ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

Data Mining Based Web Vulnerability Scanner

Nilambari Chhagan Sonawane

Former Student, MIT College of Engineering, Pune, Maharashtra, India

ABSTRACT: Due to the increase in the popularity of the web and overall web-based application, web security has also started gaining importance. In the last few years there is a significant increase in the number of web-based attacks. Generic input validation has become a most common reason for web application security vulnerabilities. SQL injection and Cross-Site Scripting (XSS) are the examples of such vulnerabilities. Majority of web vulnerabilities are easy to understand as well as to avoid, but due to unawareness of web-security many web developers are not able to understand. So as a result of that, there exist many web sites on the Internet that are vulnerable. In this project we have implemented an automated vulnerability scanner that for the injection attacks. We have implemented a system that automated scanned the injection attack vulnerabilities. This system automatically analyses web sites with the aim of finding exploitable SQL injection and XSS vulnerabilities. It is able to find many potentially vulnerable web sites.

KEYWORDS:Web, vulnerability, SQL, Cross-Site Scripting, web-security.

I. INTRODUCTION

Web application Security: Security is a vital part of your Web applications. Basically, web applications allow users access to a central resource — the Web server — and through it, to others such as database servers. By taking some precautions and implementing proper security measures, you protect your own resources as well as provide a secure environment in which your users are comfortable working with your application. Web application security is one of the types of “Information Security” and it deals specifically with security of websites, web applications and web services. Web security helps to block the web threats, reduces malware infections, decrease help desk incidents and free up valuable IT resources.

Web Vulnerability: 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 that rely on the application. The term "vulnerability" is often used very loosely.

II. LITERATURE SURVEY

The rapid and tremendous growth of Information and Communication Technology (ICT) has increased access to web applications. This increased access has paved the way for disadvantageous security and vulnerable threats in the form of attacks in web applications. Various detection and prevention techniques have been proposed by researchers in the field of web applications and technologies development.

Vulnerabilities distributed per type:
Figure presents the final distribution of the vulnerabilities per type including the doubtful cases (i.e., optimistic evaluation of the scanners). As the doubtful cases only affect the SQL Injection, it means that the number of SQL injection vulnerabilities could be overestimated. Scanners have found 177 different vulnerabilities in 25 different services, which representapproximately 8.33% of the tested services. The predominant vulnerability is SQL Injection,

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

representing 84.18% of the vulnerabilities found. This is a very important observation due to the high number of cases found and the high severity of this vulnerability.

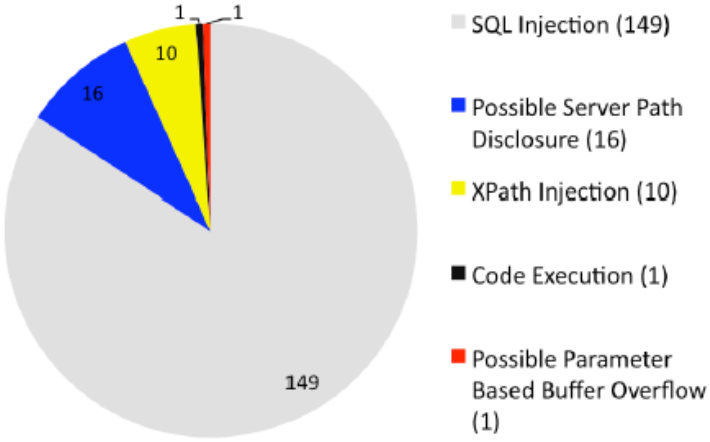

Fig1: final distribution of the vulnerabilities per type

III. PROPOSED SYSTEM

1. **URL crawling**: URL crawler is used to crawl the URL's from the search engine. Crawler will find number of web pages if we search for any keyword using search engine. It's been a recursive call for one single search because when we search any One keyword it can contains number of web pages and each web page has its URL but when we click on that URL it will again number of URL's for that one web page. So basically, the URL Crawler is used to crawl the web pages for the particular search it will automatically crawl the URL's and will show limited URL's or web pages when we search any keyword the limit is given up to 6-7 web pages will show after we will search for the particular keyword.

2. **Search engine:** Search engine is mainly used for searching contents on the World Wide Web. The search engine is mainly used for getting the limited data instead of getting huge number of data after applying the search.

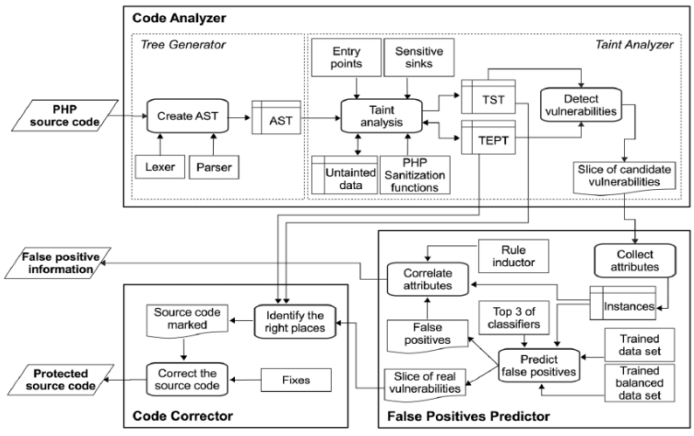

Fig 2: Existing system

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

3. **Remote sites:** When we search on search engine it shows us number of sites as a result. We get results from multiple sites and when we click on particular URL from that URL we again get multiple URL's means the sites are recursively called for this the directory traversal is used in the remote site so that the number of sites on a particular search will be appear.

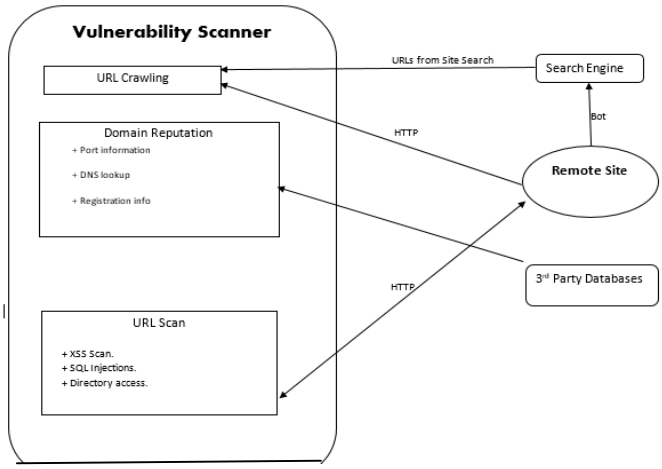

Fig 3: Block diagram of system

4. **3rd Party database:** This database is mainly used to check status of site and to check whether the site is corrupted. It also checks for black listed sites and if the site is black listed then it gives the message to check the vulnerability for such corrupted sites.

5. **Domain reputation:** Domain reputation is used to check for black listed sites so that reputation for that site will check. Domain reputation checks the vulnerability of sites using different domains mainly the RBL's are used to check mail server's IP and it checks whether the server's IP is black listed or not.

6. **URL Scan:** There are two types of methods for the URL scan GET method and POST method. These methods help to check the Vulnerability for URL. For GET method there is one specific pattern that defines and for every URL these specific pattern is used to check Vulnerability while in POST method there is no specific type of pattern is used in this the whole URL is applied to check Vulnerability.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

IV. ALGORITHM

ALGORITHM (FOR SEARCHING MORE SITES)

1. Input: seed websites and harvested deep websites
2. While cansiteless than a threshold do
3. Pick a deep website
4. Site= get Deep Websites
5. Result page= reverse search (site)
6. links= extract links (result page)
7. for each link into links do
8. page= download page (link)
9. relevant = classify (page)
10. if relevant then
11. Relevant sites= extractunvisitedsites.
12. Output= Relevant sites.

ALGORITHM (FOR INCREMENTAL SITE PRIORITIZING)

1. Input: SiteFrontire
2. HPQ= Create queue
3. LPQ= Create queue
4. if HPQ is empty then
5. HPQ.addAll(LQueue)
6. LPQ.clear()
7. End
8. site = HPQ.poll()
9. relevant = classify Site(site)
10. if relevant then
11. performInSiteExploring(site)
12. Output forms and OutOfSiteLinks
13. siteRanker.rank(OutOfSiteLinks)
14. if forms is not empty then
15. HQueue.add(OutOfSiteLinks)
16. End
17. Else
18. LQueue.add(OutOfSiteLinks)
19. End
20. output: Out-of-site links and searchable forms

Fig 4: Algorithm

V. ALGORITHM FOR PROPOSED SYSTEM

1. Start
2. Enter the URL.
3. Select the type of vulnerability you want to scan for:
   i. Default: checks for SQL injection and cross site scripting both.
   ii. SQL Injection: checks for SQL injection.
   iii. XSS: checks for cross site scripting.
4. Start scanning.
5. Check for the given conditions according to the selected vulnerability scan.
6. After observing the conditions of the selected vulnerability prepare a report

VI. DESIGN OF SYSTEM

1. Architecture

Data flow diagram (DFD):

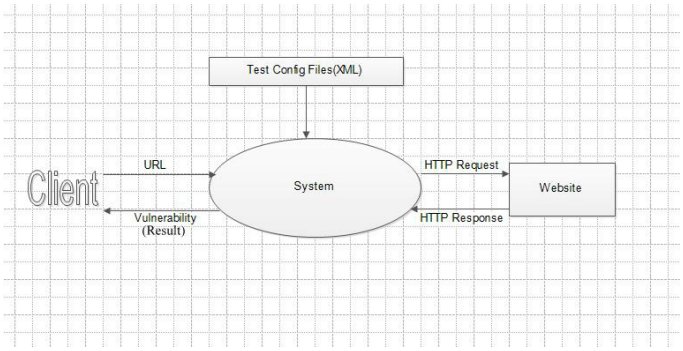

Fig 5: Level 0: Context

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

Level 0 shows the over view of the software. The client will provide URL of the website to be tested and what kind of test to be performed on it. The system will build the crafted request with the help of test criteria. The build crafted request is send to the server and scanner analyses the response return by the web server and report will be send to client.

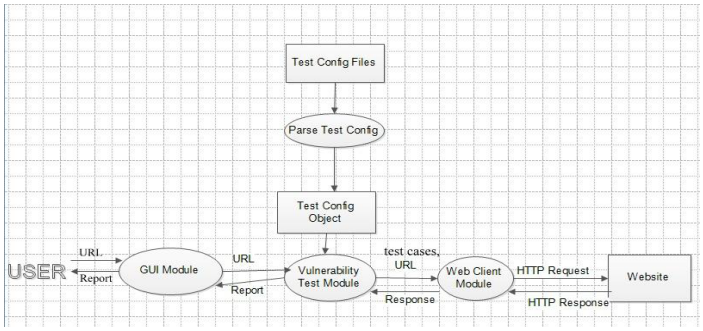

Fig 6: Level 1: System Architecture

Level 1 is basically the architecture of the system i.e. software. It has three module i.e. GUI Module. Vulnerability Test Module, Website Client Module.

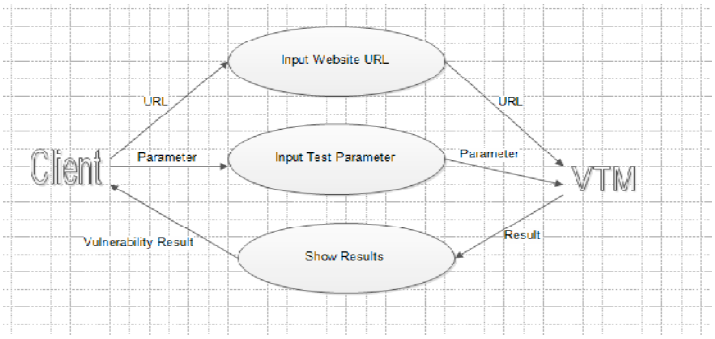

Fig 7: Level 2: GUI Module

Graphical User Interface Module is basically a user interface of the software where user will input the URL of the website and test to be performed on it. After the test is completed the result is shown here.

2. Vulnerability Test Module

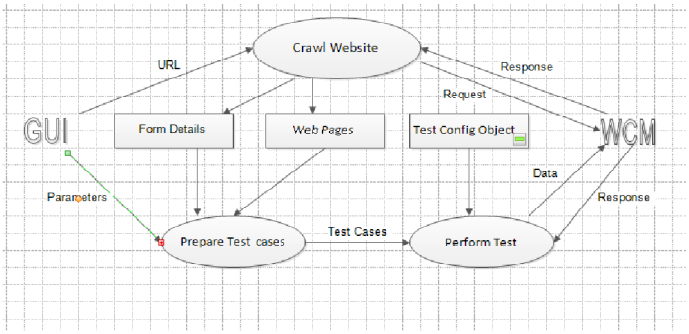

Fig 8: Vulnerability Test Module

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

Vulnerability Test Module is a module in which crawling of the website, preparing test cases and analyzing the response that is sent by the web server is done. For e.g. To check whether SQL injection attacks are possible, the vulnerability scanners send modified requests and analyze the responses returned by the server. A server may respond with a rejection page or with an execution page. A rejection page corresponds to the detection of syntactically incorrect or invalid inputs. An execution page is returned by the server as a consequence of a successful execution of the request. This page may correspond to the "normal" scenario, i.e., in the case of a legitimate use of the web site, but might also result from a successful exploitation of an injection attack

3. Web-Client Module

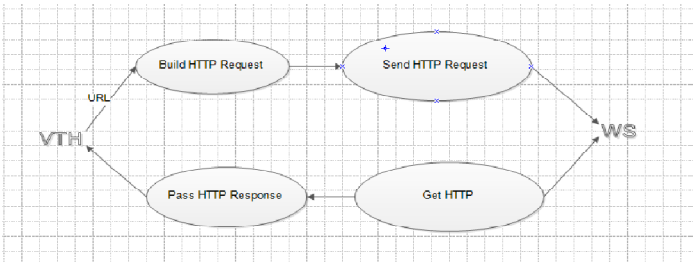

Fig 9: Web-Client Module

4. Diagrams

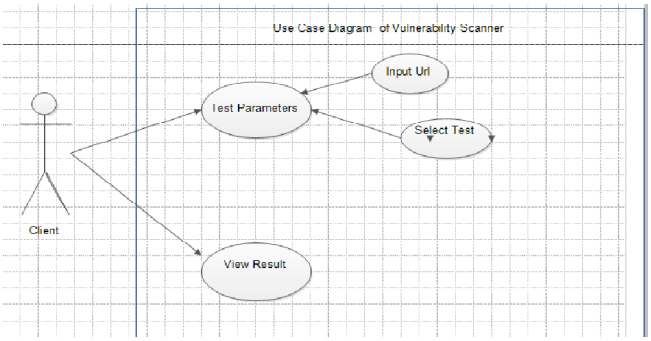

Fig 8: Use-Case Level Diagram

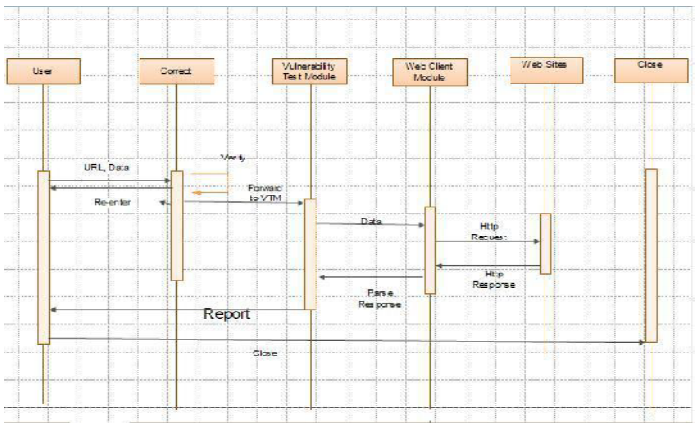

Fig 10: Sequence Diagram

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

In this module, the scanner will build the crafted Http Request with the help of test cases and it is sent to web server and in return web server sends response to Web-Client Module and it than parses the response and sends the parse response to the Vulnerability Test Module where the response is analyzed.

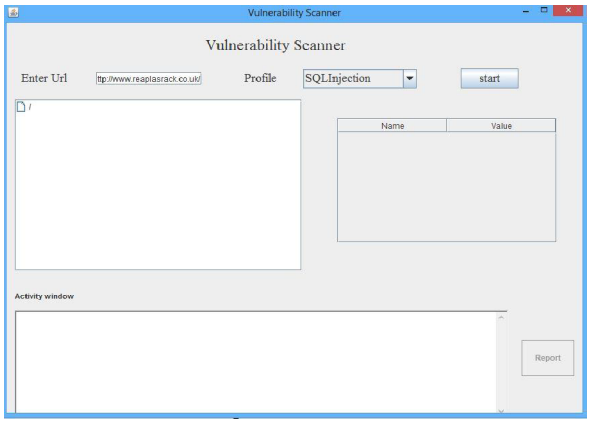

Fig 11: Snapshot- Vulnerability scanner

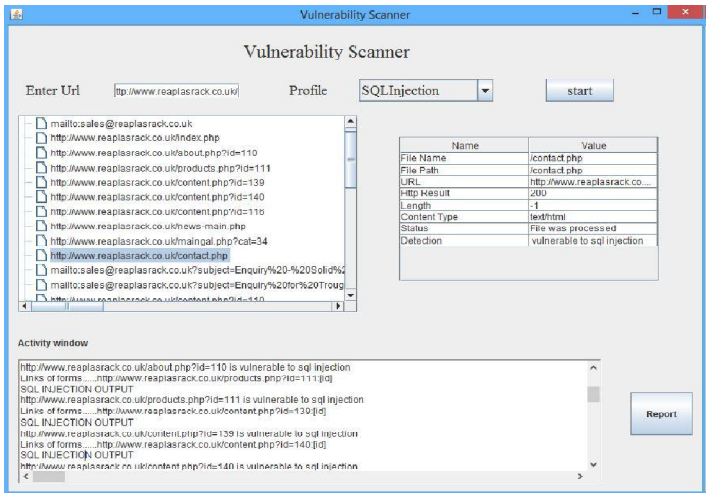

Fig 12: Output of system

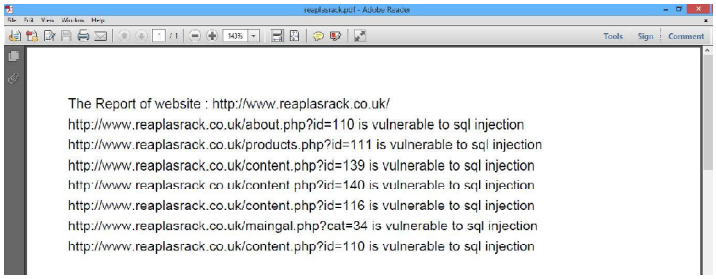

Fig 13: Report of website

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

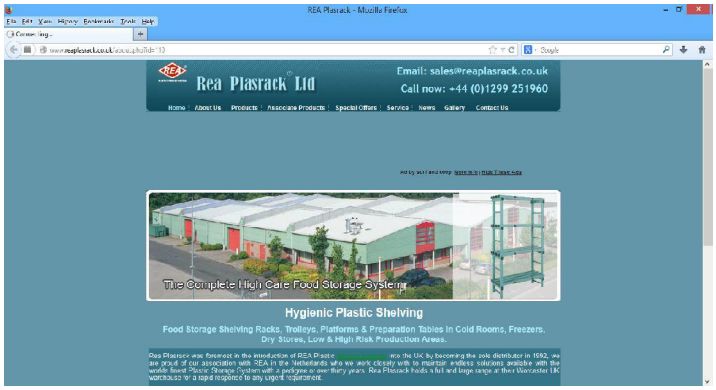

Fig 14: Demo: Actual site

Fig 15: Modifying the URL by ‘

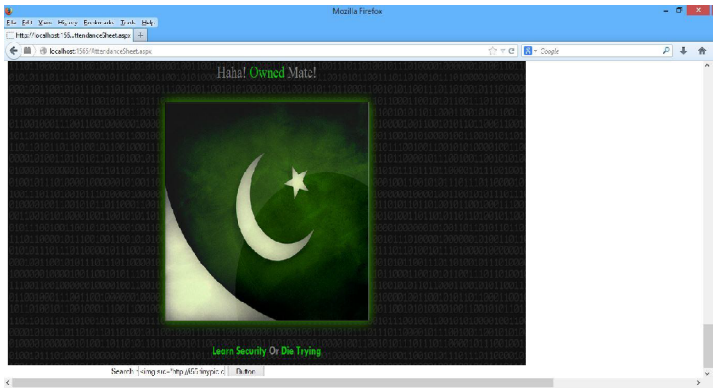

Fig 16: For XSS

VII. FUTURE SCOPE

While our initial evaluation demonstrates the promise of a context-aware approach to vulnerability scanning, it does highlight several limitations which form the foundation for future work in this area. First, we are not providing the patch for the detected vulnerable URL. To overcome this, we are planning to generate the patches and providing solutions in future. Second, rich area for future work is exploration of many more vulnerabilities, detecting them and providing solutions.

VIII. CONCLUSION

This project reveals the hidden solution to the major challenge in web security that is for the web vulnerability. In this project, we presented a Vulnerability scanner that aimed at detecting web application vulnerabilities. We have introduced our own crawler name “basic crawler” and described how the website is vulnerable to SQL Injection and

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 12, December 2018

XSS based on automatically generation of specially crafted request allowing the successful exploitation of detected vulnerabilities.

REFERENCES

[1] Kevin J Vella, "The True Nature of Web Application Security: The Role and Function of Black Box Scanners "21 Feb.2007; http://www.acunetix.com/websitesecurity/blackbox-scanners/

[2] 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

[3] Jan Tudor, "Web Application Vulnerability Statistics 2013"; June2013; whitepapers@contextis.co.uk

[4] L. K. Shar and H. B. K. Tan, "Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities," in *Proc. 34th Int. Conf. Software Engineering, 2012, pp. 1293–1296.*

[5] Luciano Barbosa and Juliana Freire. An adaptive crawler for locating hidden-web entry points. In Proceedings of the 16th international conference on World Wide Web, pages 441450. ACM, 2007.

[6] Denis Shestakov and Tapio Salakoski. On estimating the scale of national deep web. In Database and Expert Systems Applications, pages 780789. Springer, 2007.

[7] Balakrishnan Raju, Kambhampati Subbarao, and Jha Manishkumar. Assessing relevance and trust of the deep web sources and results based on inter-source agreement.ACM Transactions on the Web, 7(2): Article 11, 132, 2013.

[8] Martin Hilbert. How much information is there in the information society? Signicance, 9(4):812, 2012.

[9] Yeye He, Dong Xin, Venkatesh Ganti, Sriram Rajaraman, and Nirav Shah. Crawling deep web entity pages. In Proceedings of the 30 sixth ACM international conference on Web search and data mining, pages 355364. ACM, 2013.

[10] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin ."SmartCrawler: A Two-stage Crawler for Eciently Harvesting Deep-Web Interfaces" ,IEEE Transactions onServices Computing, Vol 99 Year 2015

[11] Zlatkovski, Dragi, Natasa Suteva, and Aleksandra Mileva. "SQL Injection Test System for Students." (2012):234-236.

[12] Wikipedia Link: https://www.owasp.org/index.php/Top_10_2013 (OWASP Top Ten 2013 Project)