International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)

ISSN: 2456-236X

Vol. **04** Issue **02** |**2020**

Creating an effective Web Vulnerabilities Scanner

Suyash Garg[1]

[1]Master Student, Department of MCA, Jain Deemed-to-be University, Bengaluru, Karnataka, India

ABSTRACT

Web Vulnerabilities Scanner is an automated tool used by both black and white hackers to scan the vulnerability in the web application. Web Vulnerabilities Scanner has come under the category of Dynamic Application Security Testing (DAST) it means it scans the web application in an operating state. This paper provides the compression with some well know vulnerability scanner tells which scanner to use on which place and which is not used and the current trend in today's area of web security and this paper does not support any kind of illegal activity.For now, this Vulnerabilities Scanner is the close source but has the plan to made open source in the later stage.

Keyword: web Vulnerabilities Scanner, Vulnerability, DAST, pen-testing, a close source

1. INTRODUCTION:

In today's world web application is becoming much more popular due to its client-server architecture. This type of architecture makes client lightweight, this make both low and high power device do the same work.From the developer point of view, its become very easy develop an application for different platform like Windows, Linux, Mac but it's also made easy for a hacker to intrude into a bigger organization by exploiting the vulnerabilities of a web application.

An employee may use web application that may not relate to his work and web application owner its nitre profit or loss form organization in which that employee work for but that web application has some.Serious vulnerabilities that hacker may find inserting and intrude in the employee system and later into the company network when employee connect his system either physical or using VPN. This type of case is not common but also not impossible in today's world.

Due to this reason, it's becoming more and more important to create a secure application regarding what your web application do it may be a simple blogging site or complex E-commerce site. Due to the dynamic nature of web application has many flaws like:

- Design issue
- misconfigured web servers
- Not validating input
- Cookies
- Weak session id
- Weak encryption
- Unprotected Http header

And many more

Mostly security test tool can be classified into two broad categories:

I) Static Application Security Testing (SAST)

II) Dynamic Application Security Testing (DAST)

1). Static Application Security Testing (SAST):- These types of tools are used to analyze source code or compiled code. This type of tool is mostly used by the developer or white box tester to find flaws in code or flaws when it's converted into executable, but this type toll finds the very small percentage of security flaws in actual application and even less in case of a web application. Some open-source SATS tool is as follows:

- .NET Security Guard – only work with .NET applications. It will find SQL injections, LDAP injections, XXE, cryptography weakness, XSS and more.

- Sink Tank - Byte code static code analyzer for performing source/sink (taint) analysis.
- Sonar Qube - Scans source code for more than 20 languages for Bugs, Vulnerabilities, and Code Smells.
- Visual Code Grepper (VCG) - Scans C/C++, C#, VB, PHP, Java, and PL/SQL for security issues and for comments which may indicate defective code.

2) Dynamic Application Security Testing (DAST):- These types of tools are used to test applications is an operational state without hindering the actual working of the application. This type of tool is generally used in behavioural testing. In behavioural testing process see how the application behaves in different scenarios some times its also known as Black Box testing Web Vulnerability Scanner also comes under this category. Open Web Application Security Project (OWASP) it's a non-profit foundation whose aim to make the web a much secure place OWASP provides a resource for free to everyone it's also generated report regarding top vulnerability.

2.1. OWASP 10 vulnerability:

Injection:- Injection, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query.
Broken Authentication:- Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
Sensitive Data Exposure:- Many web applications and APIs do not properly protect. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.
XML External Entities (XXE):- Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
Broken Access Control:- Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality or data, such as access to other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
Security Misconfiguration:- This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information.
Cross-Site Scripting (XSS):- XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
Insecure Deserialization:- It often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
Using Components with Known Vulnerabilities:- Applications and APIs using components with known vulnerabilities may undermine application defences and enable various attacks and impacts.
Insufficient Logging &Monitoring:- Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract or destroy data.

3. WORKING MECHANISM

This vulnerability scanner consists of four modules:-

- Scan Engine
- Scan Database
- Report Module
- User Interface as shown in figure 1.

I) Scan Engine:- Performs security checks according to a different algorithm, discovering vulnerabilities.
II) The Scan Database:- Stores and manages vulnerability information, scan results, and other data used by the scanner.
III) The Report Module:- This module generatesa report of than scanning to the user, the format of the report is generally HTML or PDF.

IV) The User Interface:- This module takes input from the user. It may be either a graphical user interface (GUI) or command-line interface(CLI) for now its only support command, GUI is for future work.

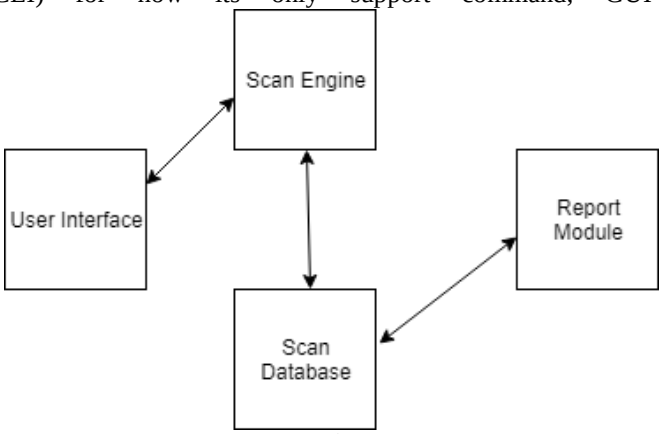

Fig 1 Module of Web Scanner

3.1. Scanning process

The scanning process starts as soon as the user enters the URL or the IP address of the website. The process of scanning vulnerability scanner its divide into three main part.

- Crawling Component
- Attacker Component
- Analysis Component

3.1.1 Crawling Component:

When the user givesURL of the web site the crawling kick into action generally web scanner takes root for deep check or link of the specific page can also be given. The user-supplied web address use as a starting point the crawler crawl an ever page and create a tree including image and input field and link to external document and site. A web scanner also has the configurable options for the maximum link depth, maximum number of pages per domain to crawl, maximum crawling time, and the option of dropping external links

3.1.2 Attack phase

After the crawling phase is complete web scanner start the attacking the component of the web page like input fields, cookies, external link etc. In most cases, the attack part is generally focused on the web form because web form is the main entry point to intrude into Database and web or application server. Attacker extracts the action, target address, method (GET or POST) that webform submit. The attacker main focus is to collect its CGI (Common Gateway Interface) parameter. Then attacker actual attack the web field by using the appropriate value. The appropriate is mostly chosen with the help of attack database this database is built inside the web scanner. The form is to upload its content to the web and the output or response is generated by the web server or application server is sending to the analysis component.

3.1.3 Analysis Component

The analysis component analysis and interpret the response made by the web server or application server. The analysis uses both static and knowledge-based algorithm to check whether the attack is successful or not. The output generated by the knowledge-based algorithm generate categorize into four-part and put into the confusion matrix as shown in fig 2 section 5.1 will discuss testing in detailed.

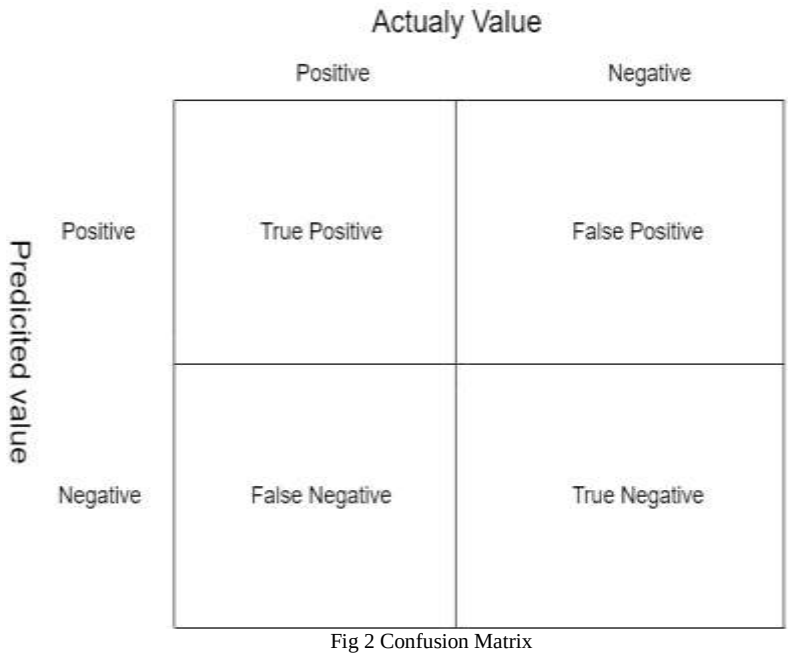

Fig 2 Confusion Matrix

4. CHALLENGES WHEN CREATING WEB VULNERABILITY SCANNER

To create a good Web Vulnerabilities Scanner we have to consider some point

Test plan and several test cases are made to check the effectiveness of web scanner, need to create a detail test plan to how a tool is tested and how to test the result are going interpret and how to generate a detailed and summarize report a PDF and HTML based report is good and it's easy for the user to compare form tool report a non-global format is not advisable. A test case is must prepared on the basis of how actually attacker is going to exploit the vulnerabilities User submits a request in a normal way but attacker in an unexpected way means the code of scanner is not know what to do with that kind of code most attacker examine application response and or notice the change in application behaviour. A web application is must need with some vulnerability to test web scanners existing web app can be used as general-purpose for specific purpose there custom made web app.

Some good vulnerable web:

- buggy web application (Bwapp)
- OWASP Juice Shop
- Damn Vulnerable Web App (DVWA)

5. TESTING THE WEB SCANNER:

We use a number of the algorithm in the phase of testing to check the vulnerability in the sample web application. To compare the algorithm with each other we use Confusion matrix.

5.1. Confusion matrix:

We use Confusion matrix to check which algorithm is work best with which vulnerability. First, we take the output of the scanner and put it into the confusion matrix as shown in Fig 2.
Confusion matrix consist of four cells:

- True positive(TP):- its define the how many numbers of time web scanner tell true when its indented to be true.
- False-positive(FP):- it defines how many numbers of time web scanner tell false when it's indented to be true also know as Type one error.

- False-negative(FN):- it defines how many numbers of time web scanner tell true when its indented to be false also know as Type two error.
- True negative(FN):- its define the how many the number of time web scanner tell false when its indented to be false.

In our web scanner False positive (Type one error) is considered more dangerous as compared to False negative (Type two error), because it reduces the effectiveness of the scanner mean it ignore much vulnerability that is present in the web application, but simply looking in a number does not tell anything so we use mathematically approach know as accuracy.

5.1.1. Accuracy:

Its define how many time web scanner algorithm find correct result out of the total number of the result the value of accuracy is lie between 0 -1. Accuracy can be calculated by the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Eq 1 Formula of Accuracy

But there is a problem with accuracy approach in some case the accuracy is .8 and the result is very bad and in some case, the accuracy is .4 but the result is actually very good. So to correctly identify evaluate the result we use another two mathematically approach know as Recall and Precision.

Recall:- Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the current data set in attack database is correct for a particular type of attack or vulnerabilities. The value of recall is lying between 0 – 1. Recall can be calculated by the following formula:

$$Recall = \frac{TP}{TP + FN}$$

Eq 2 Formula of Recall

Precision:- Precision is defined as the total number of positive guess in which how many are meant to be positive. The value of Precision is lying between 0 – 1. Precision can be calculated by the following formula:

$$Precision = \frac{TP}{TP + FP}$$

Eq 3 Formula of Precision

By calculating Recall and Precision we come across two cases know as High recall, low precision and Low recall, high precision:

High recall, low precision**:-**In this case that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives, but this scenario is very dangerous this actually means web scanner is not able to scan much vulnerability present in the web application. Low recall, high precision**:-** This case shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP), but report generate is this scenario is useless and waste for the user because this report contains the many vulnerabilities that not even exist in the web application.

To solve this problem we use another approach to mathematically check result know as F-measure.

F-measure:-Its is the ration of recall and precision. The value of F-measure is lie between 0 – 1,It can be calculated by the following formula:

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

Eq 4 Formula of F-measure

By calculation f-measure, we come to the conclusion that the higher the value of f-measure better result in every case and it also solves the problem of High recall, low precision and Low recall, high precision even if one of the two problems is occurring and the value f-measure is high we mathematically confirm that the scan done by the algorithm that has a higher value of f-measure is much better whose algorithm whose value is less of the web application is very good.

6. FUTURE WORK

Now the Web Vulnerabilities Scanner is not supported GUI and it only detects SQL injection Vulnerability in web application with very good precision so future work of this Vulnerabilities Scanner is to make its GUI and support most of the Vulnerability present in the today's world and make application mush faster and include the feature so that user can save and retrieve their work at any time and made report generated by the scanner compatible with some open-source scanner and make option for external plugins so that user add an extra feature in it last but not least make our application much secure then it's was today.

7. CONCLUSION:

In this paper, several prospects about web vulnerability scanner vast studied and how it works and study the different vulnerability and scanner.

8. ACKNOWLEDGEMENT

I'm sincerely thankful to Jain University for providing me with the opportunity to write a research paper on this topic. I'm also thankful to Mr Subarna Panda, Assistant Professor and Dr Lakshmi JVN, Associate Professor for guiding me in every single stage of this research paper and supporting throughout the process in preparing a meaningful paper. I am also thankful to Dr.N.MNachappa (Head of School-CS& IT) of Jain (Deemed-to-be University) who have helped me during the course of this research paper in different ways. Through this paper, I have learnt vulnerabilities functions and prepare mitigation process. It has helped me analyze how the information can be secured and its advantages and disadvantages.

8. REFERENCE

[1] https://owasp.org/
[2] SecuBat: a web vulnerability scanner https://www.researchgate.net /publication /221023838 _SecuBat_ a_Web_ ulnerability_scanner
[3] A Case Study on Web Application Vulnerability Scanning Tools <https://ieeexplore.ieee.org/ abstract/document/6918247>
[4] Using web security scanners to detect vulnerabilities in web services <https://ieeexplore.ieee. org/abstract/document/5270294>
[5] Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner https://www.usenix .org/conference/usenixsecurity12/technical-sessions/presentation/doupe
[6] An automated vulnerability scanner for injection attack based on injection point https://ieeexplore.ieee .org /abstrac t/document/568553