

WEB APPLICATION AND WEB SERVER FOOTPRINT MAKER AND ANALYZER

U. Sarath kumar², K. Sathish kumar³, M. vignesh⁴, G. Premnath⁴

⁴ Assistant professor., Department of CSE, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai.

¹⁻³ Final Year B.E., Department of CSE, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai.

Abstract:

The Web Application and Web Server Footprint Maker and Analyzer is a toolkit more like a framework to ease in other words automate the tasks of a penetration tester. This framework was designed to help the young startup companies which come into play of product development. As these companies could not afford a hand full of product tester. Our product will fill the pits of lacking testers in the process of penetration testing. This tool automates the RECON phase of penetration testing. It carries out testing for SSL certificates, ciphers and gives insights of server OS and services by analyzing the open port and their responses. Main aim of this project is to help the young startup companies to initiate the first phase of penetration testing that is the RECON phase and generate a detailed report on all possible vulnerabilities that are possible on their website. The URL of the web app under test is given as input to the program. The output will be given as either a csv file or text file based on the need.

Keywords: Phishing Attack, SSL , Vulnerabilities , Port scanning, Server fingerprinting, stealth scan.

I. INTRODUCTION

Security is one of the main user concerns in modern world. This tool will carry out a series of tests to detect the vulnerability of the given website or the webserver. The information from this tool will help in

uncovering the hidden vulnerabilities. Our main focus is on the recon phase and on scanning phase as these are the most time consuming phases of penetration testing. The URL of the web app under test is given as input to the program. The output will be given as either a csv file or text file based on the need. The tool should not produce much traffic as it may cause DOS (Denial of service) attack which is still unintentional so we optimize the scanning procedures. Our tool does the both and gives the result. We also added an SMS notification mechanism to update about the analysis . Consider a web server with windows server 2008 with smb allowed this seems safe but there is an exploit called eternal blue which can exploit this smb. This can be detected based on two characteristic web server OS and the port status of port 445.

II. RELATED WORK

Different approaches can be found in the literature to secure the web application and web server. In authors propose to keep the web application and web server safe from the attacks. There are seven principle areas of infiltration testing execution standard.. They include everything identified with an infiltration test with the underlying correspondence and searching behind a pen test, through the insight social affair

and risk displaying stages where analyzers are running in the background so as to show signs of improvement comprehension of the tried association, through defencelessness research, misuse and post abuse, the specialized security skill for the analyzers come to play and consolidate with the trade comprehension, lastly detailing, which catches the whole procedure, in a way that sounds good to the client and gives the most incentive to it. This rendition can be viewed as a v1.0 as the center components of the standard are set, and have been "street tried" for over a year through the business. A v2.0 is in progress soon, and will give more granular work as far as "levels" as in power levels at which every one of the components of an infiltration test can be performed at. As no pen test resembles another, and testing will go from the more ordinary web application or system test, to an all out red group engagement, said levels will empower an association to characterize how much complexity they anticipate that their foe will display, and empower the analyzer to venture up the power on those territories where the association needs them the most. A portion of the underlying work on "levels" can be found in the knowledge gathering segment. The principle segments characterized by the standard as the reason for infiltration testing execution are Pre-engagement Interactions, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post Exploitation, Reporting.

III. STEALTH PORT SCANNING

The normal connection uses a complete three way handshake to find the open ports and their services which is more resource intensive as 3 way handshake on all 65535 port is similar to attack on the server ie)DOS(Denial-of-service). But this can

be avoided by using syn scan method. A SYN packet is sent by scanner. If the port is considered to be open and SYN/ACK is sent back and the remote end is trying to open a TCP connection. The RST is sent by the scanner to break down the connection before it can be established fully, as preventing the connection attempt appearing in application logs. An RST will be sent, if the port is closed. If it is filtered, the SYN packet will have been dropped and no response will be sent.

IV. MODULUS DESCRIPTION

These are eight modules in the project:

- Port Scanner
- Robots Extractor
- Server Fingerprint Maker
- SSL Examiner
- WHOIS Lookup Maker
- Directory Scanner
- SMS Sender
- Report Maker

1.Port Scanner

Port scanner uses the given url to scan for open ports. It uses a stealth way for scanning called SYN scan. In SYN scan, first a SYN packet is sent to a port if the port responds with a SYN/ACK then the port is open and we will send RST packet instead of ACK which doesn't complete the three way handshake. If RST packet

is sent then the port is closed. If not response is got then the port is filtered. This doesn't make any logs in the server side.

2. Robots Extractor

The ROBOTS extractor extracts the robots.txt file associated with the web application. This file will be used to find the directories and files which is not exposed to search engines and crawlers.

3. Server Fingerprint Maker:

Server fingerprint maker fingerprints the server's operating system and the technologies that are being used such as Apache server running over ubuntu or such. These data are used to find the vulnerabilities related to the server technology or operating system.

4. SSL Examiner:

The SSL examiner examines the SSL certificates that are being used by the web application and exposes the cryptographic methods that are being used with the web application.

5. WHOIS Lookup Maker:

Whois lookup maker does a basic whois lookup at icann.org to get the details of the owner of the web application and the registering authority of the web application.

6. Directory Scanner:

The directory scanner scans the web application's directories based on a wordlist of a huge dataset. The process is multithreaded so that the leftover files with sensitive data get exposed.

7. SMS Sender:

The sms sender sends the status messages to the mobile phone via text message using twilio API. This gives status about every module's process and their level of completion.

7. Report Maker:

The report maker exports the results of the application into a text file into a directory which is used to display on a browser. The port scanning results are exported into a csv file which can be viewed easily on any statistic tools.

V. EXISTING SYSTEM

Automated tools are the most widely used method to protect the web application such as skipfish, acunetix, . In the existing system the tool will get the Location of the website and initiate the target identification phase. In this phase the tool will check the target reachability by using the serving requests over the HTTP protocol. It also fingerprints the web server to identify the technologies that the web server might use. The tools also run a number of security tests against the target website to discover vulnerabilities and the alerts are reported in real-time.

DISADVANTAGES OF EXISTING SYSTEM

- In the existing system only the website are tested and web servers are left vulnerable for further attacks.
- Most of the existing system are highly paid version and does not provide reliable result.
- The existing tools provide high traffic which causes the Server to crash and halt the testing processes for several minutes.
- There is high possibility of false positive rate in the generated test result by the existing system.

VI. PROPOSED SYSTEM

This proposed system presents WASFA, an automated tool for self-protecting web application and server. This tool additionally performs Subdomain enumeration, Header analysis and Directory brute forcing. All the tests are completely automated and requires minimum to no human interaction. The Proposed System performs a stealthy scanning of web server with no sign of detection. The proposed system tests ssl certificates and ciphers used by the web application and detects vulnerabilities on them if any by using the data from previous attacks.

ADVANTAGE OF PROPOSED SYSTEM

- Our framework is an all-in-all tool to automate the task. The tool performs all the major testing.
- In the proposed system we have made our tool to produce less traffic usage compared to other existing Systems.
- The proposed system is a free tool in order to enhance the low and developing companies.

- Provides more technical details about the analysis which can be used for further exploitation of the machine under test.

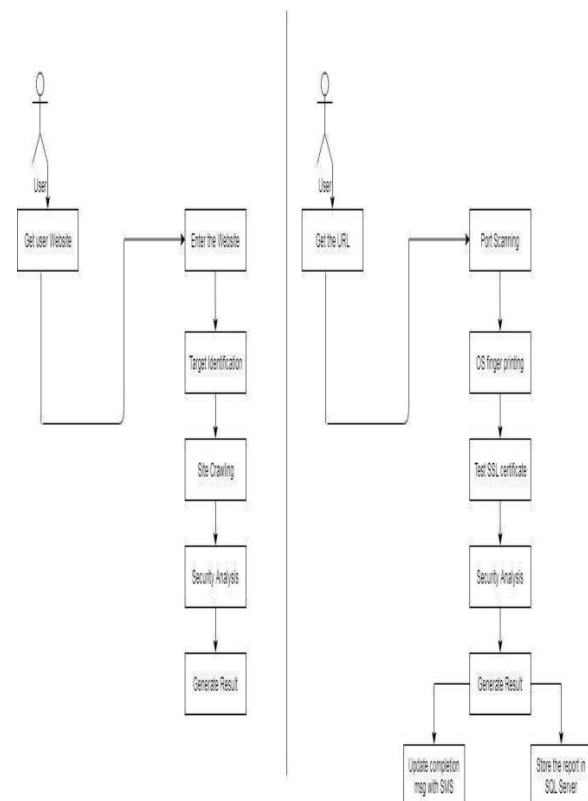


Fig 1. Architecture Diagram

VII. CONCLUSIONS AND FUTURE ENHANCEMENT

In our project, we have designed a tool that will do the initial phase of penetration testing RECON phase. This tool will be very much help for the young start-up companies because they no need to hire lot of Tester because this tool will help them to reduce their testing burden. With a hand full of testers with the help of this tool they can test a massive number of Websites and servers in comparatively less time. Apart from Website testing we also perform Web Server testing which is a greater advantage. We get the URL as an input and provide a detailed report of all possible vulnerabilities. Along with this we also provide an SMS alert to intimate that the testing has been completed and report has been generated.

We are planning to move the entire application stack to a web based User Interface so that one can easily deploy it to a cloud based environment. The introduction of machine learning to the vulnerability testing is not possible now as the dataset for such predictions are not available which may be available in near future. In future a HTML parser which would parse the complete web hierarchy of HTML which would help in make a statistical model of the application which would help in fixing a site wide vulnerability.

IX. REFERENCES

- [1] H. Shahriar and M. Zulkernine, "Mitigating program security vulnerabilities: Approaches and challenges," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 11:1–11:46, Jun. 2012.
- [2] K.-S. Lhee and S. J. Chapin, "Buffer overflow and format string overflow vulnerabilities," *Software: Practice and Experience*, vol. 33, no. 5, pp. 423–460, 2013.
- [3] T. S. Chen and M. W. Cheng, "A New Data Hiding Scheme in Binary Image"(2012)
- [4] M. Shahzad, M. Z. Shafiq, and A. X. Liu, "A large scale exploratory analysis of software vulnerability life cycles," in *ICSE '12. IEEE Press*, 2012, pp. 771–781.
- [5] N. Jovanovic, E. Kirda, and C. Kruegel, "Preventing cross site request forgery attacks," in *Proceedings of the Second International Conference on Security and Privacy in Communication Networks. IEEE Computer Society*, 2012.
- [6] T. Pietraszek and C. V. Berghe, "Defending against injection attacks through context- sensitive string evaluation," in *RAID '06*, 2011, pp. 124–145.
- [7] Z. Su and G. Wassermann, "The essence of command injection attacks in web applications," in *Proceedings of the 33rd ACM Symposium on Principles of Programming Languages*, 2014, pp. 372–382.

