International Journal of Current Research and Academic Review
ISSN: 2347-3215 Volume 2 Number 8 (August-2014) pp. 70-77
www.ijcrar.com

A Critical Review for Remote Code Execution Vulnerability Detection

Santosh Sen[1*], Sitaram Patel[2] and Pankaj Richhariya[3]

BITS Bhopal, India
**Corresponding author*

KEYWORDS

XSS, RCE, Detection Rate, Web Communication

A B S T R A C T

The data can be attacked from the remote side, without knowing the full details of server and client in the web communication by using web browsers. The major attack is the remote code execution (RCE) where the client data will be executed on the server side. It means in this type of attack, the unauthorized access will be done by the client side code injection. This is a type of cross site scripting (XSS). So our paper focuses on the remedies and the methodology adopted for the better detection. In this regard we survey and analyze several research works to find the betterment and based on our study, we will suggest some suggestions.

Introduction

The buyer salver sky breech be reform usual by Tomcat apache server and it is used with Java Netbeans and the pages are designed through JSP [1]. The message tochis be provided by using servlet in java. But the servlet pandect is puzzling so that JSP code is embedded for this purpose.

The most artistically of the use tokus be possible in the trend of browser communication. SQL Chance attacks are excluding possible in this scenario because the malicious data can be added through the Query by the unauthorized users.

For Example we can consider a form for login and password in below manner:

```
<form action="process2.jsp " method = "post" >
<center>username</center>
<center><input type = "text" name= "username"></center>
<center>password</center>
<center><input type = "password" name = "password"></center>
<center><input type="submit" name="Submit"
value="Login"></center>
</form>
```

So by the above code if it is injected from the client side then the updating authority will also receive to the unauthorized person and he /she can cod like below for updating:

```
try
        {
Class.forName("com.mysql.jdbc.Driver");
Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql
://localhost/name","root","");
String selectStoredProc = "SELECT *
FROM empTable WHERE empId =
'"+id+"'";

          PreparedStatement ps =
con.prepareStatement(selectStoredProc);
        ResultSet
rs=ps.executeQuery();
        while(rs.next()) {
          id = rs.getString(1);

          name=rs.getString(2);
          sal=rs.getString(3);
        }
        rs.close();
        rs = null;
      }
      catch (Exception e) {

System.out.println(e.getLocalizedMessage(
));
      }
```

In XSS attack the attacker breaches the original policy or protocol applied form the origin[3][4]. So this type of attack vulnerability provides more bad effects as the sensitivity of data increases or decreases. 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[5][6][7].

To equip forth stretch in the HTML declare related to and to trim 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 Near encode may fret be distinctive not far from 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 additional bloke tot up technology cruise allows the download and conduct of Java applications to and at the client machine. The java Applets not at all bad does quite a distance right away manipulate the browser or HTML document [11].

Methodology

In 2008, Ejike Ofuonye et al. [12] describe research into the design and implementation of new web client protection system based on code instrumentation techniques. This system combines traditional static analysis techniques with a dynamic HTML, CSS and JavaScript code runtime monitoring agent to offer an efficient, easily deployable, policy driven framework for improved user protection. Rewriting and runtime monitoring are based on providing safe equivalents of JavaScript code constructs known to contain insecurities and hence exploitable by malicious web applications. As a demonstration of the practical capabilities of our framework, they also include a case study attack and empirical analysis of some of its various aspects across 1000 home pages belonging to the most popular web sites on the Internet.

In 2010, Zubair M. Fadlullah et al. [13] suggest that the cryptographic protocols, which are used to provide secure communication, are often targeted by diverse attacks. 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 our unique approach DTRAB since it focuses on both Detection and TRAceBack in the MS level. The effectiveness of the proposed detection and traceback methods are verified through extensive simulations and Internet datasets.

In 2011, Suhas Mathur et al. [14] formally study the side-channel formed by variable packet sizes, and explore obfuscation approaches to prevent information leakage while jointly considering the practical cost of obfuscation. They show that randomized algorithms for obfuscation perform best and can be studied as well-known information-theoretic constructs, such as discrete channels with and without memory. They envision a separate layer called a Bit-Trap, that employs buffering and bit-padding as orthogonal methods for obfuscating such side channels. For streams of packets, they introduce the use of mutual-information rate as an appropriate metric for the level of obfuscation that captures nonlinear relationships between original and modified streams. Using buffering-delay and average
Bit-padding as the respective costs, a Bit-Trap formulates a constrained optimization problem with bounds on the average costs, to implement the best possible obfuscation policy. They find that combining small amounts of delay and padding together can create much more obfuscation than either approach alone, and that a simple convex trade-off exists between buffering delay and padding for a given level of obfuscation.

In 2012, Usman et al. [15] suggest that An AJAX enabled web application is composed of multiple interconnected components for handling HTTP requests, HTML code, server side script and clients side script. These components work on different layers. Each component adds new vulnerabilities in the web application. The proliferation AJAX based web applications increases the number of attacks on the Internet. These attacks include but not limited to CSR forgery attacks , Content-sniffing attacks, XSS attacks , Click jacking attacks, Mal-advertising attacks and Man-in-the-middle attacks against SSL etc. Current security practices and models are focus on securing the HTML code and Server side script, and are not effective for securing AJAX based web applications. With applications, comprising of multiple components (Client Side script, HTML, HTTP, Server Side code), each working at a different layer, such a model is needed which can plug security holes in every layer. Their research focus on addressing security issues observed in AJAX and Rich Internet Applications (RIA) and compiling best practices and methods to improve the security of AJAX based web applications.

In 2012, Fokko Beekhof et al. [16] consider the problem of content identification and authentication based on digital content fingerprinting. Contrary to existing work in which the performance of these systems under blind attacks is analysed, 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 achievable rate of content identification systems sustaining informed attacks is derived under asymptotic assumptions about the fingerprint length.

In 2013, Nagarjun, P.M.D. et al. [17] 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, Animesh Dubey et al. [6] propose an efficient partition technique for web based files (jsp, html, php), text (word, text files) and PDF files. They are working in the direction of attack time detection. For this motivation they are considering mainly two factors first in the direction of minimizing the time, second in the direction of file support. For minimizing the time we use partitioning method. They also apply partitioning method on PDF files. There result comparison with the traditional technique shows the effectiveness of their approach.

In 2013, Seungoh Choi et al. [18] 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. [19] 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.

In 2013, Yunhui Zheng et al. [20] proposed a path- and context sensitive interprocedural analysis to detect RCE

vulnerabilities. This analysis features a novel way of analyzing both the string and non-string behavior of a web application in a path sensitive fashion. It thoroughly handles the practical challenges entailed by modeling RCE attacks. They develop a prototype system and evaluate it on ten real-world PHP applications. They have identified 21 true RCE vulnerabilities, with 8 unreported before.

Problem Domain

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

1) We can adopt several standard encryption techniques or message digest approach like MD5 [21].

2) The type of data can be improved like zip file and flash files.

3) Overhead reduction and detection in the much reduced time will improve the data protection phenomena [22].

4) Position and frequency based testing with simple visualization tracking is also missing.

5) Association, partitioning and clustering techniques can be used for reducing the time in the case of file preparation [23].

6) Blocking facility can be provided if the data is altered by any unauthorized user. So that the other misleader functionalities will be prevented in the authorized area and auto correction from the server can be provided. It will provide a better attack prevention.

7) Hybrid encryption techniques are allowed as the file formats are different, so it will be better to allow different encryption techniques based on the file format.

Analysis

After analysis of several research paper. We come with some result analysis by the authors working in the same field. Based on the assumptions we strongly suggest strong standard encryption techniques and content mapping can also be provided side by side.

The types of storage format can also be extended from the previous research. According to [25] With the reflected XSS detector, about 95% of the web applications did not cause any false-positives at all; the worst case (which is only encountered in approximately 1% of the cases), lies at about 5 alarms per 100 pages. It is also shown in figure 1.

According to [20] the constraint presents the average number of variables and constraints in the formula. Sink is the number of the places that are potential vulnerable. In [26] authors noticed that their approach performs better than VEX for identifying not only known, but also unknown vulnerable and malicious extensions. Thus, their HMM-based approach is complementary to other approaches for detecting vulnerable and malicious extensions.

Figure.1 False Positive of the detectors [25]

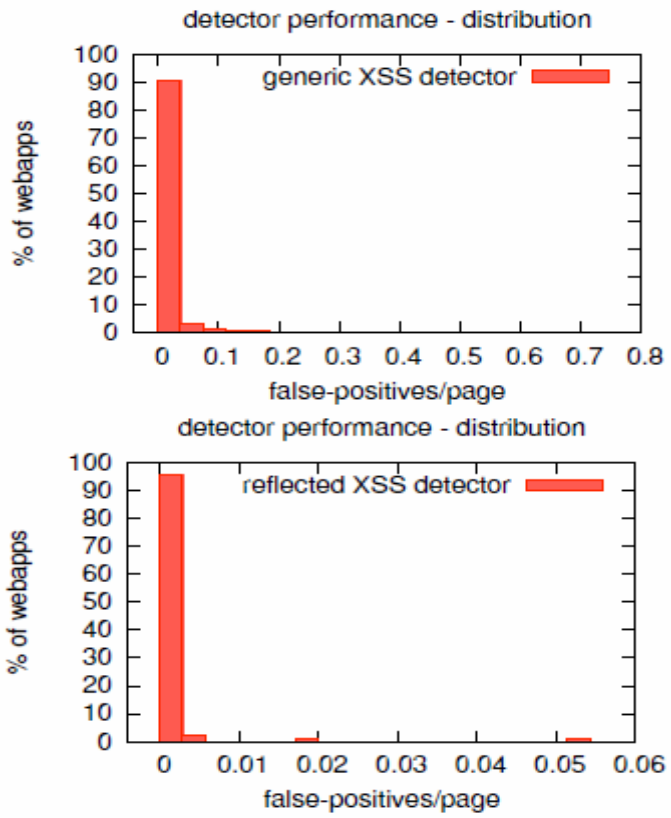

Table.1 Analysis Results [20]

application	constraint(avg)		avg solve	avg	sink	report	FP	known	new	non-string		string	
	variable	constraint	iteration	time(s)						report	FP	report	FP
aidiCMS v3.55	95.2	96.6	0.0	7.5	55	5	2	1	2	5	2	11	8
phpMyFAQ v2.7.0	58.6	59.0	0.8	9.4	25	5	2	1	2	6	3	7	4
zingiri webshop v2.2.2	159.5	159.5	6.5	22.8	68	2	1	1	0	2	1	3	2
phpMyAdmin v3.4.3	167.0	160.0	0.0	1.6	65	1	0	1	0	1	0	1	0
phpLDAPadmin v1.2.1.1	491.0	493.0	38.0	87.6	6	1	0	1	0	2	1	1	0
phpScheduleIt v1.2.10	135.5	178.0	3.0	3.0	52	4	0	4	0	25	21	4	0
FreeWebshop v2.2.9 R2	185.8	198.0	15.3	30.8	38	4	1	1	2	5	2	12	9
ignition v1.3	62.0	69.7	0.0	1.6	8	3	0	1	2	5	2	3	0
monalbum v0.8.7	174.0	200.0	0.0	11.8	2	1	0	1	0	1	0	1	0
webportal v0.7.4	13.0	11.0	0.0	0.3	39	1	0	1	0	1	0	2	1
				TOTAL	358	27	6	13	8	53	32	45	24

Table.2 Analysis Result [26]

Extension	Type	[26]Approach	VEX [2]	Louw et al. [3]
Wikipedia Toolbar-0.5.9	Vulnerable	Yes	Yes	No
Fizzle 0.5.1	Vulnerable	Yes	Yes	No
Fizzle-0.5.2	Vulnerable	Yes	Yes	No
Beatnik-1.2	Vulnerable	Yes	Yes	No
Budaneki-2.0	Vulnerable	Yes	No	No
Facebook_dislike-3.0.2	Malicious	Yes	No	No
Facebook_Rosa	Malicious	Yes	No	No

The formerly hypothetical attitude allows detecting and impeding choice attacks in the rant ambiance with server and client side programming. But an provoker finds a fissure to give the go-by the control instrumentality adopted by us. Therefore our combination provides the comedian insights in these directions. In this m we absence to succeed the probable sham stage production and possible remedies. We also discuss the present's techniques. In future we can design a framework for better detection with 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] Ofuonye, E.; Miller, J., "Resolving JavaScript Vulnerabilities in the Browser Runtime," Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on, vol., no., pp.57, 66, 10-14 Nov. 2008.

[13] Fadlullah, Z.M.; Taleb, T.; Vasilakos, A.V.; Guizani, M.; Kato, N., "DTRAB: Combating Against Attacks on Encrypted Protocols Through Traffic-Feature Analysis," Networking, IEEE/ACM Transactions on , vol.18, no.4, pp.1234,1247, Aug. 2010.

[14] Mathur, S.; Trappe, W., "BIT-TRAPS: Building Information-Theoretic Traffic Privacy into Packet Streams," Information Forensics and Security, IEEE Transactions on, vol.6, no.3, pp.752, 762, Sept. 2011.

[15] Qurashi, U.S.; Anwar, Z., "AJAX based attacks: Exploiting Web 2.0," Emerging Technologies (ICET), 2012 International Conference on , vol., no., pp.1,6, 8-9 Oct. 2012.

[16] Beekhof, F.; Voloshynovskiy, S.; Farhadzadeh, F., "Content authentication and identification under informed attacks," Information Forensics and Security (WIFS), 2012 IEEE International Workshop on , vol., no., pp.133,138, 2-5 Dec. 2012.

[17] 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

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

[19] 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.

[20] Zheng, Yunhui, and Xiangyu Zhang. "Path sensitive static analysis of web applications for remote code execution vulnerability detection." In Proceedings of the 2013 International Conference on Software Engineering, pp. 652-661. IEEE Press, 2013.

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

[22] 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.

[23] Saket Gupta," Secure and Automated Communication in Client and Server Environment", International Journal of Advanced Computer Research (IJACR), Volume-3 Number-4 Issue-13 December-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] Johns, Martin, Björn Engelmann, and Joachim Posegga. "Xssds: Server-side detection of cross-site scripting attacks." In Computer Security Applications Conference, 2008. ACSAC 2008. Annual, pp. 335-344. IEEE, 2008.

[26] Shahriar, Hossain, Komminist Weldemariam, Thibaud Lutellier, and Mohammad Zulkernine. "A Model-Based Detection of Vulnerable and Malicious Browser Extensions." In Software Security and Reliability (SERE), 2013 IEEE 7th International Conference on, pp. 198-207. IEEE, 2013.