a glossary on the internet. All the service providers heavily leverage IT and IT Services. Web Application plays a significant role in providing these services. While Digital opens infinite opportunities to increase business and enhance delivery, it also exposes the business to an unseen world of cyber-attacks. To prevent the business from digital dysfunctioning, organizations pro-actively and continuously perform Vulnerability Assessments & Penetration Tests on their IT Assets (i.e. Web Applications, Network Devices, Servers, Security Devices, etc.). We propose a framework that captures the footprint of an organization, useful for the information gathering phase during penetration testing called Reconnaissance. Reconnaissance refers to the preparatory phase where a penetration tester seeks to gather as much information as possible about a target of evaluation before launching a penetration test. Our Python tool helps in locating and saving organization-specific data. Such data repositories will help in the vulnerability assessment of an organization. This will include designing a user-friendly graphical user interface. In the end, it will generate a report of vulnerability assessment.**

Keywords -- Vulnerability, Penetration, Webapp, Assessment, Reconnaissance, Footprinting.

I. INTRODUCTION

A. Fundamentals

Web Application plays a significant role in providing IT services. While the digital world opens infinite opportunities to increase business and enhance delivery, it also exposes the business to an unseen world of cyber-attacks. To prevent the business from digital dysfunctioning, organizations pro-actively and continuously perform Vulnerability Assessments & Penetration Tests on their IT Assets such as Web Applications, Network Devices, Servers. We propose a framework that captures the footprint of an organization, useful for the information gathering phase during penetration testing called Reconnaissance. Reconnaissance refers to the preparatory phase where a penetration tester seeks to gather as much information as possible about a target of evaluation before launching a penetration test.

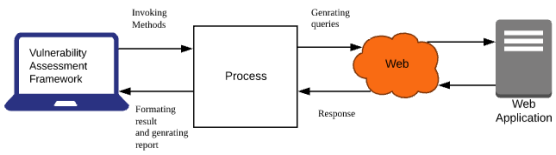

Fig 1. Vulnerability assessment framework introduction

B. Objectives

When a web application is designed and deployed it might be having vulnerabilities in it. Testing or more specifically White box testing is done before deploying an application in the production environment. While doing this job various tools and frameworks are to be used and corresponding reports are to be generated. The main objective is to build a tool that will be easy to use, integrating various other open-source tools and automating them, and generating the end report of the assessment.The main objectives to be achieved:

- To understand top 10 vulnerabilities mentioned on OWASP and how to exploit them.
- To understand frameworks like Octave.
- Developing python modules to carry out tasks of assessment.
- Implementing UI for above components.
- Test tool on DVWA and get feedback.
- Iterate and improve according to feedback.
- Generating report with help of modules and queries to DVWA.

C. Scope

Our project goal is to find and give a detailed report of vulnerabilities in the web application of IT organizations/ companies. Our project requirements are DVWA for testing, web browsers like Firefox, system installed Linux OS, Python environment, database to store reports of previous vulnerability assessment tests. Our project deliverables are Python modules of tools, UI, report module. List of users using our framework would be Penetration Testers, Cyber Security Researchers,etc. Our framework features include user-friendly UI, automating assessment tasks, detailed report generation,etc.

II. LITERATURE SURVEY

In this survey, the relevant techniques in the literature are reviewed. It describes various techniques currently being used. We have reviewed four research papers in the domain of vulnerability and network security.

A. Literature review techniques

The techniques in this category are adapted to the individual needs, interests and preferences of the user or society. They are tools for suggesting items to users in this domain. Various techniques in this category are listed here. These techniques have various advantages and are used extensively in the literature.

B. Technique One

Administrators need to perform vulnerability scans periodically which helps them to uncover shortcomings of network security that can lead to devices or information being compromised or destroyed by exploits. Different tools have different approaches and outputs integrating the output and making it easy to understand. In this we are considering

NMAP & OpenVAS. On the basis of impediments of NMAP and OpenVAS, another tool is developed which holds the best of both devices alongside overcoming a few drawbacks. Further vulnerabilities scanning is performed by comparing the information obtained from a network scan to a database of vulnerability signatures to produce a list of vulnerabilities that are presumably present in the network. Along with performing network scanning and vulnerability assessment, an auto-scan mechanism is also added in a new tool to test devices when they are compromised. In other words, network mapping, vulnerabilities and configuration faults in the network are shown in various formats.

C. *Technique Two*

This project provides flexibility because of modular code. Tool is developed by dividing it into modules hence this makes the tool open to future development. This tool is developed to test against the web application with HTTPS hence with SSL certification only. In this technique penetration testing is also done. Net Nirikshak 1.0, Samurai framework, Safe3 scanner, Websecurify and SQLmap are automated using Python. Heterogeneous output of these tools is integrated and a report is generated. Manual testing of the vulnerabilities of the application was successfully performed. Conversion of the local server from HTTP to HTTPS was successfully done by creating a self-signed certificate. An automated tool for the vulnerability assessment of HTTPS web applications was successfully developed.The tool is currently capable of performing: Whois Scan, Basic Port Scanning, Certificate Verification, SSL Connection with the Server, Grabbing of HTTP/HTTPS links, SQLI Vulnerability Detection. The tool has been well automated, hence it does not demand any special expertise from its users, unlike other tools.

D. *Technique Three*

Complexity of systems and the number of systems are increasing every day. This leads to more and more vulnerabilities in Systems. Attackers use these vulnerabilities to exploit the victim's system. In this paper we proved Vulnerability Assessment and Penetration Testing as a Cyber defence technology, how we can provide active cyber defence using Vulnerability Assessment and Penetration Testing. We described the complete life cycle of Vulnerability Assessment and Penetration Testing on systems or networks and proactive action taken to resolve that vulnerability and stop possible attacks. From this paper we understand prevalent Vulnerability assessment techniques and some famous premium/open source VAPT tools. We have described the complete process of how to use Vulnerability Assessment and Penetration Testing as a powerful Cyber Defence Technology.

TABLE I. LITERATURE SUMMARY

Sr no.	Summary of literature survey		
	Paper	***Advantages***	***Disadvantages***
1	Network Scanning & Vulnerability Assessment with Report Generation by Nikita Y Jhala	Integrated two tools and output is made easy to understand	Tools still uses original modules on surface code
2	An Automated tool for Vulnerability Assessment of HTTPS Web Applications by Anand Ramesh	It is developed in such a way that it is open to further development and new functionalities can be easily added in the form of modules	This tool doesn't work with applications that does not have SSL certifications
3	Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology by Jai Narayan Goel, BM Mehtreb	The survey shows combined techniques for improved performance.	It improves the user preferences for suggesting items to users.

III. PROPOSED SYSTEM

A. *Overview*

In today's cybersecurity world, for doing vulnerability assessment different methodologies and tools are available. These tools have specific applications and help in exploring a particular scope. There are tools to map networks, Identify underlying system architecture, visualize different nodes in business architecture, and so on. When a system is forked with this tool output is generated and this output is to be analyzed by a security expert and the end report is generated. This may lead to confusion as different tools are used to work around this process and output is heterogeneous. We are proposing a system or rather a framework to solve these issues. In this proposed framework we are going to automate various tasks in vulnerability assessment using python by integrating these existing scanning tools and provide combined classified results which will be easy to understand and hence report generation will be simplified. A unique and easy to understand interface will be provided to interact with which will make our system easy to use.

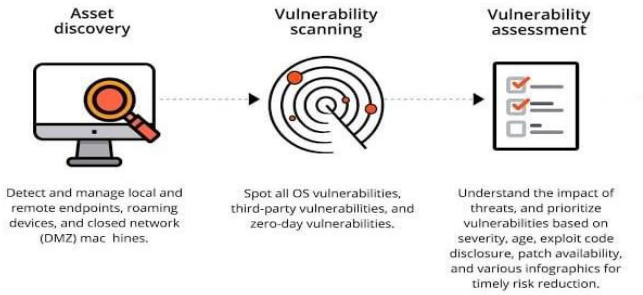

Fig 2. Vulnerability Assessment Framework proposed system

B. *Existing System Tools*

- OpenVAS: This is an open source tool serving as a central service that provides vulnerability assessment tools for both vulnerability scanning and vulnerability management. OpenVAS supports different operating systems.The scan engine of OpenVAS is constantly updated with the Network Vulnerability Tests OpenVAS scanner is a complete vulnerability assessment tool identifying issues related to security in the servers and other devices of the network OpenVAS services are free of cost and are usually licensed under GNU General Public License (GPL).

- Nikto: Nikto is a greatly admired and open source web scanner employed for assessing the probable issues and vulnerabilities. It is also used for verifying whether the server versions are outdated, and also checks for any particular problem that affects the functioning of the server. Nikto is used to perform a variety of tests on web servers in order to scan different items like a few hazardous files or programs. It is not considered as a quiet tool and is used to test a web server in the least possible time. It is used for scanning different protocols like HTTPS, HTTPd, HTTP etc. This tool allows scanning multiple ports of a specific server.
- Nmap: Nmap is a handy addition to the value-added reseller (VAR) and consultants' vulnerability assessment toolbox. Nmap performs a SYN Scan, which works against any compliant TCP stack, rather than depending on idiosyncrasies of specific platforms. It can be used to quickly scan thousands of ports, and it allows clear, reliable differentiation between ports in open, closed and filtered states. If Nmap is compiled with OpenSSL support, it can even connect to an SSL server to deduce the service listening behind that encryption layer. Another advantage of running version detection is that Nmap will try to get a response from TCP and UDP ports that a simple port scan can't determine are open or filtered, and Nmap will change the state to open if it succeeds.

TABLE II. SUMMARY OF EXISTING SYSTEM TOOLS

Category	Existing System Tools	
	Tool	***Description***
Host-Based	Metasploit	An open-source platform for developing, testing and exploiting code.
Network-Based	Cisco Secure Scanner	It is developed in such a way that it is open to further development and new functionalities can be easily added in the form of modules
	Wireshark	Open Source Network Protocol Analyzer for Linux and Windows.
	Nmap	Free Open Source utility for security auditing.
	Nessus	Agentless auditing, Reporting and patch management integration.
Database-Based	SQL lite	Dictionary Attack tool door for SQL server.
	Secure Auditor	Enable users to perform enumeration, scanning, auditing, and penetration testing and forensic on OS.
	DB-scan	Detection of Trojan of a database, detecting hidden Trojan by baseline scanning.

C. Proposed System

As per the previously mentioned documents, there are tons of tools or micro tools available to carry out every task. These tools are written in different languages and produce heterogeneous output. This creates confusion while analyzing these outputs and generating reports.We propose a method to solve these issues in our method. There are top 10 vulnerability lists available on OWASP also information to exploit them is available and well documented. We are going to follow this and implement it in our methodology. There are many frameworks available. These frameworks define the way to approach vulnerability assessment and penetration testing. We are going to follow one of this framework named Octave to understand how these assessments are carried out in the real world and design our workflow accordingly. This stage is very important to maintain standards as it will help security experts without causing any issues while performing tests on the system. We are going to perform the reconnaissance stage as it is considered as one of the important steps in any pen-testing session there are tools available like NMAP, Wireshark, and different APIs are provided by different developers. We are going to use these tools and available modules developed for these tools and scrap output generated. This output is then formatted according to needs and presented to the user. To scan various vulnerabilities in a web application Nikto and OpenVAS are some top-rated tools available in open source. Modules of this tool are then used to scrap output and a separate module is developed to integrate it with our framework.

D. Implementation Details

Implementation of our project is done with the help of the Python language. As python has a huge open source community and a bunch of modules to work around we have chosen Python for programming this tool. We are having huge dependencies from other software and different languages. The implementation approach will be discussed below.

E. Implementation Details

Implementation of our project is done with the help of the Python language. As python has a huge open source community and a bunch of modules to work around we have chosen Python for programming this tool. We are having huge dependencies from other software and different languages. The implementation approach will be discussed below.

F. Understanding dependency tree

The dependency tree is important as it is always needed to be satisfied while development and installing our tool. Hence we are going to maintain dependencies for our tool from different tools and as per our programming needs

G. Developing modules

We are following the modular approach in our project. This will give us freedom over the development process as each module will be fully functional and deployable parts of code. This also makes the project upgradable in the future. These modules will be based on the existing tool and modules available in the Python environment. The integration of these modules needs to be done in this stage of development which will help to call the required module when needed.

H. Developing GUI

GUI is an important part of our framework. Even though many tools available up till now are more command-line oriented and many security experts are familiar to use such tools we are considering using GUI for interactions. As a

GUI it is easy to understand and navigate. We are going to use a framework for this purpose like QT Creator or Flask. This will allow the rapid development of GUI. Once the user interacts with the framework previously designed modules in python will be called or invoked.

I. Designing report generator

Report generation is the last but important phase of vulnerability assessment. In this detected vulnerabilities are ranked, flagged, and classified this will help security person or pentester to consult proper authority about the generated report or this report can be imported and be part of the bigger picture.

IV. BLOCK DIAGRAM

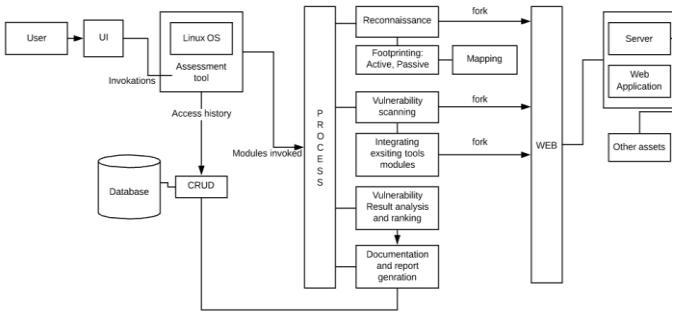

Fig 3. Block diagram of Tool

- Assessment tool: This tool will consist of the following components: different process modules, database and user interface. This tool will invoke different processes.
- Database: To store reports of previously generated reports and analysis. Thisdata can be accessed, stored, deleted using CRUD.
- Process block: This block will invoke and maintain different module calls. As a modular approach is used, the tool will be open for further development.
- Reconnaissance: This stage is for generating recon about web applications like mapping different assets available for application, discovering hidden nodes. This is done with help of footprinting modules available and designed by tool.
- Footprinting: Footprinting stage is important to map organisation resources to later iterate over them two methods will be used mainly as passive and active footprinting.
- Vulnerability scanning: In this stage vulnerabilities will be detected by using various different tools by integrated modules . Later detected vulnerabilities will be organised and ranked.
- Documentation: In this stage detected and ranked vulnerabilities will be documented according to risk level and this generated report is referred by security researchers and hence stored and retrieved from the database using CRUD.

V. HARDWARE AND SOFTWARE SPECIFICATIONS

The experiment setup is carried out on a computer system which has the different hardware and software specifications as given in Table III and Table IV respectively.

TABLE III. HARDWARE DETAILS

Processor	2 GHz Intel
HDD	180 GB
RAM	2 GB
NIC	Any

TABLE IV. SOFTWARE DETAILS

Operating System	Linux OS with GUI
Programming Language	Python, SQL
Database	MySQL
Code Editor	Visual studios

VI. USE CASE DIAGRAM

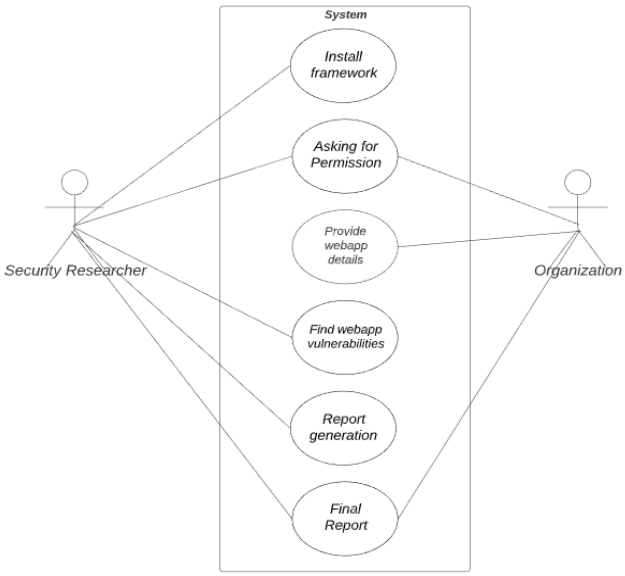

Fig 4. Use Case Diagram

VII. SUMMARY

In this paper, we tried to explain the methodology we are going to apply while developing our framework. We analyzed what are currently tools available to a pentester and found out the advantages and disadvantages of using them standalone and how they generate output. We understood from our literature survey that how these tools can be combined together and how the output of these tools can be integrated and served to the user as one. Then we analyzed the pros and cons to put this method in action and proposed our own method. We discussed how this tool will work by looking at UML diagrams. Implementation details of this framework are discussed in brief as it can be considered as our road map to develop this tool.

VIII. FUTURE SCOPE

This application framework is following a modular approach from start to finish hence we are open to changes also adding new functionality to the project. At any level, we can introduce new modules. We are planning to add more footprinting methods to applications to assess web applications better. GUI can be improved further so it can be used by a beginner and still he/she will carry out the task.

ACKNOWLEDGMENT

We would like to show our gratitude and thanks to Dr Sandeep Joshi, Principal, Pillai College of Engineering for giving us a good guideline for the project throughout numerous consultations. The help and guidance given by him from time to time gave us the motivation to complete the project. We also take this opportunity to express a deep sense of gratitude to Dr Satishkumar Verma, HOD, IT department, for his cordial support, valuable information and guidance, which helped in completing this task through various stages. We are grateful for the cooperation during the completion of our task. We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Aju Palleri for introducing us to the methodology of work and her exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. We would like to thank all the people for their help indirectly and directly to complete our task.

REFERENCES

[1] Nikita Jhala (Nirma University), "Network Scanning & Vulnerability Assessment with Report Generation".

[2] Octave framework.

[3] OWASP Top Ten vulnerability.

[4] Top vulnerability scanning tools.

[5] Nmap network scanning manual.

[6] Anand Ramesh (Institute for Development and Research in Banking Technology), "AN AUTOMATED TOOL FOR VULNERABILITY ASSESSMENT OF HTTPS WEB APPLICATIONS".

[7] Front end development Kivy framework.

[8] Django documentation.

[9] Jai Narayan Goel, BM Mehtreb (University of Hyderabad), "Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology".