International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 07 | July 2018 www.irjet.net p-ISSN: 2395-0072

Development of Vulnerability Scanner

Arul Louise Jennifer.A[1], R. Senthil Kumaran[2]

[1]Department of Electronics and Communication Engineering, IFET College of Engineering, Villupuram
[2]Associate Professor, Department Of Electronics and Communication Engineering, IFET College of Engineering, Villupuram

---

Abstract -*Nowadays, various security solutions are available that are used to protect a system/network from external threats. Antivirus software, firewalls etc. can only be used to detect viruses, malware available in system due to unauthorized attacks but not help prevent it. To overcome this drawback, a software/protocol is needed to help analyze the vulnerabilities present in the system internally thereby preventing the system from facing unwanted attacks in the future. The idea is to let the scanner parse (analyze) through APIs to recognize which ones might open the framework to threat. A database that stores a list of all known vulnerabilities is used to spot potential issues(i.e.) the more data the scanner has, the more precise its execution that helps resolve the issue quickly. Then, Pen test can be used to determine their exact locations and rectify it.*

***Key Words*: External threats, vulnerabilities, API, Pontes, Antivirus, Firewall**

1. INTRODUCTION

We make use of web application broadly in each field from instruction to business, from distinctive individual to extensive associations. Everything from social data to touchy data is put away on the database. Web applications are basic piece of our life these days. In the meantime, the assaults utilizing web application vulnerabilities and the harm caused by them are expanding. Be that as it may, it is hard to grow totally secure Web applications. Web applications have turned out to be progressively prevalent for conveying security basic administrations. In any case, web applications are additionally exposed to different dangers and assaults, also, consequently various apparatuses, including business ones and open source programming, were created for recognizing web application vulnerabilities. Besides, checking all web application vulnerabilities by hand is troublesome. These are tedious, mistake inclined and exorbitant. Hence a decent robotized instrument to identify Web application vulnerabilities is required. Among top ten web application vulnerabilities code available, infusion assaults are more unsafe. SQL Injection Attacks and Cross Site Scripting are anything but difficult to perform and permit to get to more sensible information. As the data innovation comes by our day by day schedule, the security of the clients utilizing it has turned out to be more essential. In this way, it is required to imagine new and diverse assault identification innovations or procedures which consider all security factors and ensure that clients are safe from dangers of being assaulted. To resolve the above issues, the software must have capacities, for example, a Web crawler, a Web scratching apparatus and a Fuzzing instrument at any rate. A Web Crawler is an Internet bot that methodically peruses the World Wide Web, commonly for the reason of Web ordering; Web scraping is a PC programming strategy for collecting data from online sites; Fuzzing is a product testing strategy, regularly mechanized or on the other hand semi-computerized, that includes giving invalid, unforeseen, or irregular information as inputs of a PC program. These devices are used to crawl a web application, find application layer vulnerabilities and shortcomings, either by controlling HTTP messages or by reviewing them for suspicious qualities.

Generally, there are two principle approaches for recognizing vulnerabilities: white-box testing and black box testing. These strategies go about as intermediary and recognize the weaknesses prior to the code getting executed on server.

1) **White box testing**: In this method we require access to use the source code. By really seeing the source code; we recognize the vulnerabilities in a static fashion. The source code must be reviewed before it sent to the server.
2) **Black Box Testing**: This method doesn't require data about source code. It recognizes the vulnerabilities by passing random sets of input.

Web attacks are a form of vindictive act performed by the attacker to increase unapproved information. Some of the common vulnerabilities are explained below:

1.1 Cross-Site Scripting

Cross-webpage Scripting (XSS) refers to customer side code infusion assault wherein an attacker can execute malicious contents (likewise generally alluded to as a malicious payload) into a legit site or web application. XSS is among the most uncontrolled of web application vulnerabilities and happens when a web application influences utilization of invalidated or decoded client to enter inside the output it generates.
By utilizing XSS, an attacker does not focus on a user specifically. Rather, he/she will misuse a weakness inside a site or web application that the user would visit, basically utilizing the powerless site as a vehicle to convey a vindictive content to the user's program.

While XSS can be exploited inside VBScript, ActiveX and Flash, obviously, the most generally mishandled is JavaScript – principally on the grounds that JavaScript is key to most perusing encounters.

1.2 XPath Injection

XPath Injection is same like SQL Injection, an assault procedure used to abuse applications which are utilized XML reports to store data. At the point when client speak with application by giving html frame input, application shapes xml inquiry by these information. These questions explore XML reports. XPath is a straightforward enlightening dialect which questions the XML record which stores data in the frame of characteristics and hubs and access the delicate data. XPath doesn't give get to control instrument. In this manner anyone can get to finish database (XML archive) by querying it. In this manner XPath Injections may be significantly more hazardous.

1.3 SQL Injection

SQL Injection (SQLi) refers to an infusion assault wherein a hacker can execute malignant SQL commands that control a web application's database server. Since an SQL Injection weakness could influence any site or web application that makes utilization of a SQL-based database, the weakness/vulnerability is one of the most seasoned, most predominant and most unsafe of web application vulnerabilities.

By utilizing SQL Injection vulnerability, given the correct conditions, a hacker can utilize it to sidestep a web application's validation and approval components and recover the data/info of a whole database. SQL Injection can likewise be utilized to include, change and erase records in a database, influencing information trustworthiness.

To such a degree, SQL Injection can give an attacker unapproved access to touchy information including, client information, actually identifiable data (PII), exchange mysteries, protected innovation and other delicate data.

2. RELATED WORKS:

I. W-VST: A Test bed for Evaluating Web Vulnerability Scanner:

In this paper, the proposed method makes use of a propelled perplexity network to appraise the execution of Web weaknesses scanners and afterwards, a practical approach with three principle stages in assessing weakness scanners by considering the deterioration of repetitive weakness alarm. Characterization of the repetitive ready issue in scanner assessment is done in light of two traits, genuine duplication (TD) and false duplication (FD). As needs arises, a Web Vulnerability Scanner Test bed was developed (W-VST). Two examinations have been made to assess its execution. The test results show that the assessment approach can check the execution of scanners and W-VST is productive in device assessment.

II. Evaluation of Web Vulnerability Scanners:

In this paper, evaluation of two open source vulnerability scanners OWASP are done namely: Zed Attack Proxy (OWASP ZAP) and Skip fish using vulnerable web applications Damn Vulnerable Web Application(DVWA) and The Web Application Vulnerability Scanner Evaluation Project (WAVSEP).They run on the OWASP Broken Web Applications Project virtual machine. This system leads to drawbacks such as non-target information like X-Content-Type-Options header missing.

III. Design of Efficient Web Vulnerability Scanner:

This paper makes use of existing web weakness to distinguish various approaches with their preferences and inconveniences. A new method is proposed that productively distinguishes SQL Injection, X path Injection and Cross Site Scripting assaults. The goal is to make strides in discovering the effectiveness of vulnerability scanner while keeping up low false positive and false negative rate. This is done by clustering of response pages generated by server for each injection point while maintaining the state of each page.

3. Proposed System

The proposed system makes use of the following steps:

- A test packet with a “Hello” message with source and destination address listed in it is sent across all ports available in the system that determines the API’s that are susceptible to vulnerabilities.
- The scanner returns with information whether a port is listening or not.[TCP service is mostly used].
- The ports that are found to be active are subjected to a preliminary checking (i.e.) whether a Firewall is strong or not.
- Then, crucial data about the network is collected such as type of OS used, TCP packet, and Service delivery used and so on.
- The collected data is then compared with an in-built vulnerability database containing known vulnerabilities which helps in identifying the threats to the system.
- Finally, a Pen Test is applied to the network to obliterate these threats.

The above mentioned flowchart (Fig 1.) can be implemented by use of the UniScan software that has an in-built database as well as checks for vulnerabilities automatically. UniScan is prevalent Linux based software used for detecting vulnerabilities in a site or web application running on a server. Security Pen

testers find it exceptionally valuable for various sort of specification to gather data about the web application and server and discover vulnerabilities in the web application. UniScan is a Perl based device and is anything but difficult to utilize and which comes preloaded with Kali Linux. The dialog box will be displayed as follows (Fig 2.) where all the available checkboxes are selected and scanning is started.

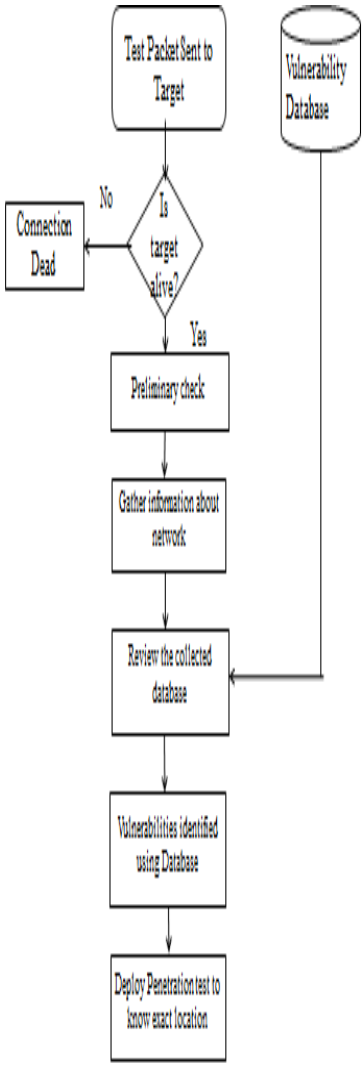

Fig 1. Flowchart for Vulnerability Scanner

Fig 2. UniScan Snapshot

4. CONCLUSIONS

Hence, the use of a vulnerability scanner ensures that the system is safe at all times and secure against the threats and viruses available across networks. In the near future, a vulnerability scanner that automatically updates its in-built database can be created that helps boost the chances of the network to fare well under any attack or threat from an unauthorized network. An ongoing process in vulnerability scanning is done to ensure that a system that is declared as safe may still be unsafe/likely to be attacked and measures are taken to overcome it.

REFERENCES

[1] Yuan-Hsin Tung[1], Shian-Shyong Tseng[*2], Jen-Feng Shih[1], Hwai-Ling Shan[1] "VST: A Testbed for Evaluating Web Vulnerability Scanner", 2014 14th International Conference on Quality Software.

[2] Yuma Makino1 ,Vitaly Klyuev1,Evaluation of Web Vulnerability Scanners, The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 24-26 September 2015, Warsaw, Poland.

[3] http://blog.securityfuse.com/2015/06/scan-website-vulnerabilities-with.html

[4] https://biztechmagazine.com/article/2011/03/pros-and-cons-vulnerability-scanning