A XSS Attack Detection Method Based on Subsequence Matching Algorithm

Zhang Jingyu*
Information Engineering University
Zhengzhou, Henan，China
BITzhangjingyu@163.com

Hu Hongchao
l Information Engineering University
Zhengzhou, Henan，China
1309194701@qq.com

Huo Shumin
Information Engineering University
Zhengzhou, Henan，China
huoshumin123@163.com

Li Huanruo
l Information Engineering University
Zhengzhou, Henan，China
viaviavialhr@outlook.com

***Abstract*—XSS vulnerabilities are one of the main threats to current Web security, and measures must be taken to reduce the growing threat of XSS attacks. This research proposes a detection technique using a subsequence matching algorithm. The key point of the matching algorithm during detection is to find the common subsequence of the input parameters and the generated data, and set a threshold to limit the length of the common substring. The results obtained show that the proposed technique can successfully detect XSS vulnerabilities. Therefore, the technology proved to be more effective in detecting XSS attacks.**

Keywords- Web application vulnerability; Cross Site Scripting ;XSS; dynamic analysis;static analysis ;subsequence matching;DOM XSS

I. INTRODUCTION

With the continuous development of Web technology, browsers, as the main carrier of application and content dissemination, provide users with many conveniences: search engines, social networking sites, online shopping, online transactions, etc.[1]. With the rapid development of web application technology, more and more frequent web application vulnerability are attracting enough attention. Web application vulnerability are when attackers use browsers or attack tools to send special requests to the Web server in URLs or other input areas (such as forms, etc.) to discover vulnerabilities in Web applications, thereby further manipulating and controlling the website, viewing, modify unauthorized information[2].

Cross Site Scripting(abbreviated as XSS),is an attacker inserting malicious Script code into a Web page. When the user browses the page, the Script code embedded in the Web will be executed, so as to achieve the purpose of maliciously attacking the user, and the attacker can get higher permissions (such as performing some operations), private web content, sessions, cookies and other content. XSS vulnerability account for about 40% of all web vulnerability, and this attack is one of the most widely used web attack techniques. In the past ten years, XSS has been one of the top three web vulnerability in OWASP for web application vulnerability[3]. This is because vulnerability usually occur on a global scale. The World Hacking Incident Database released a ranking of the percentage of web vulnerability in 2018. XSS is one of the top ten types of vulnerability.

Static analysis and dynamic analysis are common methods to detect XSS vulnerabilities[4]. Static analysis refers to obtaining information about possible vulnerabilities by analyzing the source code of a program. Due to high efficiency and accuracy, many web application security researches have shifted their focus to static analysis, but static analysis requires the source code of the target website. Dynamic analysis is mainly to simulate the process of an attacker's attack. Write a script containing specific code to the injection point, and then analyze the response content of the page returned by the server. If there is specific data in the response content, it indicates that there is an XSS attack. Dynamic detection uses a large amount of test load to reduce the false negative rate of test results, but too much test load will lead to low detection efficiency.

In order to improve the false negative rate and low detection efficiency of XSS dynamic detection methods, this paper proposes an XSS leak detection model based on subsequence matching.

II. TYPES OF XSS VULNERABILITY

XSS refers to a malicious attacker inserting malicious html code into a web page. When the user browses the page, the html code embedded in the web will be executed to achieve the special purpose of the malicious user. Common XSS vulnerability are usually divided into three categories: Reflected XSS, stored XSS and DOM XSS[5].

A. Reflected XSS

Reflected XSS simply reflects the data entered by the user from the server to the user's browser, the flow chart is shown in Figure 1. To exploit this vulnerability, the attacker must in some way induce the user to visit a carefully designed *URL* (malicious link) in order to carry out the attack. Reflected XSS usually appears in functions such as search.[6] It needs to be triggered by the attacker by clicking on the corresponding link, and is greatly affected by defense methods such as *NoScript*, so it is less harmful than the Stored XSS.

978-0-7381-1083-7/21/$31.00 ©2021 IEEE

2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID) | 978-1-6654-1537-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/AIID51893.2021.9456515

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on August 16,2023 at 10:31:04 UTC from IEEE Xplore. Restrictions apply.

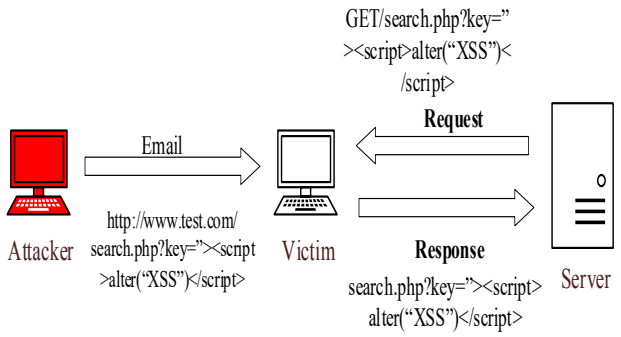

Figure 1: Reflected XSS attack flow chart

B. *Stored XSS*

Stored (or HTML injection/persistent) XSS vulnerability most often occur on websites or web mail sites driven by community content and do not require special links to execute[7]. Hackers only need to submit the XSS exploit code (reflective XSS is usually only in the URL) to a site where other users may visit.The flow chart is shown in Figure 2.These areas may be blog comments, user comments, message boards, chat rooms, HTML emails, wikis, and many other places. Once the user visits the infected page, execution is automatic.

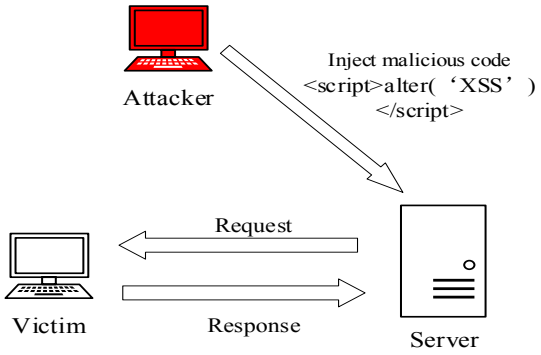

Figure 2: Reflected XSS attack flow chart

C. *DOM XSS*

The XSS attack formed by modifying the DOM node of the page, the DOM XSS is based on the DOM document object model. For the browser, the DOM document is an XML document. With this standard technology, the DOM can be easily accessed through JavaScript.

When it is confirmed that there is a DOM-type XSS attack in the client code, it is possible to induce a user to visit the URL constructed by himself. The steps are similar to the reflection type, but the only difference is that the constructed URL parameters do not need to be sent to the server side, so it can achieve the detection effect of bypassing WAF and avoiding the server.

The main hazards of XSS security vulnerabilities are[8]：

（1） Website hacking: first embed malicious attack code into the web application, and use some vulnerabilities in the browser to execute the code. When a user browses to a web page with malicious attack code, a Trojan horse is planted in the user's computer.

（2） Phishing deception: fraudulent users by imitating the URL addresses of well-known websites and constructing similar page content to steal users' private information and damage their interests.

（3） Stealing user information: This is the most common hazard of XSS security vulnerabilities. It means that attackers use XSS vulnerabilities in web applications to intercept user cookie information, and then steal user private information.

（4） Sending spam: An attacker can use XSS vulnerabilities to send spam to certain user groups, resulting in a serious waste of network resources and affecting the use of other normal users.

（5） XSS worms: Worms can use the network environment to achieve autonomous self-replication and infection, which in turn causes the theft of users' private information, website hanging on horses and other hazards.

It can be seen that XSS vulnerabilities[10], as one of the main Web security vulnerabilities, have extremely destructive power. How to accurately detect XSS vulnerabilities in Web applications and better defend against XSS vulnerability has become a hot topic in the research of Web security vulnerabilities.

III. A Detection Model of XSS Attack

According to the XSS attack principle, there is a direct correlation between the input parameters and the data generated in the feedback page: the injected script will appear in both the HTTP request and the HTTP response[9]. They will exist in both input parameters and generated data. Therefore, detecting XSS vulnerabilities only needs to match the input parameters and generate data under the appropriate approximation. However, attackers generally carefully construct input parameters in order to bypass the filtering module of the Web application, so that the input parameters and the generated data are different in form.

Therefore, we propose a vulnerability detection method based on subsequence matching, which can effectively identify the variant form of input parameters.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on August 16,2023 at 10:31:04 UTC from IEEE Xplore. Restrictions apply.

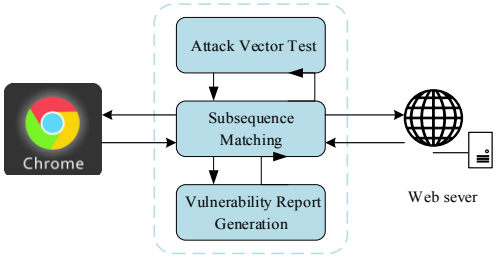

Figure 3:A XSS detector model architecture

The vulnerability detection module is mainly composed of three modules: attack vector test, subsequence matching, and vulnerability report generation[11]. The main functions are as follows:

A. *The Attack Vector Test Module*

This part mainly uses the *URL* and input vector information in the *TestVector* table to construct and send *HTTP* requests with attack vectors. The *TestVector* table is used to record input vector related information, and its table structure is shown in TABLE I.

The *inUrl* field indicates the *URL* where the input vector is located, the *inType* field indicates whether the *URL* is a link similar to "*url?id=xxx*" or a link with a *form*, the *inAction* field indicates the link address for form submission, and the *inMethod* field indicates that the *URL* is a get request or a post request, the *inVectors* field represents the input vector group.

TABLE I. STRUCTURE OF THE TESTVECTOR TABLE

Structure of the TestVector table			
name	***types***	***length***	***Is null?***
inURL	varchar	100	Yes
inType	int	2	Yes
inAction	varchar	100	No
inMethod	varchar	40	Yes
inVectors	varchar	100	Yes

B. *The Subsequence Matching Module*

When the input parameter is the basic attack vector, the traditional string matching algorithm may be effective. In general, when performing XSS vulnerability, in order to bypass the filtering module of the Web application, the attacker will use the mutation rule set to carefully construct various mutation attack vectors as input parameters, so that the input parameters and the generated data are different in form. This leads to the failure of traditional string matching algorithms.

The key point of the matching problem during detection is to find the common subsequence of the input parameters and the generated data, and set a threshold to limit the length of the common substring. Only when its length exceeds this threshold, can it be determined that the tested input vector has XSS vulnerabilities. The reason for setting the threshold is the malicious script length will not be lower than a certain length, that is the set threshold.

C. *The Vulnerability Report Generation Module*

The XSS attack report generation module is mainly based on the detection results of the previous modules to generate related vulnerability detection reports and record logs. Vulnerability reports generate reports by recording the intrusion time of the XSS attack, the content of the attack structure, and the means of attack, so that the developer can perform subsequent maintenance and recovery operations.

IV. EXPERIMENT AND TEST

This paper proposes a vulnerability detection method based on subsequence matching, and designs an XSS vulnerability detection model. Therefore, this experiment needs to verify the accuracy and low false negative rate of the vulnerability detection method based on subsequence matching:

A. *Experimental environment*

The software and hardware environment required for the experiment is shown in the table：

TABLE II. SOFTWARE AND HARDWARE ENVIRONMENT

software and hardware environment			
software		***hardware***	
CPU	Intel(R) Core(TM)i5-7300CPU@2.50GHz	OS	Windows10
Memory	8GB	Java development platform	MyEclipse8.5、JDK1.7.0_15

B. *Experiments and results analysis*

In order to verify the accuracy and low miss rate of vulnerability detection method based on subsequence matching, this experiment will respectively test XSS vulnerability detection method of website ***a.com*** using string matching and subsequence matching.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on August 16,2023 at 10:31:04 UTC from IEEE Xplore. Restrictions apply.

XSS Detector

URL	Method	ParamName	Result
https://www.a.com/admin/sc909	post	title	NO
http://a.aazz.com/admin/directorup.php	post	research	NO
http://a.aazz.com/admin/directorup.php	get	pkey	NO
https://www.a.com/sdmin/post/sc909	post	pkey	NO
https://a.com/admin/trier.de/db/	post	atitle	XSS
https://a.web.com/admin/hifui	get	stitle	NO
https://a.com/workers.dev/help	get	work	NO

Figure 4: Subsequence matching detection result

Without XSS Detector

URL	Method	ParamName	Result
https://www.a.com/admin/sc909	post	title	XSS
http://a.aazz.com/admin/directorup.php	post	research	XSS
http://a.aazz.com/admin/directorup.php	get	pkey	XSS
https://www.a.com/sdmin/post/sc909	post	pkey	NO
https://a.com/admin/trier.de/db/	post	atitle	XSS
https://a.web.com/admin/hifui	get	stitle	NO
https://a.com/workers.dev/help	get	work	XSS

Figure 5: String match detection result

From the statistics of the vulnerability detection results in Figure 4 and Figure 5, it can be seen that the vulnerability detection method based on string matching cannot be detected, and the vulnerability detection party based on subsequence matching can generally be detected.

At the same time, the false negative rate of subsequence matching is much lower than that of string matching, and the number of sending http requests is also reduced. It can be seen that the vulnerability detection method based on subsequence matching proposed in this paper is more accurate and has a low false negative rate.

V. Conclusion

With more and more application areas of web applications, vulnerability against XSS vulnerabilities are one of the most common attack methods. This paper studies the existing XSS attack detection methods, analyzes the advantages and disadvantages of static detection and dynamic detection, proposes an XSS vulnerability detection method based on subsequence matching, and designs an XSS vulnerability detection model based on subsequence matching.

By introducing the function and realization of each module in detail, and analyzing and comparing the experimental results, this model can complete the preliminary detection of XSS vulnerabilities.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0804004, and in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61521003.

References

[1] B.Soewito, F.E. Gunawan.Prevention structured query language injection using regular regular expression and escape string[J].Procedia Comput. Sci,2018,135:678-687.

[2] OWASP Top Ten.Top 10 Web Application Security Risks [EB/OL]. https://owasp.org/www-project-top-ten/,2020-12-1.

[3] Germán E. Rodríguez,Jenny G. Torres,Pamela Flores,Diego E. Benavides. Cross-site scripting (XSS) attacks and mitigation: A survey[J]. Computer Networks,2020,166.

[4] Chenghao Li,Yiding Wang,Changwei Miao,Cheng Huang. Cross-Site Scripting Guardian: A Static XSS Detector Based on Data Stream Input-Output Association Mining[J]. Applied Sciences,2020,10(14).

[5] Joaquin Garcia-Alfaro, Guillermo Navarro-Arribas.A Survey on Detection Techniques to Prevent Cross-Site Scripting Attacks on Current Web Applications[J].2007,287-298.

[6] Mohammad Reza Faghani,Uyen Trang Nguyen. A Study of XSS Worm Propagation and Detection Mechanisms in Online Social Networks.[J]. IEEE Trans. Information Forensics and Security,2013,8(11).

[7] Bakare K Ayeni, Junaidu B Sahalu, Kolawole R Adeyanju. "Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System[J]. Inventi Impact - Fuzzy Systems,2019,2019.

[8] Pooja Chaudhary, B.B. Gupta. Plague of cross-site scripting on web applications: a review, taxonomy and challenges[J]. Int. J. of Web Based Communities,2018,14(1).

[9] Gurpreet Kaur,Bhavika Pande,Aayushi Bhardwaj,Gargi Bhagat,Shashank Gupta. Efficient yet Robust Elimination of XSS Attack Vectors from HTML5 Web Applications Hosted on OSN-Based Cloud Platforms[J]. Procedia Computer Science,2018,125.

[10] Vamsi Mohan , Dr. Sandeep Malik. Secure Web Applications Against Cross Site Scripting XSS A Review[J]. Journal of Trend in Scientific Research and Development,2017,2(1).

[11] Swaswati Goswami,Nazrul Hoque,Dhruba K. Bhattacharyya,Jugal Kalita. An Unsupervised Method for Detection of XSS Attack[J]. International Journal of Network Security,2017,19(5).

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on August 16,2023 at 10:31:04 UTC from IEEE Xplore. Restrictions apply.