Review Article

ACCENTS Transactions on Information Security, Vol 4(16)
ISSN (Online): 2455-7196
http://dx.doi.org/10.19101/TIS.2019.414002

A blowfish-RC6 (BRC6) with sending identification bit (SIB) mechanism for data security in XSS

Manish Agrawal[1]*, Kailash Patidar[2], Rishi Kushwah[3] and Sudesh Chouhan[3]

M.Tech Scholar, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India[1]

Professor and HOD, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India[2]

Assistant Professor, Department of Computer Science, School of Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Madhya Pradesh, India[3]

Abstract

A blowfish-RC6 (BRC6) with sending identification bit (SIB) mechanism for data security in XSS have been proposed. The data types covered in this dissertation are text, images, web files (script and web pages), word and PDF documents. In this process the data is requested from the server and the server prepares the file so that only legitimate users can access those data by applying BRC6 with sending identification bit (SIB) mechanism. The requested file along with the user details have been send. A SIB has been added to each file after preparation which is used as the identification bit if any other user tried to access it.

Keywords

JSP, HTML, CSS, Eavesdrop time, Alert Time.

1.Introduction

The security directions in different arena are increasing and the security standards have been increased in several aspects. But the vulnerabilities are also increasing with different style. In terms of web application cross-site scripting (XSS) attack is the most common attack type [1−5]. In different application JavaScript and PHP framework have been used. The client-side code has generally embedded in HTML pages. The complexity and the security increase parallel in the way that it allow the vulnerabilities also. It follows the different mechanism to adopt and prevent the vulnerabilities in different possible way.

XSS are a security issue that occurs in web applications. Different customers with different intensions can achieve SQL Injection strike in the unmistakable course in the web world [6−9]. The disobedient and most skillfully threating strike is SQL Injection alteration.

In this Modify the hawkish supporting completions the affirmation, by sincere register with segments, for the course of action for of permit in-help and to execute self-self-assured code [10]. As to four frameworks and estimation are proposed in [11, 12], yet there is need of progress in the said field.

The main objective of this paper is to apply security mechanism efficiently.

2.Literature review

In 2018, Madhusudhan and Shashidhara [15] discussed about cross channel scripting (XCS). They have suggested this as the dangerous web application vulnerability. They have suggested that it is performed through network protocols. It is the variant of XSS. They have analyzed and discussed XCS attack in detail prospective.

In 2018, Kaur et al. [16] suggested an offline and online based model for the malicious XSS attack detection on in online social network. They have tested their approach on five online social network for the XSS attack. Their result shows the little false

*Author for correspondence

positives and promising attack vulnerability detection.

In 2018, Bukhari et al. [17] discussed the malicious functions. They have suggested XSS as the client-side code injection attack. They have focused on type 1 or “nonpersistent cross-site scripting”. With non-determined cross-site scripting, malevolent code or content is inserted in a web demand, and after that in part or totally reverberated (or "reflected") by the web server without encoding or approval in the web reaction. The noxious code or content is then executed in the customer's web program which could prompt a few negative results, for example, the robbery of session information and getting to touchy information inside treats. All together for this sort of cross-site scripting to be effective, a malevolent client must force a client into clicking a connection that triggers the non-tenacious cross-site scripting attack.

In 2018, Marashdih et al. [18] discussed web applications based on data and conducting service. They have suggested the PHP for the common framework for the web applications. Now a day’s security concern is the major issue. They have suggested that XSS vulnerability is common in PHP framework. They have suggested that because of the several applications and tools the security is now increase but there are several vulnerabilities remain s unfelt. They have discussed the PHP aspects their popularity variants with the applications.

In 2018, Algaith et al. [19] discussed the use of Static Analysis Tools (SATs) for the vulnerability. They have suggested that the use of several tools may be helpful in increasing the detection capabilities. But they have suggested that it may increase the false alarms number. So they have discussed the combination of SATs for the better suitability. They have analyzed the results based on five diverse SATs to find two types of vulnerabilities these are SQL Injections (SQLi) and XS. For this they have considerd132 plugins of the WordPress content management system (CMS). Based on their approach they have suggested empirically supported guidance based on SAT tools to achieve the low false positive rates.

In 2018, Chen et al. [20] discussed about the root cause of XSS attack. As it is difficult to identify the correct JavaScript code and the JavaScript code injected by attackers by the JavaScript engine. They have discussed about the moving target defense (MTD). It is a novel technique to defeat attacks by frequently changing the system configuration. This paper portrays the structure and actualize of a XSS resistance technique dependent on MTD innovation. This strategy adds an irregular credit to each risky component in web application to recognize the javascript code in web application and the JavaScript code infused by aggressors and utilizations a security check capacity to confirm the irregular quality, if there is no arbitrary characteristic or the irregular property value is not correct in a HTML. Their results show that the method can effectively prevent XSS attacks.

In 2018, Ruohonen [21] discussed and examines software vulnerabilities in common Python packages used particularly for web development. Their dataset is basically on the base of PyPI package repository and the so-called Safety DB used to track vulnerabilities in selected packages within the repository. Their result suggest that the vulnerabilities in general is modestly severe and XSS type.

3.Methodology

The framework of this dissertation is consists of basic hypertext markup language (HTML) for web page designing and java server pages (JSP) for complex call and scripting need. It supports both server and client environment. For designing cascading style sheets (CSS) have been used. The port number of server can be specified to any available entity otherwise there is several other ports can be configured as per the need and the requirement. Apache Tomcat server version 7 is used for the server configuration. It is the most broadly utilized web server programming. The integrated development environment used here is Netbeans7.2. NetBeans IDE lets you rapidly and effortlessly create Java desktop, versatile, and web applications, and additionally HTML5 applications with HTML, JavaScript, and CSS. The IDE likewise gives an awesome arrangement of instruments for PHP and C/C++ designers. It is free and open source and has a vast group of clients and designers around the globe.

Data encryption process is applied by the Blowfish and RC6 algorithm. Blowfish has a 64-bit block size and a key length of some place from 32 bits to 448 bits. Blowfish is appropriate for application where the key does not change every now and again, similar to a correspondence interface or a programmed record encryption. The main benefit of RC6 algorithm is the block size and key length size increased by 512 bits.

The main and important part of RC6 is the key variability. *Figure 1* shows the complete process of the working mechanism.

A sending identification bit (SIB) has been added to each file after preparation which is used as the identification bit if any other user tried to access it.

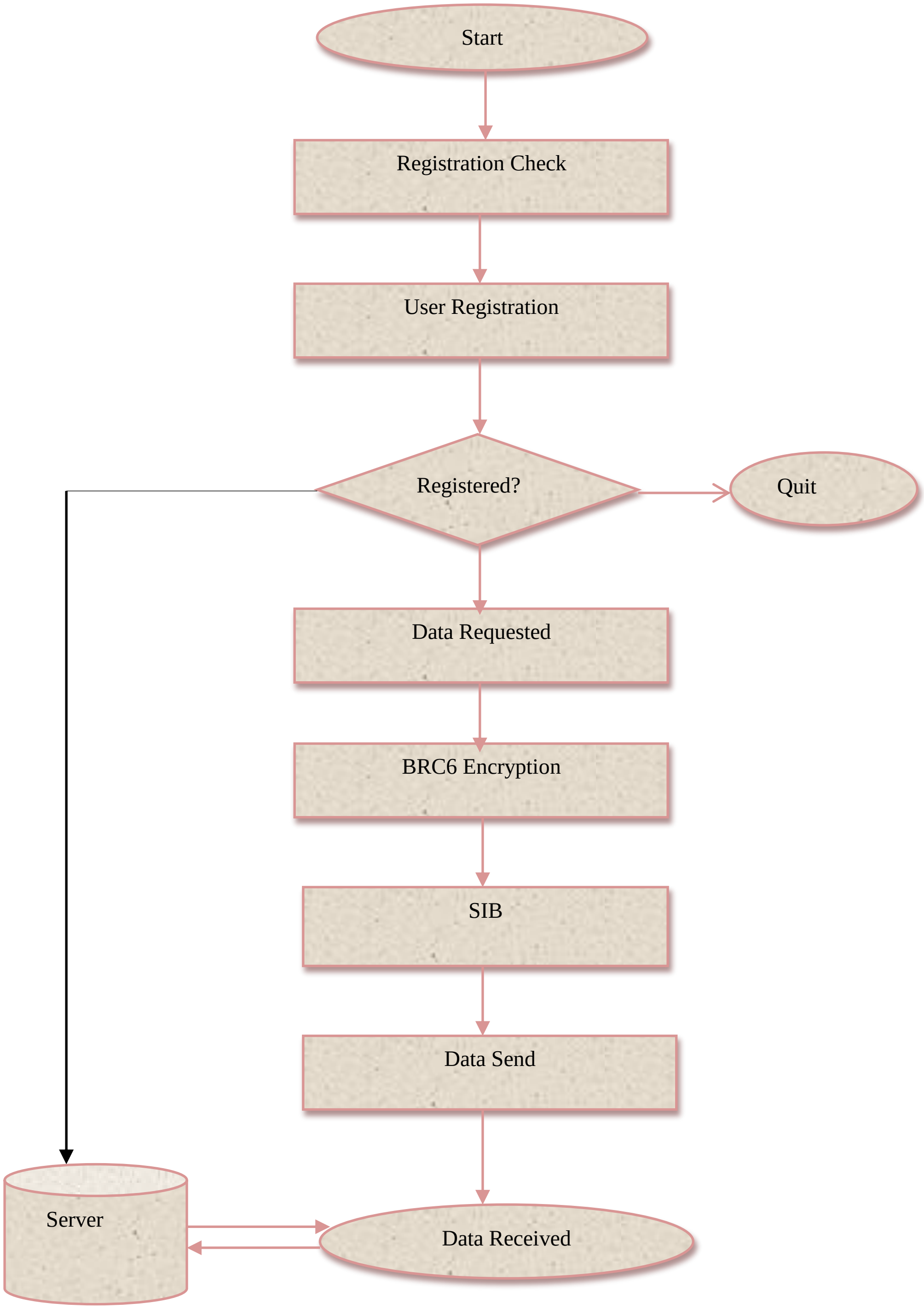

Figure 1 Flowchart

4.Results

The results obtained from our approach and the comparative study in the three different iterations cycle. *Figure 2, 3 and 4* shows the comparison from different iterations with BRC6 and with the traditional methods. The comparative study clearly shows that our approach has approaching less time in different encryption strategy along with the SIB process.

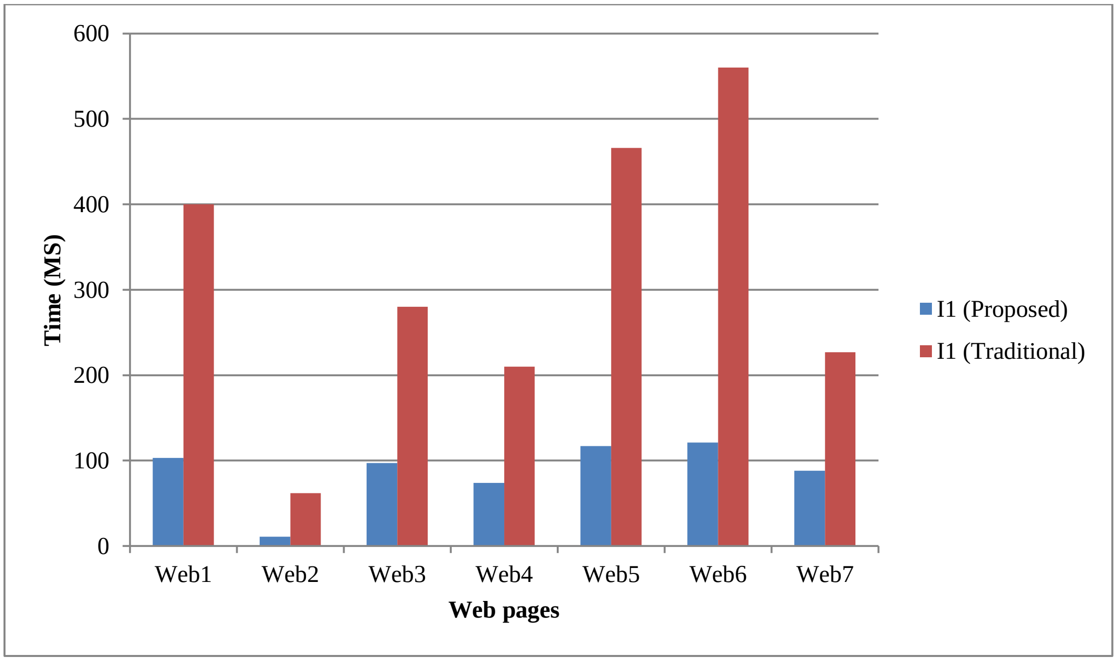

Figure 2 Results comparison from BRC6 of I1 iterations with previous method [22]

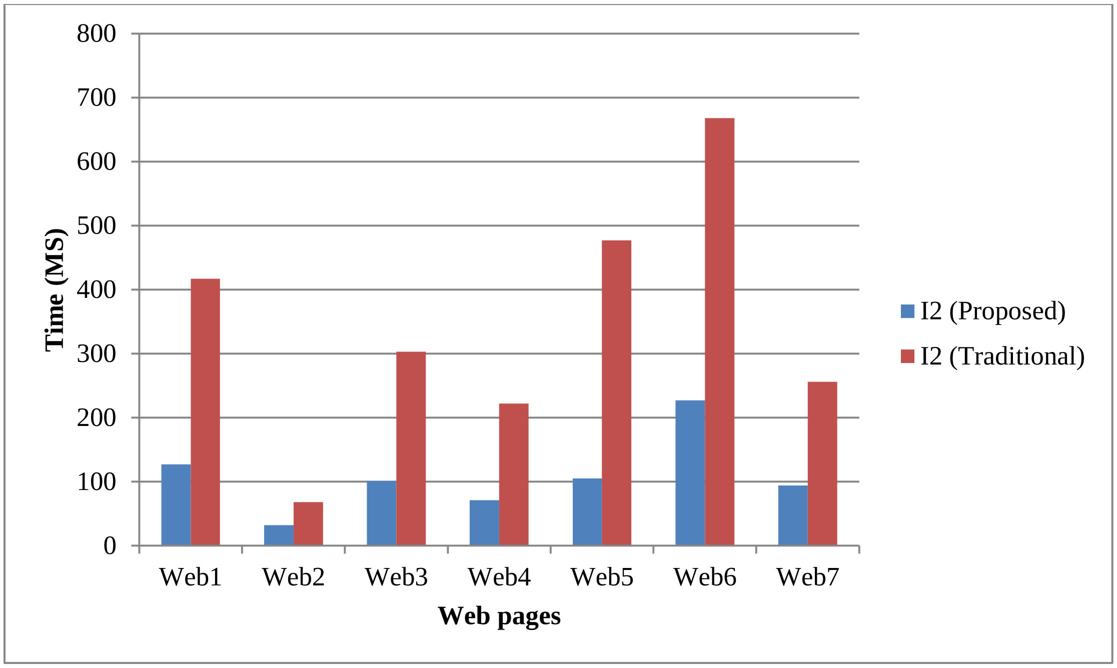

Figure 3 Results comparison from BRC6 of I2 iterations with previous method [22]

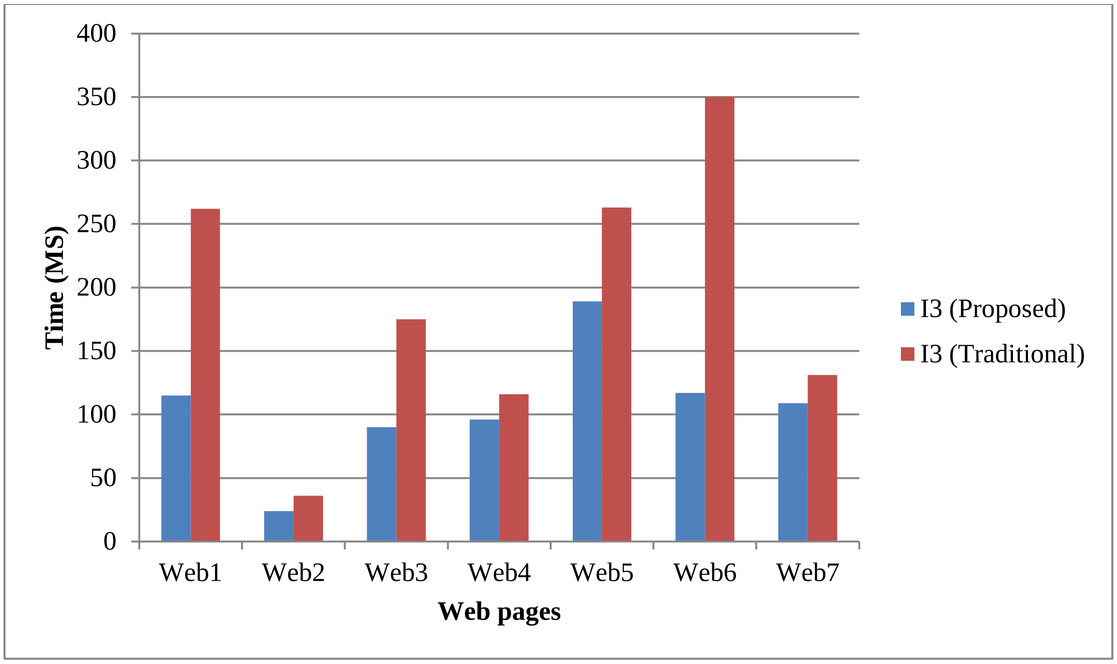

Figure 4 Results comparison from BRC6 of I3 iterations with previous method [22]

5.Conclusion

In this paper an efficient mechanism based on blowfish and RC6 (BRC6) algorithm have been proposed for better security in case of cross site scripting (XSS). First the data is requested from the server and the server prepares the file so that only legitimate users can access those data by applying BRC6 with sending identification bit (SIB) mechanism. A SIB has been added to each file after preparation which is used as the identification bit if any other user tried to access it. Prepared data is send to the client. If the authentic user access the data then there is no problem otherwise SIB bit alerts the server for the mismatch user. In our approach the attack can be prevented with the highest extend but in case of XSS attacks it can also be detected by our SIB mechanism.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Shahriar H, Zulkernine M. S2XS2: a server side approach to automatically detect XSS attacks. In international conference on dependable, autonomic and secure computing 2011 (pp. 7-14). IEEE.

[2] Conteh NY, Schmick PJ. Cybersecurity: risks, vulnerabilities and countermeasures to prevent social engineering attacks. International Journal of Advanced Computer Research. 2016; 6(23):31-8.

[3] Manimaran A, Durairaj M. The conjectural framework for detecting DDoS attack using enhanced entropy based threshold technique (EEB-TT) in cloud environment. International Journal of Advanced Computer Research. 2016; 6(27):230-7.

[4] Gupta S. Secure and automated communication in client and server environment. International Journal of Advanced Computer Research. 2013; 3(4):263-71.

[5] Shrivastava A, Choudhary S, Kumar A. XSS vulnerability assessment and prevention in web application. In international conference on next generation computing technologies 2016 (pp. 850-3). IEEE.

[6] Shar LK, Tan HB. Defending against cross-site scripting attacks. Computer. 2011; 45(3):55-62.

[7] Dubey A, Gupta R, Chandel GS. An efficient partition technique to reduce the attack detection time with web based text and PDF files. International Journal of Advanced Computer Research. 2013; 3(1):9.

[8] Kiani M, Clark A, Mohay G. Evaluation of anomaly based character distribution models in the detection of SQL injection attacks. In international conference on availability, reliability and security 2008 (pp. 47-55). IEEE.

[9] Shukla N. Data mining based result analysis of document fraud detection. International Journal of Advanced Technology and Engineering Exploration (IJATEE). 2014; 1(1):21-5.

[10] Qadri SI, Pandey K. Tag based client side detection of content sniffing attacks with file encryption and file splitter technique. International Journal of Advanced Computer Research. 2012; 2(3):215-21.

[11] Thakur BS, Chaudhary S. Content sniffing attack detection in client and server side: a survey. International Journal of Advanced Computer Research. 2013; 3(2):7-10.

[12] Valeur F, Mutz D, Vigna G. A learning-based approach to the detection of SQL attacks. In international conference on detection of intrusions and malware, and vulnerability assessment 2005 (pp. 123-140). Springer Berlin Heidelberg.

[13] Ezumalai R, Aghila G. Combinatorial approach for preventing SQL injection attacks. In international conference on advance computing 2009 (pp. 1212-7). IEEE.

[14] Junjin M. An approach for SQL injection vulnerability detection. In international conference on information technology: new generations 2009 (pp. 1411-4). IEEE.

[15] Madhusudhan R, Shashidhara. Cross channel scripting (XCS) attacks in web applications: detection and mitigation approaches. In cyber security in networking conference 2018 (pp. 1-3). IEEE.

[16] Kaur G, Pande B, Bhardwaj A, Bhagat G, Gupta S. Defense against HTML5 XSS attack vectors: a nested context-aware sanitization technique. In international conference on cloud computing, data science & engineering 2018 (pp. 442-6). IEEE.

[17] Bukhari SN, Dar MA, Iqbal U. Reducing attack surface corresponding to type 1 cross-site scripting attacks using secure development life cycle practices. In international conference on advances in electrical, electronics, information, communication and bio-informatics 2018 (pp. 1-4). IEEE.

[18] Marashdih AW, Zaaba ZF, Suwais K. Cross site scripting: investigations in PHP web application. In international conference on promising electronic technologies 2018 (pp. 25-30). IEEE.

[19] Algaith A, Nunes P, Jose F, Gashi I, Vieira M. Finding SQL injection and cross site scripting vulnerabilities with diverse static analysis tools. In European dependable computing conference 2018 (pp. 57-64). IEEE.

[20] Chen P, Yu H, Zhao M, Wang J. Research and implementation of cross-site scripting defense method based on moving target defense technology. In international conference on systems and informatics 2018 (pp. 818-22). IEEE.

[21] Ruohonen J. An empirical analysis of vulnerabilities in python packages for web applications. In international workshop on empirical software engineering in practice 2018 (pp. 25-30). IEEE.

[22] Ruse ME, Basu S. Detecting cross-site scripting vulnerability using concolic testing. In international conference on information technology: new generations 2013 (pp. 633-638). IEEE.