ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

A Critical Review on Detecting Cross-Site Scripting Vulnerability

Urmi Chhajed[1], Ajay Kumar[2]

P.G. Student, Department of Computer Science and Engineering, JECRC University, Jaipur, India[1]

Assistant Professor, Department of Computer Science and Engineering, JECRC University, Jaipur, India[2]

Abstract: There are several work is going on in the direction of securing Cross-Site Scripting Vulnerability. The work is also going on to finding the possible threats in the direction of attack detection. Cross-Site Scripting is a type of attack where malicious script is executed for accessing the data from any unauthorized scripts or the communication node. Cross-Site Scripting (XSS) is one of the major attack types in the communication process through java server pages. Web browsers are used in the execution of commands in web pages to enable dynamic Web pages attackers to make use of this feature and to enforce the execution of malicious code in a user's Web browser. Data alteration and visualization is also a major issue now days it is also called content sniffing. The flexibility of HTML techniques also flexible for the attackers which provide easy traceability. So in our paper we study and analyses different methodologies and algorithms which are used earlier in this direction to find the new direction in the detection of Cross-Site Scripting Vulnerability.

Keywords: XSS, HTML, JSP, Attacker

I.INTRODUCTION

For setting a real time server client environment, we generally prefer TOMCAT server with JSP (Java Script) environment. The JavaScript language [1] is widely used to enhance the client-side display of web pages. It was developed by Netscape as a light-weight scripting language with object-oriented capabilities and was later standardized by ECMA [2]. Any server side script first passes the command to the server and then it will be displayed by any HTML browser on-the-fly by an embedded interpreter. In any case, JavaScript encipher roam is not bring to an end may front a show-card vector for attacks against a user's ambience. We moreover disagree this air as a purchaser environment newcomer disabuse of which a consumer duff beseech the matter or data transfer can be possible through this point.

For the get advance manifold of the interest often performs a sandboxing workings, position the java present to programs nub do some guarded applications desolate. As we know the JavaScript programs are not provide trusted communication with the limited admissions of the browser. It can be misacted by downloading the code and retransferring it, so securing in the side is the greater demand in this era. This can be confusing the users to know about the changes. Pacify notwithstanding how JavaScript interpreters had a number of flaws in the aged, in the present climate nicest web site take advantage of JavaScript functionality. The topic with the true JavaScript moor mechanisms is mosey scripts may be directed by the sand-boxing mechanisms and agree to the same-origin policy, but windless violate the security of a system. This can be achieved forthwith a operator is lured into downloading vile JavaScript code from a trusted web site. Such an exploitation technique is called a cross-site scripting (XSS) attack [3, 4]. XSS is used to allow attackers to execute script in the victim‟s browser, which can hijack user sessions, deface web sites, insert hostile content, and conduct phishing attacks. Any scripting language supported by the victim‟s browser can also be a potential target for this attack. Web based

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

applications are accessed using Web based communication protocols and use Web browsers as graphical user interface. The most dangerous threat is alteration of the data in text, pdf files and images contents which is called content sniffing attack[5][6][7]. In this type of attach the data will be received by the client but the data is not correct or updated by the attacker.

To customize back plasticity in the HTML song and to digest round-trip delays, browsers offered the choice to encompass program pandect into the HTML permit depart is present and flawless on the catch by an interpreter integrated into the browser [8]. Java Script code may not be mixed up with Java Server Pages (JSP); JSP code is executed at the server side and not at the client browser [9][10]. The Java Applets is an option purchaser combine technology cruise allows the download and conduct of Java applications to and at the client machine. The java Applets average does quite a distance right away manipulate the browser or HTML document [11].

The remaining of this paper is organized as follows. The related work in section 2. In section 3 we discuss about problem domain. In section 4 we discuss the analysis. The conclusions and future directions are given in Section 5. Finally references are given.

II.RELATED WORK

In 2007, José Fonseca et al. [12] propose a method to evaluate and benchmark automatic web vulnerability scanners using software fault injection techniques. The most common types of software faults are injected in the web application code which is then checked by the scanners. Their results are compared by analyzing coverage of vulnerability detection and false positives. Three leading commercial scanning tools are evaluated and the results show that in general the coverage is low and the percentage of false positives is very high.

In 2009, Genta Iha et al. [13] suggest preventing XSS attacks, there are several solutions based on blacklist filtering or whitelist filtering. Unfortunately, these solutions cannot solve XSS vulnerabilities completely. They propose a binding mechanism, which is comparable to the binding mechanism for SQL. They show the evaluation results of this mechanism by implementing this mechanism into the web browser (Firefox 3.0).

In 2010, Zubair M. Fadlullah et al. [14] to combat against attacks on encrypted protocols; they propose an anomaly-based detection system by using strategically distributed monitoring stubs (MSs). They have categorized various attacks against cryptographic protocols. The MSs, by sniffing the encrypted traffic, extract features for detecting these attacks and construct normal usage behavior profiles. Upon detecting suspicious activities due to the deviations from these normal profiles, the MSs notify the victim servers, which may then take necessary actions. In addition to detecting attacks, the MSs can also trace back the originating network of the attack. They call their unique approach DTRAB since it focuses on both Detection and TRAceBack in the MS level. The effectiveness of their proposed detection and traceback methods are verified through extensive simulations and Internet datasets.

In 2011, Misganaw Tadesse Gebre et al. [15] proposed a server-side ingress filter that aims to protect vulnerable browsers which may treat non-HTML files as HTML files. Their filter examines user uploaded files against a set of potentially dangerous HTML elements (a set of regular expressions). The results of their experiment show that the proposed automata-based scheme is highly efficient and more accurate than existing signature-based approach.

In 2012, Takeshi Matsudat et al. [16] proposed a new detection algorithm against cross site scripting attacks by extracting an attack feature of cross site scripting attacks considering the appearance position and frequency of symbols. Their proposed algorithm learns the attack features from given attack samples. They prepared samples for learning and testing, to

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

show the effectiveness of their proposed algorithm. As the result their proposed detection method was successfully detected attack test samples and normal test samples.

In 2012, Fokko Beekhof et al. [17] consider the problem of content identification and authentication based on digital content fingerprinting. They investigate the information theoretic performance under informed attacks. In the case of binary content fingerprinting, in a blind attack, a probe is produced at random independently from the fingerprints of the original contents. Contrarily, informed attacks assume that the attacker might have some information about the original content and is thus able to produce a counterfeit probe that is related to an authentic fingerprint corresponding to an original item, thus leading to an increased probability of false acceptance. They demonstrate the impact of the ability of an attacker to create counterfeit items whose fingerprints are related to fingerprints of authentic items, and consider the influence of the length of the fingerprint on the performance of finite length systems. Finally, the information-theoretic achieveble rate of content identification systems sustaining informed attacks is derived under asymptotic assumptions about the fingerprint length.

In 2012, Dawei Wang et al. [18] suggest Payload-based approaches are effective to known DOS attacks but are unable to be deployed on high-speed networks. To address this issue, flow-based DOS detection schemes have been proposed for high-speed networks as an effective supplement of payload-based solutions. Author suggest that the existing flow-based solutions have serious limitations in detecting unknown attacks and efficiently identifying real attack flows buried in the background traffic. In addition, existing solutions also have difficulty to adapt to attack dynamics. To address these issues, they propose a flow-based DOS detection scheme based on Artificial Immune systems. They adopt a tree structure to store flow information such that we can effectively extract useful features from flow information for better detecting DoS attacks. They employ Neighborhood Negative Selection (NNS) as the detection algorithm to detect unknown DoS attacks, and identify attack flows from massive traffic. Because the strong tolerance of NNS, the proposed solution is able to quickly adapt attack dynamics.

In 2013, Nagarjun, P.M.D. et al. [19] propose variants of RTS/CTS attacks in wireless networks. We simulate the attacks behavior in ns2 simulation environment to demonstrate the attack feasibility as well as potential negative impact of these attacks on 802.11 based networks. They have created an application that has the capability to create test bed environment for the attacks, perform RTS/CTS attacks and generate suitable graphs to analyze the attack's behavior. They also briefly discuss possible ways of detecting and mitigating such Low rate DoS attacks in wireless networks.

In 2013, Seungoh Choi et al. [20] prove that Interest flooding attack can be applied for Denial of Service (Dos) in Content Centric Network (CCN) based on the simulation results which can affect quality of service. They expect that it contributes to give a security issue about potential threats of DoS in CCN.

In 2013, Michelle E Ruse et al. [21] propose a two-phase technique to detect XSS vulnerabilities and prevent XSS attacks. In the first phase, they translate the Web application to a language for which recently developed concolic testing tools are available. Their translation also identifies input and output variables that are used to generate test cases for determining input/output dependencies in the application. Dependencies indicate vulnerabilities in the application that can be potentially exploited when the application is deployed. In the second phase, based on the input/output dependencies determined in the first phase, they automatically instrument the application code by including monitors. The monitors check exploitation of vulnerabilities at runtime. In addition to being both as efficient and effective as the available XSS attack detection techniques, their two-phase method is also capable of identifying XSS vulnerabilities that occur due to (a) conditional copy (of inputs to outputs) and (b) construction of malicious string inputs from the concatenation of singularly benign inputs.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

III.PROBLEM DOMAIN

Table 1 Page load times (ms) with and without SWAP deployment [25]

Size(kB)	w/o SWAP	w/ SWAP	SWAP	Factor
1	27.31	196.11	168.80	7.18
10	53.84	200.50	146.66	3.72
50	120.50	331.80	211.30	2.75
100	166.23	427.66	261.43	2.57

In [25] authors presented SWAP, a server-side solution for protecting users of a Web application from cross-site scripting attacks. Due to previous, automated modifications to the Web application, this component is able to distinguish between benign, and malicious scripts. The proxy prevents each malicious response from being delivered to the client, and thus effectively inhibits the attack to be carried out on the client's browser. They have implemented a prototype, and conducted experiments, showing the efficacy of SWAP to successfully detect and defeat cross-site scripting attacks.

In [26] authors presented a new approach to software development which can pose many security challenges that bypass the domain of crosssite referencing and issues in data integrity, user authentication, and data confidentiality emerge. They presents a security framework using well-known cryptographic techniques that can be used in Server-Side mashup model and will provide solutions to most common mashup security attacks such as CSRF, XSS and other relevant security issues. They suggest certain other parameters such as access control and non-repudiation which are also essential for a system's security.

Table 2: Overhead Summary [27]

Program name	Policy Checked	Delay w.o. filter (ms)	Delay with filter (ms)	Increment (%)
JSPBlog	4	2829	2967	4.87
EasyJSP	9	3022	3187	5.45
MeshCMS	20	3091	3291	6.41
EmployDirectory	2	2937	3005	2.31

In [27] authors proposes a server side XSS attack detection approach based on boundary injection that specify expected features of dynamic content generation location. The expected benign features identified from the server side code are checked during response page generation to detect XSS attacks. They apply the proposed approach for JSP programs and evaluate with four real world JSP programs. The results indicate that the approach suffers from zero false negative. They suggest future work includes identifying ways of reducing policy checks without affecting attack detection capability. They also plan to apply the concept of boundary injection and policy generation for mitigating other vulnerabilities such as Cross Site Request Forgery.

In [28] authors have shown that their technique is efficient and can easily identify conditional copy vulnerabilities.
They suggest to extend this technique to detect and prevent different types of injection attacks in Web applications written in other languages. The overall objective is to develop a generic framework that allows grammar-based automatic translation of Web applications written in any language into intermediate test-language, as well as automatic instrumentation of Web applications based on the results of testing the test-language.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

In [29] authors introduced a method for preventing abuse of stolen cookies. It uses one-time password and challenge-response authentication to judge whether an accessing user is valid or not. With keeping usability, it can prevent the abuse of stolen cookies after their expiry and offers anti-replay attack property. They plan to implement their proposed method and evaluate its feasibility including latency.

IV.ANALYSIS

After discussing several research works we can come with some problem area in the traditional approaches which are following:

1) Security mechanism will be applied when the data will be in the communication environment [22].
2) To identify the attack by position of attack and their alteration type.
3) Clustering and partitioning techniques can be used [24].
4) Data Reformation can be applied if the attack is detected.
5) Different types of file formats are used for preventing the attacks [23].

V.CONCLUSION AND FUTURE DIRECTION

The previously described technique allows detecting and preventing different attacks in the web environment with XSS attacks. But an attacker finds a gap to bypass the protection mechanism adopted by us. So our paper provides the possible insights in these directions. In this paper we want to discover the possible attack scenario and possible remedies. We also discuss the present's techniques. In future we can design a framework based on better cryptography techniques with classifier and cluster enable mechanism accepting different file formats.

REFERENCES

[1] D. Flanagan. JavaScript: The Definitive Guide. December 2001. 4th Edition.
[2] ECMA-262, ECMAScript language specification, 1999.
[3] David Endler. The Evolution of Cross Site Scripting Attacks. Technical report, iDEFENSE Labs, 2002.
[4] CERT. Advisory CA-2000-02: malicious HTML tags embedded in client web requests.
[5] Syed Imran Ahmed Qadri, Prof. Kiran Pandey, "Tag Based Client Side Detection of Content Sniffing Attacks with File Encryption and File Splitter Technique", International Journal of Advanced Computer Research (IJACR), Volume-2, Number-3, Issue-5, September-2012.
[6] Animesh Dubey, Ravindra Gupta, Gajendra Singh Chandel," An Efficient Partition Technique to reduce the Attack Detection Time with Web based Text and PDF files", International Journal of Advanced Computer Research (IJACR),Volume-3 Number-1 Issue-9 March-2013.
[7] Anton Barua, Hossain Shahriar, and Mohammad Zulkernine, "Server Side Detection of Content Sniffing Attacks", 2011 22nd IEEE International Symposium on Software Reliability Engineering.
[8] Richard Sharp and David Scott," Abstracting Application Level Web Security," In Proceedings of the 11th ACM International World Wide Web Conference (WWW 2002), May 7-11, 2002.
[9] Peter wurzinger, Christian Platzer, Christian Ludl, and Christopher Kruegel,"SWAP:Mitigating XSS Attacks using a Reverse Proxy," In proceedings of the 2009 ICSE Workshop on Software Engineering for secure systems,pp.33-39,2009.
[10] Engin Kirda, Nenad Jovanovic, Christopher Kruegel and Giovanni Vigna,"Client-Side Cross-Site Scripting Protection," ScienceDirect Trans.computer and security ,pp.184-197,2009.
[11] Nao Ikemiya and Noriko Hanakawa, "A New Web Browser Including A Transferable Function to Ajax Codes", In Proceedings of 21st IEEE/ACM International Conference on Automated Software Engineering (ASE '06), Tokyo, Japan, pp. 351-352, September 2006.
[12] Fonseca, J.; Vieira, M.; Madeira, H., "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," Dependable Computing, 2007. PRDC 2007. 13th Pacific Rim International Symposium on , vol., no., pp.365,372, 17-19 Dec. 2007.
[13] Iha, G.; Doi, H., "An Implementation of the Binding Mechanism in the Web Browser for Preventing XSS Attacks: Introducing the Bind-Value Headers," Availability, Reliability and Security, 2009. ARES '09. International Conference on , vol., no., pp.966,971, 16-19 March 2009.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

[14] Zubair M. Fadlullah, Tarik Taleb,Athanasios V. Vasilakos, Mohsen Guizani and Nei Kato, "DTRAB: Combating Against Attacks on Encrypted Protocols Through Traffic-Feature Analysis", IEEE/ACM Transactions On Networking, Vol. 18, No. 4, August 2010.

[15] Misganaw Tadesse Gebre, Kyung-Suk Lhee and ManPyo Hong, "A Robust Defense against Content Sniffing XSS Attacks", IEEE 2010.

[16] Matsuda, T.; Koizumi, D.; Sonoda, M., "Cross site scripting attacks detection algorithm based on the appearance position of characters," Communications, Computers and Applications (MIC-CCA), 2012 Mosharaka International Conference on , vol., no., pp.65,70, 12-14 Oct. 2012.

[17] Fokko Beekhof, Sviatoslav Voloshynovskiy ,Farzad Farhadzadeh," Content Authentication and Identification under Informed Attacks", IEEE 2012.

[18] Dawei Wang; Longtao He; Yibo Xue; Yingfei Dong, "Exploiting Artificial Immune systems to detect unknown DoS attacks in real-time," Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on , vol.02, no., pp.646,650, Oct. 30 2012-Nov. 1 2012.

[19] Nagarjun, P.M.D.; Kumar, V.A.; Kumar, C.A.; Ravi, A., "Simulation and analysis of RTS/CTS DoS attack variants in 802.11 networks," Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on , vol., no., pp.258,263, 21-22 Feb. 2013

[20] Seungoh Choi, Kwangsoo Kim, Seongmin Kim, and Byeong-hee Roh," Threat of DoS by Interest Flooding Attack in Content-Centric Networking" IEEE 2013.

[21] Ruse, M.E.; Basu, S., "Detecting Cross-Site Scripting Vulnerability Using Concolic Testing," Information Technology: New Generations (ITNG), 2013 Tenth International Conference on , vol., no., pp.633,638, 15-17 April 2013.

[22] Ashutosh Kumar Dubey,Animesh Kumar Dubey, Mayank Namdev, Shiv Shakti Shrivastava,"Cloud-User Security Based on RSA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment", CONSEG 2012.

[23] Bhupendra Singh Thakur, Sapna Chaudhary,‖ Content Sniffing Attack Detection in Client and Server Side: A Survey‖, International Journal of Advanced Computer Research (IJACR), Volume-3 Number-2 Issue-10 June-2013.

[24] Ashutosh Kumar Dubey, Animesh Kumar Dubey, Vipul Agarwal, Yogeshver Khandagre, "Knowledge Discovery with a Subset-Superset Approach for Mining Heterogeneous Data with Dynamic Support",Conseg-2012.

[25] Wurzinger, P.; Platzer, C.; Ludl, C.; Kirda, E.; Kruegel, C., "SWAP: Mitigating XSS attacks using a reverse proxy," Software Engineering for Secure Systems, 2009. SESS '09. ICSE Workshop on , vol., no., pp.33,39, 19-19 May 2009.

[26] Ali, S.; Khusro, S.; Rauf, A., "A cryptography-based approach to web mashup security," Computer Networks and Information Technology (ICCNIT), 2011 International Conference on , vol., no., pp.53,57, 11-13 July 2011.

[27] Shahriar, H.; Zulkernine, M., "S2XS2: A Server Side Approach to Automatically Detect XSS Attacks," Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on , vol., no., pp.7,14, 12-14 Dec. 2011.

[28] Ruse, M.E.; Basu, S., "Detecting Cross-Site Scripting Vulnerability Using Concolic Testing," Information Technology: New Generations (ITNG), 2013 Tenth International Conference on , vol., no., pp.633,638, 15-17 April 2013.

[29] Takahashi, H.; Yasunaga, K.; Mambo, M.; Kwangjo Kim; Heung Youl Youm, "Preventing Abuse of Cookies Stolen by XSS," Information Security (Asia JCIS), 2013 Eighth Asia Joint Conference on , vol., no., pp.85,89, 25-26 July 2013.