Website Vulnerability Scanner

Prajnesh Kundar[1] Ajinkya Karode[2] Rahul Jangida[3] Prof. Shweta Sharma[4]
[1,2,3]Student [4]Professor
[1,2,3,4]Department of Computer Science
[1,2,3,4]Atharva College of Engineering, Mumbai, India

Abstract— Among the Distributed systems, web services tend to work over dynamic connections. Such type of technology was specifically designed to pass an SOAP message through various firewalls using open ports. However, this technology was then later on involved in various security threats such as injection attacks, denial of service (DoS) etc. A Survey stated more than 90% of the websites have at least one of the mentioned security threats. Detection of vulnerabilities encourages the software developers to use different security testing to reduce harmful attacks. So giving a Black-box approach, this research, with the help of a crawler, will be able to detect various security threats like SQL injection, cross site scripting, header manipulation, Click jacking etc. The idea of combining a crawler is for not only scanning the threats for a given URL but also scanning the links within URL. With this research an authentication page will also be attached so as to check whether the user is authorized or not. User will also be able to store the results for future reference.

Key words: SQL Injection, Vulnerability, Black-Box, Crawler, Header Manipulation

I. Introduction

We know that an ever increasing number of highly profiled data breaches have plagued organizations over the past years, a great number of these breaches come via injection attacks. Website Vulnerability scanner parses URLs from the target website to find different types of vulnerabilities.

Website Vulnerability Scanner is useful in finding out bugs in website and also it shows the type of Vulnerabilities present in that website. This project has been developed in JAVA, HTML, Apache Tomcatv6.0, JSP, MS SQL Server2012,

This scanner checks different web applications for popular security problems such as SQL injection, cross-site scripting, Click Jacking and other vulnerabilities. Website Vulnerability Scanner is an automated procedure of recognizing vulnerabilities on Web Applications/Websites in a network. Also this scanner will not only scan the given URL but also the different URL in it using a Web Crawler.

II. Literature review

According to Curphey and Araujo there are eight categories of web application security assessment tools: source code analyzers, web application (black box) scanners, database scanners, binary analysis tools, runtime analysis tools, configuration management tools, HTTP proxies, and miscellaneous tools[5]. The most common of these web application assessment tools are source code analyzers and web application scanners [3]. Source code analyzers generally achieve good vulnerability detection rates, but are only useful if the web application's source code is available. On the other hand, web application vulnerability scanners are the tools which most closely mimic web application attacks, but have been known to perform rather poorly[1]. The web application vulnerability scanner is the type of tool that is analyzed in this thesis because the techniques that these tools currently use to scan web applications need to be improved. Improving the scanning techniques of these scanners will allow them to achieve better performance and, therefore, increase their credibility [4]. However, in order to understand and improve web application scanners, the common vulnerabilities that they aim to detect must be understood first[2].

According to Johari, R.; Sharma, P., In the article "A Survey on Web Application Vulnerabilities (SQLIA, XSS).Exploitation and Security Engine for SQL Injection", main objective of this paper is only on the study of various types of Structured Query Language Injection attacks and Cross Site Scripting vulnerabilities and their security techniques. They proposed to future study on Structured Query Language Injection attacks[6].

Following is a list of some common vulnerabilities and its detection techniques[7]:

A. *Unvalidated Input*

All user input that provided to the web applications requested, need to check by against strict format that specifies exactly what input must be allowed. Ensure that all parameters are validated before they are used. A tool or library is most effective, as the performing the checking should be placed.

B. *Broken Access Control*

The code implementation of the access control policy should be verified. Penetration testing can be useful in verifying if there are problems in the access control. In the web application, if there is categories of users that can be accessed through the interface, verify each interface to make sure that only authorized users can allowed access.

C. *Broken Authentication and Sessions Management*

Detailed review of authentication mechanisms to ensure that user's credentials are protected and only an authorized user can change them. Review your session management mechanism, that session identifiers are always protected.

D. *Improper Error Handling*

Error handling should be consistently focus on the entire application. A code review will reveal how the system is intended to handle various types of error. Simple testing can determine how your site responds to various kinds of input errors.

E. *Parameter Modification*

Ensure that application must not allow parameter values, query string or form GET parameters to the URL. These are handled through dynamic parameter detection.

F. Insecure Configuration Management

Web server configuration files must not be accessed by the user, unauthorized files and directories permissions, Server software flaws or misconfigurations that permit directory listing and directory traversal attacks, some of these problems can be detected by scanning

III. PROPOSED SYSTEM

This system tends to replace the existing manual system for the scanning process which is a time consuming, less interactive and highly expensive. The main features of this system will be creating report and find vulnerabilities such as Cross site scripting, SQL injection, Click jacking and storing Scanned data, process initiation, and after that it generates a report of the whole scanned websites. Web crawler crawls the given URL and finds out all the links, form tags etc where data can be entered. The end result of crawling is a collection of links present on the site at a central location. It finds out the flaws in the website using black box approach. We are going to detect various ways in which the site can be attacked.

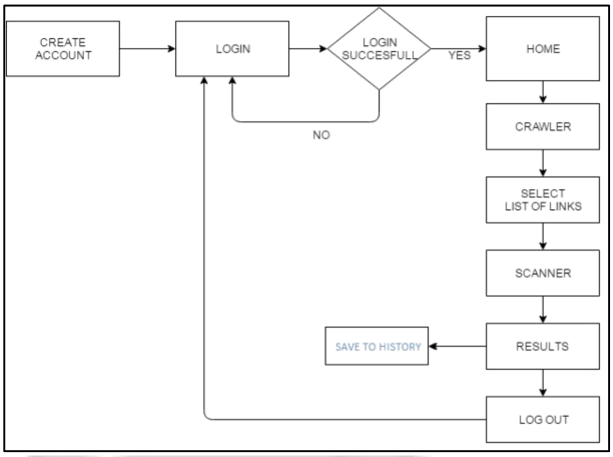


Fig. 1: Work Flow Diagram

A. Step 1:

Firstly the user has to create a account and make sure to use a unique username. If the user is already registered he can use the sign in link present on the page

B. Step 2:

Then using the username and Password user can login to our Project.

C. Step 3:

Once a correct username and corresponding password is entered the user is redirected to Home Page from where user can either go to History of his/her Scanning ,start a new scanning or can go to setting to change personal setting to the page.

D. Step 4:

On Selecting History user can view the results and when the test was performed.

E. Step 5:

On Selecting Scanner user is redirected to a webpage where the URL of the link to be scanned is taken as input.

F. Step 6:

The Output of the Scanner page is a list of Links present on the URL provided by user from Which User can select the sites he want to scan for the Vulnerabilities.

G. Step 7:

Once the submit button is pressed different testing codes are executed and the corresponding output is produced as Name of Site and on next line Name of Vulnerability and Vulnerable/NotVulnerable(based on tests)

H. Step 8:

From the Results page user can Store the Result to History and/or go to any page of choice.
Also

IV. DESIGN DETAILS

Following are diagrams represent the flow of the Website Vulnerability Scanner.

A. Use Case Diagram:

Fig 2 Shows the Interaction of user to the Server and the Various Options available to user.

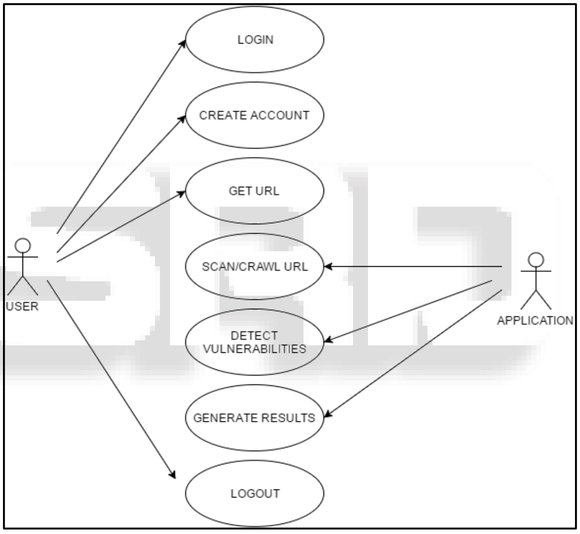


Fig. 2: Use Case Diagram

B. Sequence Diagram:

Figure 3 Shows the Sequence in which the Execution of Website Vulnerability Scanner is going to take place.

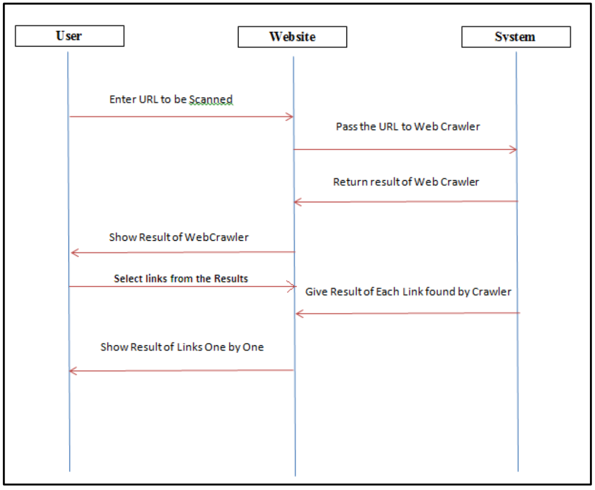
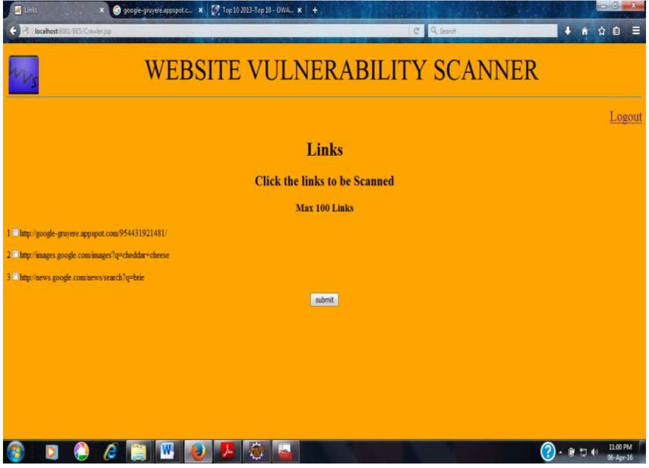

User
Website
System
Enter URL to be Scanned
Pass the URL to Web Crawler
Return result of Web Crawler
Show Result of WebCrawler
Select links from the Results
Give Result of Each Link found by Crawler
Show Result of Links One by One

Fig. 3: Sequence Diagram

V. Result

The following are the screenshots of the results of this Crawler for sitehttp://google-gruyere.appspot.com which is a specially designed vulnerable site by Google inc. for testing and learning of developers. fig 2 Shows the three links detected by the crawler.


Links
google-gruyere.appspot.c...
Top 10 2013-Top 10 - OWA...
localhost:8081/BE5/Crawler.jsp
Search
Logout
submit
11:00 PM
06-Apr-16

WEBSITE VULNERABILITY SCANNER

Links

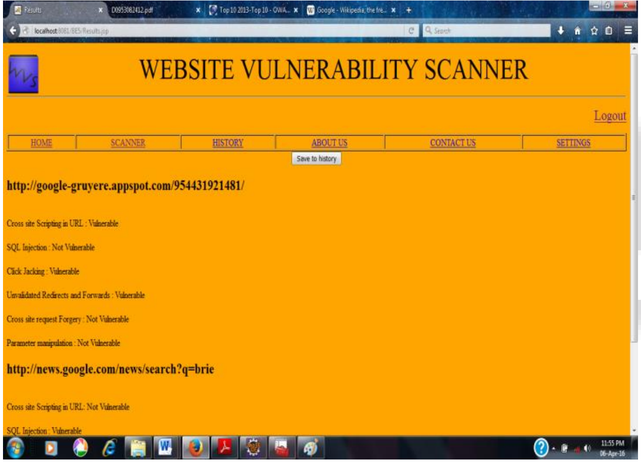
Click the links to be Scanned

Max 100 Links

1 □ http://google-gruyere.appspot.com/954431921481/
2 □ http://images.google.com/images?q=cheddar+cheese
3 □ http://news.google.com/news/search?q=brie

Fig. 4: Output of Crawler

Figure 5 shows the output of the Selected links from the Output of Crawler



WEBSITE VULNERABILITY SCANNER

http://google-gruyere.appspot.com/954431921481/

Cross site Scripting in URL : Vulnerable

SQL Injection : Not Vulnerable

Click Jacking : Vulnerable

Unvalidated Redirects and Forwards : Vulnerable

Cross site request Forgery : Not Vulnerable

Parameter manipulation : Not Vulnerable

http://news.google.com/news/search?q=brie

Cross site Scripting in URL: Not Vulnerable

SQL Injection : Vulnerable

Fig. 5: Final Result

A. *Input*

- URL to be Scanned.
- Selection of Links to be Scanned.
- Save to History Button.

B. *Output*

- List of Links in the Given URL.
- Result after Testing.
- Save to History.

VI. Conclusion

Web Vulnerability Scanner is used for website security scanning that checks for SQL injection, Click jacking, Cross Site Scripting and other vulnerabilities. It checks the strength of the password on authentication pages. Once the scan is completed a detailed report is provided and those reports pinpoints where the vulnerabilities exist. By having a look at the report it is clear that majority of the security alerts are informational type and are not considered high level alerts. Overall we may find that facebook.com is having fewer alerts and is more secure as compared to 3rtechnologies.net. It is also noticed that there is a lot of difference in the time required to scan both the portals .If a website has more vulnerabilities then it takes more time for scanning and detection.

Acknowledgement

We would like to thank our project guide Prof. Shweta Sharma for her enormous co- operation and guidance. We have no words to express our gratitude for a person who wholeheartedly

We are also thankful to our Principal Dr. S.P. Kallurkar, our HOD Prof. Mahendra Patil, who have guided us to develop a very good project idea.

References

[1] M Curphey and R.Arawo .Web application security assessment tools. Security and Privacy, IEEE, 4($):32-41, 2006.

[2] Wimmer Maria and Bredow Bianca Von," E-Government: Aspects of Security on different layers ", 12th IEEE International workshop on Database and Expert systems application, munich germany 2004.

[3] R.Auger .The Web Application Security Consortium-Remote files Inclusion. Available at http://projects.webappsec.org/remotefileinclusion, 2009.

[4] T. Jim, N. Swamy, and M. Hicks. Defeating script injection attacks with browser enforced embedded policies Session:Defending against Emerging Attacks 2007.

[5] David Endler. The Evolution of Cross Site Scripting Attacks. Technical report, iDEFENSE Labs, 14151 Newbrook Drive 2002

[6] Johari, R.;Sharma, P. USIT, GGSIP Univ., Delhi, India Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQLInjection Communication Systems and Network Technologies (CSNT), 2012 International Conference on Date of Conference: 11-13 May 2012

[7] Katkar Anjali S., Kulkarni Raj B. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012