ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

A Survey on Vulnerability Scanner for SQL Injection based on Deep Web Harvesting

Vrushali Narendra Bora

Student, Dept. of Computer Engineering, MCOERC, Nashik, Savitribai Phule Pune University, Maharashtra, India

ABSTRACT: Deep harvesting is a focused area of researchthese days. In deep websites contents lie behind search-able web interfaces. The deep websites are not registered to any search engine. In deep web harvesting framework site location and balanced in- site exploration carried out. This leads to wide coverage of data on sites and high efficiency for interfacing websites. In the proposed work, a web vulnerability scanner for SQL injection based on deep web harvesting is designed. In this system a web application security testing tool that audits web application by checking for vulnerabilities like SQL injection, directory access, injection of vulnerabilities and attacks is framed and analysed. The system will also focus on retrieval of deep web interfaces from large scale sites and on improving the performance of harvesting by increasing its rate.

KEYWORDS: Security,SQL injection, Internet Application, deep web harvesting, web crawler.

I. INTRODUCTION

The deep (or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. It is challenging to detect the deep web databases, because they are not registered with any search engines, are usually comparatively distributed, and keep constantly changing. The Deep Web has been acknowledged as a important division in the coverage of search engines because web crawlers employed by search engines rely on hyperlinks. Various accounts have assumed that the Deep Web has an structure of magnitude more data than the currently searchable World Wide Web Furthermore, the Deep Web has been a deep-standing challenge for the database community because it represents a large fraction of the structured data on the Web.

Website security is most important need in today's world, an enterprise and should be a priority in any organization. Progressively, hackers are concentrating their intensions on web-based applications: shopping carts, forms, login pages. Accessible anytime from anywhere in the world, insecure web applications provide easy approach to backend corporate databases and allow hackers to perform illegal activities using the attacked sites. A victim's webpage can be worn to fire acts. Hackers already have a wide repertoire of attacks that they regularly fire across organizations including SQL Injection, Cross Site Scripting, Directory Traversal Attacks, Authentication Attacks, Directory Enumeration and other exploits.

II. RELATED WORK

A. Locating deep web content sources

Locating deep web content sources A recent study shows that the harvest rate of deep web is less only 647,000 distinct web forms were found by sampling 25 million pages from the Google index . Generic crawlers are mainly developed for characterizing deep web and directory construction of deep web resources, that do not limit search on a specific topic, but attempt to fetch all searchable forms.. The Database Crawler in the Meta Query is designed for automatically discovering query interfaces. Database Crawler first finds root pages by an IP-based sampling, and then performs shallow crawling to crawl pages within a web server starting from a given root page. The IP based sampling avoids the fact that one IP address may have several virtual hosts, thus missing many websites. To overcome the drawback of IP based sampling in the Database Crawler, Denis propose a stratified random sampling of hosts to characterize national deep web, using the Host graph provided by the Russian search engine Yandex. I-Crawler mixed pre-query and post-query access for classification of searchable forms.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

B. Selecting Relevant sources

Existing hidden web directories, [4], [7] usually have low coverage for relevant online databases [3], which limits their ability in satisfying data access needs [3]. Focused crawler is developed to visit links to pages of interest and avoid links to of topic regions [1], [2], [3], [4]. Describe a best-first focused crawler, which uses a page classifier to guide the search [4]. The classifier learns to classify pages as topic-relevant or not and gives priority to links in topic relevant pages. However, a focused best-first crawler harvests only 94 movie search forms after crawling 100,000 movie related pages [3]. An improvement to the best-first crawler is proposed in [2], where instead of following all links in relevant pages, the crawler used an additional classifier, the apprentice, to select the most promising links in a relevant page. The baseline classifier gives its choice as feedback so that the apprentice can learn the features of good links and prioritize links in the frontier. The FFC [2] and ACHE [3] are focused crawlers used for searching interested deep web interfaces. The FFC contains three classifiers: a page classifier that scores the relevance of retrieved pages with a specific topic, a link classifier that prioritizes the links that may lead to pages with searchable forms, and a form classifier that filters out non-searchable forms. ACHE improves FFC with an adaptive link learner and automatic feature selection. Source Rank [2], [3] assesses the relevance of deep web sources during retrieval. Based on an agreement graph, Source Rank calculates the stationary visit probability of a random walk to rank results. Different from the crawling techniques and tools mentioned above, Smart Crawler is a domain-specific crawler for locating relevant deep web content sources. Smart Crawler targets at deep web interfaces and employs a two-stage design, which not only classifies sites in the first stage to filter out irrelevant websites, but also categorizes searchable forms in the second stage. Instead of simply classifying links as relevant or not, Smart Crawler first ranks sites and then prioritizes links within a site with another ranker.

C. Security mechanisms using vulnerability

1. Detection of Vulnerabilities by Security Scanner

Security scanners identify defects and weaknesses by a collection of signatures of known vulnerabilities. These signatures are updated regularly as new vulnerabilities are discovered. Like XSS and SQL injection are discover in search of vulnerabilities , the scanners execute lots of pattern variations adapted to the specific test in order to detect the vulnerability. There are two main approaches to testing web applications for vulnerabilities. The "white box" focus on consists of the testing of the source code of the web application. Static code analysis is a type of white-box analysis[3][4]. This can be done manually or by using code testing tools like FORTIFY [3], Code Secure, etc. These static analyser tools analyse source code to detect vulnerabilities, such as SQL injection and cross-site scripting. The black-box vulnerability scanner without knowing the internal design of the web application uses fuzzy techniques over the web HTTP requests, simulates numerous scenarios such as hackers' intentional attacks or general users' inadvertent attacks, and provides an automatic way to find for, neglect the repetitive and monotonous task of doing hundreds or even thousands of tests by hand for each vulnerability type. There are many commercial web vulnerability scanners for black-box testing such as Acunetix Web Vulnerability Scanner [1], HP Web Inspect [4], IBMAppScan [5]. The testing results of vulnerabilities for web applications are quite different from scanner to scanner. According to the survey presented in [3], black-box testing is the second most used technique to calculate the effectiveness of security. In our experiments, we use our proposed approach to calculate four famous "black box" commercial scanners, App Scan, Web Inspect, Paros, and Acunetix.

2. Web Vulnerability Scanner Evaluation Criteria

Vulnerability scanners are considered as a solution for detecting vulnerabilities and security threats in web applications. Among the studies focusing on tools evaluations [2][3], the Web Application Security proposed "Web Application Security Scanner Evaluation Criteria project to come up with a set of detailed evaluation criteria and a framework for conducting a formal scanner calculation. The goal of the WASSEC is that for the tools given to users they need to conduct a solid evaluation and make their own informed resolve on which scanner(s) best meet their needs. P.E. Black. proposed guidelines for describe the functional specifications of Source Code Security Analysis Tool and Web Application Security Scanner in NIST Special Publication 500-268[7] and 500-269[6]. Through the development of tool functional specifications, project aims to better quantify the state of the art for contrast classes of software security assurance tools. The documents constitute a specification for a particular class of software assurance tool, which is referred to here as a web application security scanner. By examining the steps in scanning processes, we can reasonably assume that costs of vulnerability scanner include construction cost, functioning cost, and analysis cost, with operation and analysis ones being the main parts in cost evaluation. These processes are generally labour-intensive and often

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

involve substantial human resources, including developers, domain experts, and security experts. There are several studies focusing on reducing cost in vulnerabilities detection, such as [5][6]. However, the issue of redundant alerts have not been considered in the previous research. In general, when a certain defect is found repeatedly, the developers would spend double effort to solve it. In this paper, we proposed a cost-effective evaluation approach to evaluate vulnerability scanner by considering issue of redundant alerts.

III. PROPOSED SYSTEM

Manual vulnerability auditing of all your web applications is complex and time-consuming, since it generally involves processing a large volume of data. It also demands a high level of expertise and the ability to keep track of considerable volumes of code used in a web application. In addition, hackers are constantly finding new ways to exploit your web application, which means that you would have to constantly monitor the security communities, and find new vulnerabilities in your web application code before hackers discover them.

It is an web application security testing tool that analyse your web applications by checking for exploitable vulnerabilities. In general, Web Vulnerability Scanner scans any website or web application that is in stock via a web browser and uses the HTTP/HTTPS protocol. Web Vulnerability Scanner offers a secure and unique solution for analysing of-the-shelf and custom web applications including those utilizing JavaScript, AJAX and Web applications.

A. URL Crawling.
B. Search Engine
C. 3rd Party Database
D. Domain Reputation
E. CMS Scan
F. URL Scan

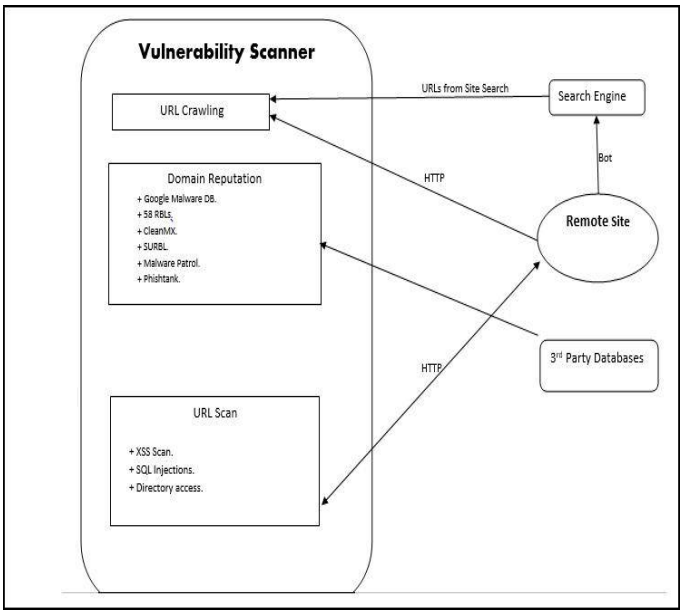

Fig.1. System Architecture

IV. CONCLUSION AND FUTURE WORK

The proposed work, a web vulnerability scanner for SQL injection based on deep web harvesting is designed. In this system a web application security testing tool that audits web application by checking for vulnerabilities like SQL injection, directory access, injection of vulnerabilities and attacks is framed and analyzed. The system will also focus

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

on retrieval of deep web interfaces from large scale sites and on improving the performance of harvesting by increasing its rate.

REFERENCES

1. Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin 'SmartCrawler: A Two-stage Crawler forEfficiently Harvesting Deep-Web Interfaces',IEEE Transactions on Services Computing, Volume 99,2015.
2. 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,ACM:2007.
3. Denis Shestakov and TapioSalakoski,'On estimating the scale of national deep web'In Database and Expert Systems Applications, Springer:2007.
4. Balakrishnan Raju, KambhampatiSubbarao, and JhaManishkumar, 'Assessing relevance and trust of the deep web sources and results based on inter-source agreement',ACM Transactions 2013.
5. Olston Christopher and Najork Marc, 'Web crawling Foundations and Trends in Information Retrieval',2010.
6. Larry Suto, 'Analyzing the Accuracy and Time Costs of Web Application SecurityScanners,SanFrancisco', 2010.
7. J. Fonseca, M. Vieira, and H. Madeira, 'Testing and comparing web vulnerability scanning tools for sql injection and xss attacks', 2007.

BIOGRAPHY

Vrushali Narendra Bora A M.E student in a Computer Engineering Department MCOERC Nashik, SavitribaiPhule PUNE University, pune. Her research interest are Web Security, Information Security, Cyber Security.