www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 8 Aug 2015, Page No. 13777-13780

Prevention Of Cross Site Scripting Attack Using Filters By String Based Approach

S. Karthika Devi, Dr.E. Ramaraj[2],

[1]M.phil Research Scholar, Department of Computer Science &Engineering,
Alagappa University, Karaikudi
Skarthikadevi1992@gail.com

[2]Professor, Department of Computer Science &Engineering,
Alagappa University, Karaikudi
eramaraj@rediffmail.com

Abstract: ***Among the Web application vulnerabilities Cross Site Scripting attack is most common attack. It is a kind of attack in which the intruder can able to change the entire code of the process by hooking unnecessary data along with the code of data. It becomes a challenging issue to sanitize every user query form through which the malicious code would be hooked. In this paper a method is proposed, by which the Cross site scripting attack on web applications will be considerably reduced. The proposed method provides single solution to various kinds of attacks that is created by the attackers. The main objective is to prevent the attack, by incorporating the data dictionary along with the client side scripting rather than separate arrangement. Our approach is examined with real web application and results are evaluated. From the experimental results it is analyzed that by using the method, it does not need a very long rule generation or separate data dictionary. This method reduces time complexity, without random generation of input values. The implementation shows that the proposed method works well for the real time cross site scripting attacks.***

Keywords: Cross Site Scripting attacks, Web Applications, Query Generation, Rules Generation, Data Dictionary, Input Sanitation

1. Introduction

Cross-site scripting attacks create a serious threat to the security of real time web applications [1]. Information security is the practice of securing information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. Information security plays an important role in protecting the data and assets of an organization [2]. Defacement of websites, server hacking and data leakage are the main areas in the information security. Organizations need to be fully aware of the need to allocate more resources to the protection of information assets, and information security must become a top concern in both government and business [3]. The wide range of new services due to the advances in computer networking changed the day-to-day activities of human life. While these technologies have brought in phenomenal convenience to the end-users, deficiencies in the underlying technologies have had negative impact on these services [4]. The rapid growth of Internet resulted in feature rich, dynamic web applications. This increase resulted in the harmful impact of security flaws in such applications. Vulnerabilities leading to compromise of sensitive information are being pare ported continuously, resulting in ever increasing financial damages. A web application is an application that is accessed via a web browser over a network such as the Internet or an Intranet.

Web-application attacks are considered to be one of the major security concerns of a large number of applications, especially those deployed in health care, banking and e-business operations [5]. Most Common way to detect these vulnerabilities is to use fully automated tools such as Web Vulnerability Scanners. Cross-Site Scripting Flaws occur when accessing information in intermediate trusted sites. Our proposed method with reduced data dictionary indeed append with the input attributes defend against various scripting attacks. The paper is structures as follows. An introduction about cross site scripts is given in section 1. Various reviews and literature on scripting attacks are provided in section 2. Section 3 consists of the proposed method to eliminate the scripting attack. Section 4 reveals the experimental results from the implemented web applications. Section 5 shows the conclusion and future scope.

2. Related Works

An accepted web application architecture that automatically apply nonce spaces to the static content on the web pages is proposed by the authors on [1]. A policy language was designed by them, specifically to the nonce spaces, tested under various applications to ensure the effectiveness of the method. With the aid of this method the server can reliably identify and annotate untrusted content, the client can enforce flexible policies that prevent XSS attacks while safely allowing rich user-contributed content [6].
Navdeep Kaur and Parminder Kaur [7], gives a detailed study on input validation vulnerabilities in different web applications that are in real time environment. The authors provide a complete study on cross site scripting attacks in input validation with type of methods to eliminate the situation. According to their study, the problem can be disinfectant during the life cycle phase itself. Hence, vulnerability can be

eliminated at larger cost.

Isatou Hydara et al [8] make a detailed study on state of art in cross site scripts with a systematic literature review. Different types of applications are intended for the study by the authors that concentrates on how and where the scripting vulnerabilities or attacks taken place. From their review results dynamic query generation avoids the scripting attacks more rather than static dimensions. Scripting attacks cannot be mitigate with the aid of single solution. The absence of vulnerabilities will prevent attacks from occurring and save resources.

Steven Van Acker et.al [9], discusses on automatic discovery of scripting in web applications with a system names" Flashover". It is a combined structure of both static and dynamic code analysis with no false positives in the query. The results shows that a significant number of high valued web applications are vulnerable to cross site scripting attacks, can be eliminated by using the flash over methodology.

Preventing the websites from scripting attacks by using the E-Guard Algorithm is specified in [10]. A passive detection system to identify successful XSS attacks. Based on a prototypical implementation, the accuracy and its detection capabilities are verified by the authors. A data-set of web request and response from 20 popular web applications for this, in combination with both real word and manually crafted XSS exploits are done. The detection approach results in a total of zero false negatives for all tests, while maintaining an excellent false positive rate for more than 80 percent of the examined web application.

Improved Cross site Scripting Filter for the input validation was provided by Elangovan Uma and Arputharaj Kannan in [11]. A self aware message analysis and validation algorithm was given for detecting and providing from various scripting attacks in the real web applications. It receives request and supply responses together with specific query for input validation. A filtering policy is applied for the purpose of validating the user input query.

3. Proposed Method

3.1 Cross Site Scripting Attacks

Cross Site scripting also known as the code injection attacks, destroys the input data validation. The intruded data is combined with the code on the web applications, run during the execution time. In scripting attacks, intruders add malicious code in the user validation input phase. Most commonly JavaScript is taken into account. In the process, the context of the vulnerable website can access, among others, the session cookies of that website and transfer them to an attacker-controlled server. Hence, the whole session would be controlled by the intruders. In web based services, the cross site scripting is most frequently done attacks; the malicious code is embedded along with the input query. Detection tools are necessary to detect the occurrence of the malicious code. Almost available data dictionaries act as the antivirus processing tools. This proposed method describes the new method for detecting the occurrence of the malicious code or the script in the client side itself. Cross-Site Scripting (XSS) is a widespread security issue in many modern Web applications. One way to detect the client side vulnerabilities is to apply fully automated tools such as Web Vulnerability Scanners. XSS is a new common vulnerability which can let hackers inject the code into the output application of web page which will be sent to a visitor's web browser and then, the code which was injected will execute automatically or steal the sensitive information from the visits input.

3.2 The Proposed Scheme

A new method for the detection and prevention of the malicious code in the scripting site is provided. The method is the combined architecture with no rules generation to detect various scripts in the database. The proposed method works as follows: Cross site scripting attacks usually taken place at the user validation or in the user input query execution. In the proposed method the input query is clearly investigated, whether it consists of any malicious code. The following steps explains the working procedure of the method,

Step 1: Input the user Query in the available validation control.

Step 2: Identify for any script within the input entered by the user.

Step 3: Split the input values into two characters for each tag. (Eg. <un><am><ee>)

Step 4: Check the occurrence of the scripts such as </Script>…

Step 5: Check the given input with the predefined tags specified namely <? ?>, <!, > <# …> <s …> , </ ….>, <=…>, <Sc.

Step 6: If the separated tags consists of these predefined tags then it will be automatically detected and replaced with warning query and does not allowed for execution.

3.3 Advantages

The proposed method holds many advantages that includes the simple sanitation of queries, reduces the database storage as well as the rule generation. Random generation of the queries is also eliminated by this method. There is no need for separate data dictionary, since all the predefined tags are embed along with the user input validation code. Thus, unlike the existing data dictionaries with common factors for cross site script identification, the proposed method uniquely maintains predefined tags and prevents the site from attacks.

Table 1 : Comparative study shows the attacks on web application with respect to languages

Technique Used	*CSS*	*Command Injection*	*Directory Traversal*	*Platform Independence*
Ardila	Detect the Attack	Do not Detect the Attack	Do not Detect the Attack	Not Allowed
D-Wav	Detect the Attack	Do not Detect the Attack	Do not Detect the Attack	Not Allowed
Detection of Web application Vulnerability using Parameters(DWVP)	Detect the Attack	Detect the Attack	Do not Detect the Attack	Not Allowed
Proposed Method	Detect the Attack	Detect the Attack	Detect the Attack	Allowed

Table 2 : Evaluation Results of Common Search Queries and results in percentage

Variable Name	*Instances Found*	*Percentage*
Click tag	101	37.56 %
Page URL	97	38.91%
Click	26	15.01%
Count URL	10	3.78%
Download Address	2	0.50%

4. Results and Discussions

The experimental results evaluated using our proposed method, shows the effectiveness of the algorithm. It detects and replaces the injected scripts with valid queries instantly. Various web applications are used for the implementation purpose [Ref 6, 12, 13]. Comparative analysis is also done, to prove the performance of the proposed method. Table 1 shows the comparison with respect to various types of attacks, including languages that are used to develop the web applications. The languages used here are the CSS, Command Injection, Directory Traversal and platform independence.

From the table 1, given below it is clear that the proposed method is resist to various attacks and also it assures platform independence. In the existing scenario, the detection of vulnerabilities is taken place in the web application only after the attack takes place. The proposed method overcomes the limitation, by directly embedding to the code. Scripts entered into the server or from the clients are initially detected and replaced with the valid queries.

Nearly 10,000 URLs are collected from various web applications and valid files are used for the further process. Web browsers like Google, provides result to the search query "\filetype:swf inurl:clickTag". [Ref 9]. In this click tag is common vulnerability attack. The table 2 consists of common vulnerability and their instances found from the respective URL(Uniform Resource Locators). From the table it is clearly stated that nearly 70 % of the vulnerabilities are from the page and click tag URLs. The proposed system, overcome the difficulty in the URL and hence the attacks can be eliminated.

5. Conclusion

In this paper, detection the scripting attacks from web applications is done. The proposed method, with valid replacing queries, detect and replaces the vulnerable script that is embed along with the code given by the users. This method is suitable to detect the vulnerable attacks such as the sql code injection attacks, Scripting attacks etc. Execution time is reduces since, the detection and replacement process is embedded within the code. Another advantage of the proposed method is that there is no random generation of queries; hence there is no need for separate data dictionary. The represented method provides protection against Cross Site Scripting from the sensitive information such as cookies and session IDs are interrupted. Both static and dynamic vulnerabilities are detected by using the proposed method.

DOI: 10.18535/ijecs/v4i8.27

References

[1] Matthew Van Gundy and Hao Chen, "Noncespaces: Using randomization to defeat cross-site scripting attacks", in computers & security, Vol.31, 2012, pp: 612 – 628.

[2] Debasish Das,Utpal Sharma and D.K. Bhattacharyya, "Detection of Cross-Site Scripting Attack under Multiple Scenarios", in The British Computer Society 2013, pp:1-15.

[3] K. Selvamani, A.Duraisamy and A.Kannan, "Protection of Web Applications from Cross-Site Scripting Attacks in Browser Side", in International Journal of Computer Science and Information Security, Vol. 7, No. 3, March 2010, pp:229 – 236.

[4] Christopher Krugel, G.Vigna and William Roberston, "A Multi model Approach to the Detection of Web based Applications:, in Computer Networks, Vol 48, Issue 5, 2005, pp:717 – 738.

[5] O. Ismail et.al , "A Proposal and Implementation of Automatic detection and Collection system for Cross Site Scripting Vulnerability", in Proceedings of the 18 International Conference on Advanced Information Networking and Applications, Japan, Vol 1, 2004, pp:145- 151.

[6] Puspendra Kumar and R.K. Pateriya, "DWVP: Detection of Web Application Vulnerabilities using Parameters of Web Form", in Elsevier Proceedings, March 2013, pp:437 – 443.

[7] Navdeep Kaur and Parminder Kaur," Input Validation Vulnerabilities in Web Applications", in Journal of Software Engineering, Vol 8, Issue 3, 2014, pp:116 – 126.

[8] Isatou Hydara, Abu Bakar Md. Sultan, Hazura Zulzalil, and Novia Admodisastro, "Current state of research on cross-site scripting (XSS) – A systematic literature review", in Information and Software Technology, Vol. 58, 2015, pp: 170–186.

[9] Steven Van Acker, Nick Nikiforakis, Lieven Desmet, Wouter Joosen, Frank Piessens, "FlashOver: Automated Discovery of Cross-site Scripting Vulnerabilities in Rich Internet Applications", in ASIACCS '12, May 2–4, 2012, Seoul, Korea.

[10] M. James Stephen, P.V.G.D. Prasad Reddy, Ch. Demudu Naidu and Ch. Rajesh, "Prevention of Cross Site Scripting with E-Guard Algorithm", in International Journal of Computer Applications, Volume 22, No.5, May 2011, pp:30 – 34.

[11] Elangovan Uma and Arputharaj Kannan, "Improved Cross Site Scripting Filter for Input Validation against attacks in Web Services", in Kuwait Journal of Science, Vol 41, Issue 2, 2014, pp:175 – 203.

[12] Kieyzun, A., Guo, P. J., Jayaraman, K., & Ernst, M. D. (2009, May). Automatic creation of SQL injection and cross-site scripting attacks. In Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on (pp. 199-209). IEEE.

[13] Zhang, L., Gu, Q., Peng, S., Chen, X., Zhao, H., & Chen, D. (2010, August). D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms. In Software Engineering, Advances (ICSEA), 2010 Fifth International Conference on (pp. 501-507). IEEE.