Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-2, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Web Vulnerability Scanner Application

Pallavi Deshmane[1], Shweta Singh[2], Nakshi Doshi[3], Harshit Punatar[4], Shashank Gangar[5]
Assistant Professor, Computer Engineering, Shah and Anchor Kutchhi Engineering College, Mumbai, India[1]
Student, Computer Engineering , Shah and Anchor Kutchhi Engineering College, Mumbai, India[2,3,4,5]

Abstract*: As the popularity of the web increases and web applications become tools of everyday use, the role of web security has been gaining importance as well. The last years have shown a significant increase in the number of web-based attacks. Too many nouns web application security vulnerabilities result from generic input validation problems. Examples of such vulnerabilities are SQL injection and Cross-Site Scripting (XSS). Although the majority of web vulnerabilities are easy to understand and to avoid, many web developers are, unfortunately, not security-aware. As a result, there exist many web sites on the Internet that are vulnerable. This project implemented an automated vulnerability scanner that for the injection attacks. To this end, we implemented a system that automated scanned the injection attack vulnerabilities. Our system automatically analyzes web sites with the aim of finding exploitable SQL injection and XSS vulnerabilities. It is able to find many potentially vulnerable web sites.*

Keywords*: Website Scanner, Web based application, Website Vulnerability Analysis, CSS.*

I. INTRODUCTION

Security is a critical part of your Web applications. Web applications by definition allow users access to a central resource — the Web server — and through it, to others such as database servers. By understanding and implementing proper security measures, you guard your own resources as well as provide a secure environment in which your users are comfortable working with your application.

Web application security is a branch of "Information Security" that deals specifically with security of websites, web applications and web services.

Web Security blocks web threats to reduce malware infections, decrease help desk incidents and free up valuable IT resources. It has more than 100 security and filtering categories, hundreds of web application and protocol controls, and 60-plus reports with customization and role-based access. You can easily upgrade to Web Security Gateway when desired to get social media controls, SSL inspection, data loss prevention (DLP) and inline, real-time security from Websense ACE (Advanced Classification Engine).

II. PROBLEM DEFINITION AND APPROACH

Vulnerability is a hole or a weakness in the application, which can be a design flaw or an implementation bug, that allows an attacker to cause harm to the stakeholders of an application. Stakeholders include the application owner, application users, and other entities[1].

This technology framework was selected because of its popularity and wide-spread use. Each web application vulnerability scanner is tested against the web application following a prescribed approach that includes a set of initialization, execution, classification, and analysis procedures. By using both a secure and insecure version of a custom-built web application, false-positive and false-negative results can be linked to the techniques used by the scanners to discover vulnerabilities. This association between the techniques used and the false-positives or false negatives reported can be used to suggest improvements for web application scanning techniques[7].

Step 1: Enter the URL.
Step 2: Select the type of vulnerability you want to scan
Step 3: Start scanning.
Step 4: Check for the given conditions according to the selected vulnerability scan.

Step 5: After observing the conditions of the selected vulnerability prepare a report

III.EVALUATION

ALGORITHM FOR SQL INJECTION:

1. Initialize sql characters in an array
2. Create two lists to store the sql error messages

(i) One for storing specific database error messages like sql error messages etc

(ii) Other for storing generic database error messages

3. Initialize error values in to the maps/list mentioned above
4. Initialize the scanner method – the scanner accepts the http message as input from the the crawler - http message has details on each request or url with the parameter list
5. For each parameter in the http message

(i) Input sql characters from the sql characters array

(ii) Verify the response to check for any matches on error messages from the two maps or lists.

(iii) If a match occurs -Flag as sql vulnerability

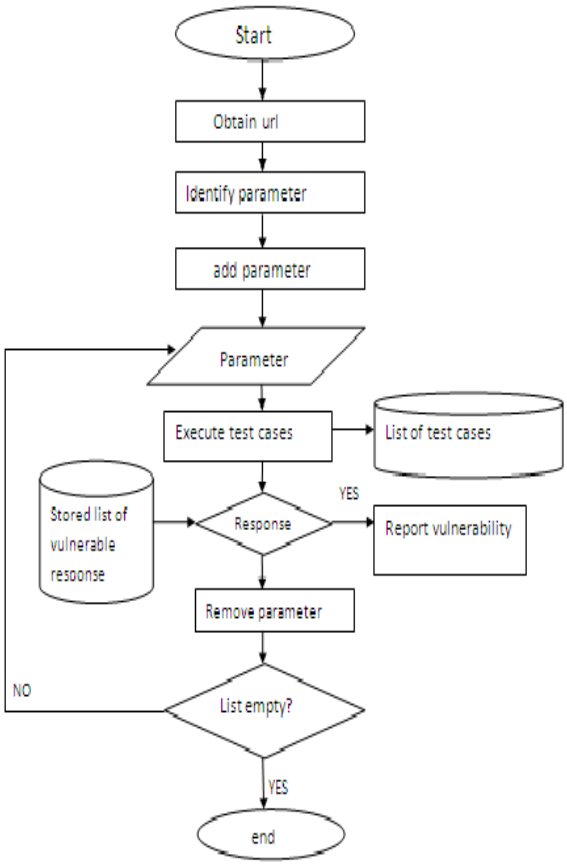
(iv) Else - Repeat step until the end of parameter list is reached

6. End

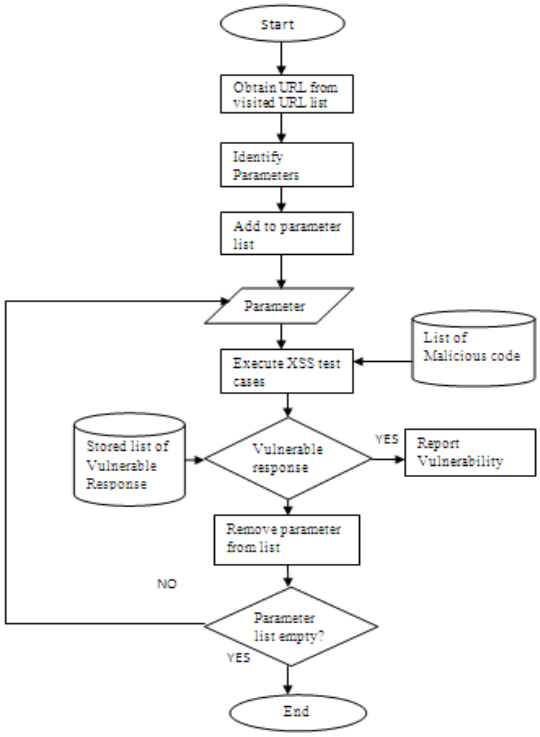
ALGORITHM FOR XSS:

1. Start
2. For each url in the list of visited urls
   a. Identify all parameters
   b. Push parameters in to parameter list
   c. For each parameter in the parameter queue
      i. Supply a script or a XSS test case as input to the parameter and pass the request
      ii.Verify the response to identify the supplied script or test case reflected back
3. Report the vulnerability if the response has a script
4. End

FLOWCHART OF SQL INJECTION



FLOWCHART OF XSS



IV. ANALYSIS

TABLE I
VULNERABILITIES NAMES

WEB APPLICATION VULNERABILITIES	
-SQL INJECTION -CROSS SITE SCRIPTING -SESSION MANAGEMENT	-CROSS SITE REFERENCE FORGERY -BUFFER OVERFLOW

TABLE II
WEB SCANNERS NAM

OPEN SOURCE WEB SCANNING TOOLS	
-GRABBER -VEGA	-WEBSECURIFY -WAPITI

As the above Table I shows some of the vulnerabilities
which occurs on web scanning process and all the names in the Table II shows those website's names who can perform scanning of list of shown vulnerabilities.

TABLE III
ATTRIBUTES OF PROPOSED SYSTEM

Attribute	**Previous Model**	**Proposed Model**
Registration	Compulsory	Not Compulsory
Download Software	Required	Not Required
User Interface	Present as GUI	Present as Web Page Based
Accuracy	Less	More
Report	Not Graphically Only Theoretically	With both
Avoidance	Not Informed	Informed .

But the proposed system will do the task efficiently as it will take the less time for scanning the websites for those vulnerabilities which are found common in most of the websites(like sql injection and XSS) compare to all which are listed above.
The proposed system will generate the report to acknowledge the list of vulnerabilities using both theoretically and graphically.

V. CONCLUSION AND FUTURE WORK

The proposed system is extensive in the execution of its detection mechanism against web application vulnerabilities. Testing of web applications for weaknesses is a significant step in safeguarding web applications. The proposed system reports vulnerabilities and presents a proficient manner while reporting discovered vulnerabilities. However since the proposed system did not scan 100% of the existing vulnerabilities. There is need to increase the algorithm crawling component in order to ensure that it executed "deep" crawling. In addition the results presented shows that the proposed system needs to be optimised to do the scanning in a short period of time. More research is needed to come up with a sophisticated algorithm that has the capacity to detect more vulnerabilities[3].

VI. RESULTS AND DISCUSSION

Proposed system will generate better results overall when compared to other tools used, although the tool will have a higher detection accuracy when compared to the other tool. The only drawback is that it was reported to take a longer time to scan than most of the web scanners that were used in this study.

A comparative study of web vulnerability scanners has also been performed by other researchers from different parts of the world. Although the tools and web applications used are not similar, the vulnerabilities are the same.

In a study conducted by Fonseca et al (2014) shows that many open source WVS have a low ability to detect vulnerability. This is in line with the results analysed after the end of this study. The researcher has developed a more sophisticated algorithm that address this concern and increased the number of vulnerabilities detected.[6]

VII. REFERENCES

1. M. Parvez, P. Zavarsky and N. Khoury, "Analysis of effectiveness of black-box web application scanners in detection of stored SQL injection and stored XSS vulnerabilities," 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, 2015, pp. 186-191. doi: 10.1109/ICITST.2015.7412085

2. X. Wang, K. Zhang and Q. Wu, "A Design of Security Assessment System for E-Commerce Website," 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2015, pp. 137-140. doi: 10.1109/ISCID.2015.

3. Karumba, Muiruri Chris, Samuel Ruhiu, and Christopher A. Moturi. "A Hybrid Algorithm for Detecting Web Based Applications

Vulnerabilities." *American Journal of Computing Research Repository* 4, no. 1 (2016): 15-20.

4. Park, N. (2015). Detection Experimentation and Validation of Web Applications using Both Static and Dynamic Analysis. International Information Institute (Tokyo). Information, 18(5 (A)), 1735.

5. Vieira, "Using Web Security Scanners to Detect Vulnerabilities in Web Services"; IEEE/IFIP Intl Conf. on Dependable Systems and Networks, DSN 2009, Lisbon, Portugal, June 2009; http://eden.dei.uc.pt/~mvieira

6. Fonseca, J., Vieira, M., & Madeira, H. (2014). Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection. Dependable and Secure Computing, IEEE Transactions on, 11(5), 440-453.

7. Park, N. (2015). Detection Experimentation and Validation of Web Applications using Both Static and Dynamic Analysis. International Information Institute (Tokyo). Information, 18(5 (A)), 1735.