Development of Mathematical Algorithm for Detecting XSS Attacks on Web Applications

KOMIL FIKRATOVICH KERIMOV[1], ZARINA ILDAROVNA AZIZOVA[2]
[1]Department of System and Application Programming,
[2]Department of Information Security,
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi,
108 Amir Temur Avenue, Yunusabad paradise, 100084, Tashkent,
REPUBLIC OF UZBEKISTAN

Abstract: - The widespread usage of web applications has led to an increase in security threats, with Cross-Site Scripting (XSS) attacks being one of the most prevalent and damaging. Detecting and mitigating XSS attacks is crucial to ensure the integrity and confidentiality of sensitive user data. This article presents the mathematical algorithm and a way to identify XSS attacks using a bounded function from below, which depends on the input string, and highlights its potential impact in bolstering web application security. To construct this function, we used special characters and keywords that are frequently found in the construction of XSS attacks.

Key-Words: - XSS, vulnerability, signatures, information security, artifacts, testing.

Received: October 19, 2023. Revised: April 5, 2024. Accepted: May 22, 2024. Published: June 24, 2024.

1 Introduction

Nowadays the internet has become an integral part of our daily lives, with web applications serving as the backbone of various online services. Web applications are increasingly targeted by malicious actors, particularly through Cross-Site Scripting (XSS) attacks. These attacks involve injecting harmful scripts into web pages, enabling data theft, session manipulation, and further exploitation. Successful XSS attacks can result in financial losses, reputational damage, and compromised trust.

Traditional approaches to detecting XSS attacks have relied on signature-based methods or pattern matching techniques. While these techniques can be effective to a certain extent, they often struggle to detect sophisticated and obfuscated XSS payloads. Furthermore, the dynamic nature of modern web applications poses challenges for static analysis-based detection methods. To address these limitations, we propose a novel mathematical algorithm for detecting XSS attacks on web applications. Our approach combines the principles of XSS vulnerability detection based on the specific weight of each attack symbol signature.

In this paper, we propose a mathematical modeling and a method to identify XSS attacks using a bounded function from below, which depends on the input string. To construct this function, we used special characters and keywords that are often found in the construction of XSS attacks. In the proposed method, it is possible to detect XSS attacks using a single special character or a single keyword. However, it can be shown experimentally that the proposed detection method using a set of multiple characters and words can detect the vulnerability of a type of XSS attacks more accurately.

1.1 Related Research Papers on XSS Attacks Detection

Article, [1], proposes a machine learning approach for detecting XSS attacks, which relies on diverse and representative training data for reliable results. Static analysis methods, [2], detect XSS vulnerabilities during development but require manual verification and adjustment of rules for improved accuracy. Combining dynamic analysis and machine learning, [3], enhances detection accuracy but poses challenges in scalability and adapting to evolving attack techniques.

Applying NLP techniques to enhance XSS detection shows promise but faces challenges with obfuscated payloads, [4]. Symbolic execution detects DOM-based XSS attacks by analyzing JavaScript code but can be computationally expensive and requires careful handling of complex code, [5]. Signature-based detection systems are effective for known attacks but may struggle with new patterns, requiring regular updates, [6]. Evaluating WAFs' effectiveness in detecting XSS attacks is crucial, considering performance, accuracy, and flexibility. Regular updates and customization of WAF rules based on known attack patterns are essential, [7]. XSS attacks inject malicious code into HTTP requests. Existing prevention techniques may not work against unknown attacks, emphasizing the need for new prevention methods, [8]. In [9], a mathematical method for detecting XSS type vulnerabilities is proposed based on the specific weight of each attack symbol signature.

These research works provide valuable insights into the different approaches and techniques for XSS attack detection. However, it is important to consider the limitations, challenges, and potential trade-offs associated with each approach.

Mathematical algorithm for detecting attacks on a web application using information related to the frequency and importance factor of attack symbols does not require complex mathematical techniques like optimization etc. Since the proposed method is based on the processing of available types of real-world vulnerabilities of XSS kind, and is general in nature, we can expect that our proposed algorithm can be used as an online cross-site scripting attack detection platform.

2 Mathematical Algorithm for Detecting XSS Attacks

In the field of web application security, detecting and mitigating XSS attacks is of paramount importance. XSS attacks exploit vulnerabilities in web applications to inject malicious scripts into web pages, compromising user data and system integrity. This section provides a concise explanation of XSS attack detection techniques, highlighting their significance in enhancing web application security.
XSS attack detection involves identifying and preventing the execution of malicious scripts injected into web pages.

Effective XSS attack detection combines multiple techniques to provide comprehensive protection. Static analysis helps identify vulnerabilities during the development phase, while dynamic analysis and machine learning-based detection offer real-time monitoring and adaptive capabilities. Signature-based detection and WAFs add another layer of defense by leveraging predefined rules and monitoring network traffic. XSS attack detection techniques play a vital role in safeguarding web applications and user data from malicious exploitation. By employing a combination of static and dynamic analysis, machine learning, signature-based detection, and WAFs, web application developers and security practitioners can strengthen their defenses against XSS attacks.

In this paper, we determine whether the input string is an XSS attack or not by using a group of special characters. For this purpose, we need to model an information object, i.e., a query consisting of a sequence of special characters from Table 1 and keywords from Table 2.

The following characters and keywords often used to construct XSS attacks:

Table 1. Special characters for defining XSS attacks

Variable	Special symbols	Importance ratio	Variable	Special symbols	Importance ratio
A_1	“	0,561	A_{17}	}	0,043
A_2	>	0,331	A_{18}	#	0,552
A_3	/	0,511	A_{19}	+	0,033
A_4	<	0,331	A_{20}	!	0,047
A_5	Spacebar	0,485	A_{21}	,	0,067
A_6	=	0,609	A_{22}	@	0,017
A_7	‘	0,318	A_{23}	?	0,047
A_8	:	0,174	A_{24}	[	0,013
A_9	.	0,997	A_{25}	]	0,013
A_{10}	(	0,532	A_{26}	-	0,144
A_{11}	)	0,532	A_{27}	~	0,003
A_{12}	-	0,144	A_{28}	*	0,023
A_{13}	;	0,622	A_{29}	|	0,003
A_{14}	(yen sign)	0,023	A_{30}	^	0,003
A_{15}	&	0,622	A_{31}	%	0,037
A_{16}	{	0,043	A_{32}	$	0,003

Table 2. Special keywords for defining XSS attacks

Variable	Key -words	Importance ratio	Variable	Key-words	Importance ratio
A_{33}	FScommand	0,003	A_{56}	Jscript	0,171
A_{34}	<script>	0,074	A_{57}	Wscript	0,012
A_{35}	</script>	0,164	A_{58}	Vbscript	0,003
A_{36}	on\w*	0,003	A_{59}	Vbs	0,003
A_{37}	style	0,097	A_{60}	Ilayer	0
A_{38}	xmlns:xdp	0,003	A_{61}	Iframe	0,015
A_{39}	Formaction	0,010	A_{62}	Applescript	0
A_{40}	Form	0,020	A_{63}	Jar	0
A_{41}	xlink:href	0,003	A_{64}	Eval	0,006
A_{42}	seekSegment Time	0,003	A_{65}	Document	0,064
A_{43}	FSCommand	0,003	A_{66}	base64	0,018
A_{44}	Applet	0,003	A_{67}	</script>	0,164
A_{45}	Audio	0,003	A_{68}	<script	0,161
A_{46}	Basefont	0,030	A_{69}	Keygen	0
A_{47}	Base	0,023	A_{70}	1Object	0,012
A_{48}	Behavior	0,003	A_{71}	Plaintext	0
A_{49}	Bgsound	0,003	A_{72}	Mochae	0
A_{50}	Blink	0	A_{73}	Style	0,097
A_{51}	view-source	0	A_{74}	Javascript	0,171
A_{52}	Embed	0,020	A_{75}	Xml	0,047
A_{53}	Livescript	0	A_{76}	Math	0
A_{54}	Mocha	0,003	A_{77}	Source	0,006
A_{55}	Bechavior	0,003	A_{78}	Svg	0,017

Suppose some input string is observed L and let $x_1, x_2, ..., x_{32}$ the frequency of occurrence in L

pecial characters from Table 1 and let $x_{33}, x_{34},..., x_{78}$ the frequency of occurrence of special keywords from Table 2, x_{79} frequency of occurrence of all other signs and numbers $0,1,2,...,9$ in the line L. From the point of view of defining XSS attacks, common characters $a,b,..., z$ and numbers 0, 1, … , 9 do not play an important role. So in this paper we always assume that the frequency of occurrence of all these symbols and numbers in the observed string L is equal to1, i.e. $x_{79} = 1$. In this way, any string L can be defined with certain characteristics as follows: $L = (x_1, x_2,..., x_{32}, x_{33},..., x_{78}, x_{79})$, as an element of some phase space X. From the definition L it is seen that any element L from the constructed space X lies on the hyperplane $G = \{L = (x_1, x_2,...., x_{32}, x_{33}..., x_{78}, x_{79}) : x_{79} = 1\}$.

Using this hyperplane equation, we can assume that the greater the frequency of occurrence of special characters and keywords in the input string, the more obvious is the proximity of the input string L to XSS attacks. Therefore, it is natural to assume that the attack definition function should be increasing in the variables $x_1, x_2,..., x_{32}, x_{33},..., x_{78}$, and decreasing in the variable x_{79}. Based on these considerations, the following function is proposed to define XSS attacks, which is an increasing function on the variables $x_1, x_2,..., x_{32}, x_{33},..., x_{78}$ and decreasing in x_{79}:

$$f(L) = f(x_1, x_2,..., x_{32}, x_{33},..., x_{78}, x_{79}) = \frac{\sum_{i=1}^{78} x_i}{\sum_{i=1}^{78} x_i + x_{79}}$$

As in this paper we always assume that the frequency of occurrence of all other characters and numbers 0,1, 2,..., 9 in the line L is equal to 1, then from the last equality we obtain

$$f(L) = f(x_1, x_2,..., x_{32}, x_{33},..., x_{78}, x_{79}) = \frac{\sum_{i=1}^{78} x_i}{\sum_{i=1}^{78} x_i + 1} \quad (1)$$

If the input string is L is an XSS attack, then this string must at least contain one special character from Table 1 or one keyword from Table 2. Therefore $\sum_{i=1}^{78} x_i \geq 1$ and because the function $f(L)$ is increasing for each of the variables x_i its minimum at $\sum_{i=1}^{78} x_i \geq 1$ is reached at the point L_0 for which $\sum_{i=1}^{78} x_i = 1$. Thus, if L random string and $f(L) \geq 1/2$, then L is probably an XSS attack, in which case it is built using a minimum of either 1 a special character from Table 1 and 1 keyword from Table 2 are used to build it, or 1 keyword from Table 2. If, on the other hand $f(L) < 1/2$ then the input string is probably normal. Therefore, function (1) can be used to recognize XSS attacks and normal strings constructed with special characters and keywords.

Further, let us refine the algorithm by using the importance coefficients of special characters in the construction of the recognition function. To do this, let us construct the function

$$f_1(L) = f_1(x_1, x_2,..., x_{32}, x_{33},..., x_{78}, x_{79}) = \frac{\sum_{i=1}^{78} k_i x_i}{\sum_{i=1}^{78} k_i x_i + 1} \quad (2)$$

where $k_i,\ 0 < k_i < 1,$ – sign importance coefficients from Table 1. The importance coefficients are calculated by examining 299 pieces of real XSS attacks. Using the values of importance coefficients and the form of function (2), it is easy to determine the minimum of the new function $f_1(L)$ under the condition $\sum_{i=1}^{78} x_i \geq 0{,}003$. The minimum of this function is reached at the point L_0, for the coordinates of which the equality $\sum_{i=1}^{78} x_i = 0{,}003$. Thus, from the new function $f_1(L)$ we have that if L arbitrary string and $f_1(L) \geq 0{,}003$, then L is probably an XSS attack and in this case at least 1 special character from Table 1 and 1 keyword from Table 2 are used to build it, or 1 keyword from Table 2. If, however $f(L) < 0{,}003$ then the input string is normal. Therefore, function (2) can be used to recognize XSS attacks and normal strings constructed with special characters and keywords.

One of the advantages of a mathematical algorithm is its adaptability. It can be trained and tuned using both past and real-time data, allowing it to constantly evolve and improve its detection capabilities. To ensure detection accuracy and minimize the number of false positives and false negatives, it is crucial to have a complete dataset that includes different vectors and scenarios of XSS attacks.

Applying the proposed method does not require complex mathematical calculations. Using statistical processing of real incoming data, the importance coefficients of all special characters and keywords involved in the construction of XSS attacks were calculated. Applying the obtained recognition algorithm for the model that takes into account the importance coefficients of special characters, a new algorithm for recognizing XSS attacks was obtained. Thus this method can be used as an online tool to detect XSS attacks.

3 Conclusion

The mathematical algorithm for detecting XSS attacks on web applications, taking into account the frequency of occurrence and importance coefficient of characters involved in the construction of incoming requests presented in this research leverages various mathematical techniques and data analysis methodologies to identify potential XSS vulnerabilities. In this case, from Tables 1 and Table 2 we can see that the importance coefficients for special characters (table 1) have a significant difference, and the importance coefficients for keywords are very close to zero. Therefore, if we do not take into account some parameters with zero coefficients when constructing functions (1), (2), the recognition of incoming queries is possible with some small error.

In conclusion, a mathematical algorithm for detecting XSS attacks on web applications is a promising approach to improve the security of web applications. Using mathematical principles and data analysis techniques, it provides an adaptable and systematic method for identifying potential XSS vulnerabilities. However, it must be used as part of a comprehensive security framework to provide a robust defense against XSS attacks.

Our further research is devoted to the development of a software tool based on the proposed algorithm and its evaluation.

References:

[1] Banerjee R., Baksi A., Singh N., Bishnu S.K., Detection of XSS in web applications using Machine Learning Classifiers, *International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech) 2020 4th*, Kolkata, 2020, pp. 1-5, doi: 10.1109.

[2] Monali Shetty, Chirantar Nalawade, Hybrid approach for Detection and Analysis of SQL and XSS vulnerabilities, *International Journal of Engineering Trends and Technology*, Vol.59,, 2018, pp. 37-41, doi: 10.14445/22315381/IJETT-V59P206.

[3] Wassermann G., Su Z., Static detection of cross-site scripting vulnerabilities, *2008 ACM/IEEE 30th International Conference on Software Engineering*, Leipzig, Germany, 2008, pp. 171-180, doi: 10.1145/1368088.1368112.

[4] Fawaz Mokbal, Dan Wang, Xiaoxi Wang, (2022). Detect Cross-Site Scripting Attacks Using Average Word Embedding and Support Vector Machine, *International Journal of Network Security,* Vol.24, No.20-28, doi: 10.6633/IJNS.202201.

[5] Sanjukta Mohanty, Arup Abhinna Acharya, Detection of XSS Vulnerabilities of Web Application Using Security Testing Approaches, *Intelligent and Cloud Computing, 2021,* pp.267-275, doi: 10.1007/978-981-15-6202-0_27.

[6] Jasleen Kaur, Urvashi Garg, Gourav Bathla, Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review. *Artificial Intelligence Review,* Vol.56., 2023, pp. 1-45, doi: 10.1007/s10462-023-10433-3.

[7] Garn B., Sebastian Lang D., Leithner M., Richard Kuhn D., Kacker R., Simos D.E., Combinatorially XSSing Web Application Firewalls, *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW),* Porto de Galinhas, Brazil, 2021, pp. 85-94, doi: 10.1109/ICSTW52544.2021.00026.

[8] Domarev V.V., *Security of information technologies. Methodology of creation of protection systems*, DiaSoft, 2002.

[9] Sonoda M., Matsuda T., Koizumi D., Hirasawa S., On Automatic Detection of SQL Injection Attacks by the Feature Extraction of the Single Character, *Proceeding in 2011 International Conference on Security of Information and Networks*, ACM, pp.81-86, 2011.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)
The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself
No funding was received for conducting this study.

Conflict of Interest
The authors have no conflicts of interest to declare.