Design and Implementation of SQL Injection Vulnerability Scanning Tool

Juanjuan Zhao and Changhua Liu
Department of Mathematics and Computer, Wuhan Polytechnic University, P.O. Box 430023.
Email: zhaojj0228@163.com and liuch@whpu.edu.cn

Abstract. According to the "Top Ten Security Vulnerabilities List" (OWASPTop 10) released by OWASP in 2017, SQL injection attacks are still at the top of the list, and there are many ways of SQL injection attacks, which cause great harm. Although there are many vulnerability scanning tools, there is still a high rate of false negatives. Aiming at the current problems of SQL injection vulnerability detection, this paper proposes a scanning tool for SQL injection vulnerabilities. First, use the crawler framework scrapy to obtain the URL associated with the form and the a tag, and segment the URL based on the improved simhash algorithm. Deduplicate the link, then analyze the injection point to modify the URL parameter value injection test, and determine whether there is a vulnerability based on the response result of the server. The experimental results show that the detection method achieves a 96.50% URL deduplication rate in the crawler module, which greatly reduces the rate of false negatives. It is more suitable for detecting whether a website has a SQL injection vulnerability.

1. Introduction
With the advent of web2.0, the Internet brings convenience to people, but it also brings many harms. Common examples are privacy data leaks, web page tampering, damage to hard disk data, paralysis of the system, and servers being remotely controlled. According to the "Top Ten Security Vulnerabilities List" published by OWASP in 2017 [1], SQL injection attacks are still the most harmful vulnerabilities. At present, many websites rely too much on WAF defense technology and ignore the improvement of code quality [2]. It is precisely because some developers lack certain security awareness that they do not pay attention to standardizing the writing of SQL statements and filtering special characters in the process of program development, resulting in the client can submit some SQL statements through global variables POST and GET for normal execution.

In the direction of researching SQL injection attack vulnerability scanning tools, many scholars have done a lot of research in recent years, the main purpose of which is to improve vulnerability scanning efficiency and detection accuracy. Tang Shujing et al. [3] introduced multithreading into the secondary development of Webkit browser engine technology to improve the efficiency of crawler, and proposed a page similarity algorithm based on edit distance to improve the detection rate of SQL injection vulnerabilities. S Nagasundari et al. [4] trained ResNet model and used ResNet model to identify different types of SQL injection attacks in order to improve the accuracy of vulnerability scanning. Kong Deguang et al. [5] proposed to segment SQL statements based on simhash algorithm to build simhash fingerprint database, and to use white list strategy to judge the legitimacy of each SQL statement to reduce the risk of hackers bypassing the defense system. Combined with the high efficiency of Simhash algorithm, the detection method can defend against unknown SQL injection vulnerabilities. Jia Xiaoyu et al. [6] proposed SVCBF algorithm to compress links on the basis of the
original Bloom filter algorithm, and mapped the processed URL into a variable-length bit vector counter through hash, thus resolving the defect that Bloom filter cannot delete and update URLs and to improve the accuracy and efficiency of deduplication.

On the basis of summing up previous researches, this paper proposes a URL deduplication algorithm using simhash algorithm for URL duplication. The website URL[7] is crawled by using the scrapy framework combined with the breadth-first search method. In the crawling process, simhash algorithm is used to remove duplicates for identical URLs to avoid duplication of requested pages. Analyzing the requested page, extracting all the requests with parameters on the page, filtering URLs with similar parameters by using simhash algorithm for all the extracted requests, performing parameter analysis on the remaining URLs, injecting illegal parameters and then sending to the server, and judging whether there is sql injection vulnerability based on the return value.

2. System Design and Analysis
SQL injection vulnerability scanning is generally divided into two stages. The first stage is the crawler stage, which crawls all URLs with parameters in the website. The second stage is the detection stage, which uses the URL obtained by the crawler to try to change parameters. The response of the server determines whether there is an injection attack [8]. Most of the current SQL injection vulnerability scanning tools use a regular expression matching method to filter URLs in the crawler module, which may cause some URLs to be missed during the deduplication process, resulting in a higher false positive rate.

Based on the above problems, aiming at the problems of URL duplication in crawler and URL similarity in injection attack, this paper respectively improves the word segmentation function of simhash algorithm and designs a vulnerability scanning system, which can greatly improve the accuracy of detection. The main implementation steps of the system are: the URL of the page obtained by the website crawler to be detected, deduplication of the obtained URL and continuing to crawl, use the deduplicated result set to continue to deduplicate the URL similarity, and modify the URL link parameters to the final result set perform an injection test.

2.1. Design of Crawler Module
The crawler module is mainly responsible for crawling the URLs related to SQL injection in the website and sending them to the detection system for detection. This system uses the scrapy framework to crawl links [9]. scrapy is a very mature crawler framework. It is used in many crawler-related articles. After obtaining the URL list, it uses the simhash algorithm to deduplicate the URL. The simhash algorithm is a locally sensitive hash algorithm proposed by Charikar in 2002[10]. In 2007 Manku proved that the simhash algorithm is very useful in identifying approximate duplicates in billions of web pages[11]. The algorithm works as follows:

For two given variables x, y, the hash function h always satisfies the following formula: $Pr_{hash}(h(x) = h(y)) = sim(x, y)$. In formula, Pr represents the possibility of h (x) = h (y), and $sim(x, y) \in [0,1]$ is a similarity function. Generally, the Jacobian function $Jac(x, y)$ To represent the similarity of the variables x, y, and $sim(x, y)$ is expressed as follows: $sim(x, y) = Jac(x, y) = \frac{|x \cap y|}{|x \cup y|}$. The algorithm is divided into five steps: word segmentation, hash, weight, merge and dimension reduction. The specific process is as follows, with "https://www.baidu.com/s?ie=utf-8&wd=sqlInjection" as an example:
If there is a URL collection: “http://xxx.com/_visitcountdisplay/a/b?siteId=64&article=58978” (1) and “http://xxx.com/_visitcountdisplay/b/a?siteId=64&article=58978” (2).

The original simhash algorithm's deduplication results will only leave (1), but the improved algorithm's deduplication results will leave (1) and (2), because the original algorithm will consider that the two URLs are exactly the same. Therefore, this will cause a lot of misses in the final URL collection, which will cause a higher false positive rate of detection results.

2.2. Design of Detection Module
The vulnerability scanning tool designed in this paper scans all forms, input boxes and hyperlinks in the web page, and uses simhash algorithm to remove similarities from the URL list obtained by the crawler (other parts are the same but only URLs with different parameter values), such as the following two URLs: “http://xxx.com/_visitcountdisplay?siteld=64&type=3&article=58963” and “http://xxx.com/_visitcountdisplay?siteld=64&type=3&article=58964”.

The parameter types of the two links are exactly the same, but the values of the parameters are different. In order to improve the scanning speed, it is not necessary to do injection tests for this type of links one by one, so it is necessary to remove similar URL. Simhash algorithm is currently recognized as the best deduplication algorithm[12], so this paper uses this algorithm in crawler and detection module to improve detection efficiency and reduce false positive rate.

3. Experimental Results and Analysis

3.1. Experimental Data Collection and Results Display
This article use python language to complete crawling of shooting range URLs, 9345 URLs were obtained, 3368 URLs were left after the crawler module removed repetitions, based on which the algorithm was used to remove similar links and 2399 links were used to inject tests, scanning out 2 SQL injection vulnerabilities. The experimental results were written and run in software vscode 1.41 using python 3.7.6. the computer hardware configuration is macbook pro (retina,13-inch, early 2015), 2.7 GHz dual-core Intel Core i5 with 16 GB of memory. The following two tables show the comparison between the original simhash algorithm and the improved running results:

| Word segmentation | Set weight | Hash value | Vector weighting |
|-------------------|------------|------------|-----------------|
| hittos            | 1          | 10110      | (1,1,1,1,1)     |
| www.baidu.com    | 1          | 11000      | (1,1,1,1,1)     |
| /s                | 2          | 00101      | (2,2,2,2,2)     |
| ie                | 2          | 01010      | (2,2,2,2,2)     |
| wd                | 2          | 11010      | (2,2,2,2,2)     |

![Diagram of simhash algorithm steps]

![Diagram of simhash algorithm steps]

(1) If there is a URL collection: “http://xxx.com/_visitcountdisplay/a/b?siteId=64&article=58978” (1) and “http://xxx.com/_visitcountdisplay/b/a?siteId=64&article=58978” (2).

(2) The original simhash algorithm's deduplication results will only leave (1), but the improved algorithm's deduplication results will leave (1) and (2), because the original algorithm will consider that the two URLs are exactly the same. Therefore, this will cause a lot of misses in the final URL collection, which will cause a higher false positive rate of detection results.

(3) The parameter types of the two links are exactly the same, but the values of the parameters are different. In order to improve the scanning speed, it is not necessary to do injection tests for this type of links one by one, so it is necessary to remove similar URL. Simhash algorithm is currently recognized as the best deduplication algorithm[12], so this paper uses this algorithm in crawler and detection module to improve detection efficiency and reduce false positive rate.

(4) The experimental results were written and run in software vscode 1.41 using python 3.7.6. the computer hardware configuration is macbook pro (retina,13-inch, early 2015), 2.7 GHz dual-core Intel Core i5 with 16 GB of memory. The following two tables show the comparison between the original simhash algorithm and the improved running results:
3.2. Result Analysis

According to the above results, it can be seen that the number of URLs remaining after the crawler deduplication part is deduplicated by using the original algorithm will be smaller than that of the algorithm used in this paper, thus it can be seen that the original deduplication algorithm will mistakenly delete non-duplicated links when deduplication. From the vulnerability scanning results, it can be seen that the remaining number after removing similar URL using the original algorithm is more than that after removing similarity using the algorithm in this paper, which indicates that the algorithm in this paper is more thorough in removing similarity. In this paper, according to different requirements, the two modules modify the word segmentation function of different simhash algorithms on the basis of the original algorithm to improve the accuracy of missed scanning. To sum up, the scanning tool can detect SQL injection vulnerabilities in websites more accurately and efficiently.

4. Conclusions

This article studies and analyzes the causes and detection methods of SQL injection vulnerabilities. Based on the current shortcomings of SQL injection vulnerability detection, a SQL injection vulnerability detection method using simhash algorithm is proposed. The main contribution of this article is that there is no need to analyze the SQL syntax, just study the composition structure of the URL, and combine the simhash algorithm to modify the algorithm's word segmentation function according to the different needs of the crawler stage and detection stage to make the URL deduplication rate increase to more than 95%. It is proved that this detection method can detect website vulnerabilities more accurately and in time, so as to protect the website security to the greatest extent.

The detection methods in this paper still have the following deficiencies, and further research is needed: there are many ways of SQL injection. This article only studies two types of injection. In the following, some injection methods need to be added to reduce the false negative rate of SQL injection vulnerability scanning. The crawler module URL deduplication weight setting needs to be adjusted for the similarity of pages on different websites, and it is necessary to repeatedly test to determine the weight. At present, the injection success rate is relatively low, and static injection is used. Whether the parameters are properly modified affects the success rate of vulnerability discovery to a large extent.

5. References

[1] Tan Jun. OWASP releases top ten web application security risks [J]. Computer and Network, 2017, 23: 52-53.
[2] Wang Qi. Research on vulnerability scanning technology for web application [D]. Jiangsu Province: Nanjing University of Posts and Telecommunications, 2016.
[3] Tang Shujing. Research and design of a web-based vulnerability scanning system [D]. Beijing: Beijing University of Posts and Telecommunications, 2016.
[4] Nagasundari S, Honnavali P B. SQL Injection Attack Detection using ResNet[C]//2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2019: 1-7.

[5] Kong Deguang. SQL injection attack detection method based on Simhash algorithm [J] .Application Research of Computers, 2019,23: 52-53.

[6] Jia Xiaoyu. Research and design of automated scanning tool for SQL injection based on improved crawler technology [D]. Beijing: Beijing University of Posts and Telecommunications, 2018.

[7] Ye Mengxiong. Design and Research of Web-based SQL Injection Vulnerability Scanning System [J]. Electronic Design Engineering, 2019, 27 (16): 20-28.

[8] Zhai Han. Design and Implementation of Web Security Scanning Tool Based on Web Crawler [D]. Beijing: Beijing University of Posts and Telecommunications, 2018.

[9] Wang Bing. Design and implementation of distributed crawler system based on Scrapy framework [D]. Anhui: Hefei University of Technology, 2019.

[10] Charikar M S. Similarity estimation techniques from rounding algorithms[C]//Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. ACM, 2002: 380-388.

[11] Manku G S, Jain A, Das Sarma A. Detecting near-duplicates for web crawling[C]//Proceedings of the 16th international conference on World Wide Web. ACM, 2007: 141-150.

[12] Lee K M. Locality-sensitive hashing techniques for nearest neighbor search[J]. International Journal of Fuzzy Logic and Intelligent Systems, 2012, 12(4): 300-307.