AN INVESTIGATION OF HEALTH SECTOR WEB APPLICATIONS IN BANGLADESH: A CASE STUDY ON CROSS SITE SCRIPTING

[1]SAMANTHA HAQUE, [2]TOUHID BHUIYAN

Daffodil International University
Email: touhidbhuiyan@gmail.com

Abstract— Extensive use of technology creates a new dimension in health sector. It contributes to endless opportunity for persuading better and more healthcare practices. Technology never been useful without ensuring the exchange of data between the systems and web application is one of the suitable media to manage that information in different stages e.g. process, store and transmission. In this present world, health sector is also started ensuring its services through online for reaching their target audiences easily. To cope up with the current requirement of business in health sector, Bangladesh also started its journey to digitalize its services by using web applications. Unfortunately, the management of health sector in Bangladesh is focusing on services in the web rather than the quality of the application. Due to lack of monitoring and control in web applications development especially, in input validation area resulting compromise of sensitive data from the medical system. Therefore, much sensitive medical information would be manipulated and redirected by exploiting the vulnerabilities like Cross-site Scripting (XSS) and Session Hijacking that are the cause of inadequate input validation. In this paper we will discuss the reason of Cross-site Scripting (XSS) and Session Hijacking vulnerability and their different exploitation types. Also we have shown the impact of those vulnerabilities in medical sector.

Keywords— Web application vulnerability, Cross Site Scripting, Medical Sector.

I. INTRODUCTION

Medical right is one of the basic needs of every human being. For ensuring it to all citizens of the countries, web application is one of the effective media to provide the health service in each door of its customer. Therefore, last few years Bangladesh Ministry of Health started to give different services to the citizen of the country and also inspired all counterparts to use the online service. By the implementation of web application, the service providers are able to ensure better medical support and awareness to its audience. On the other hand, people of Bangladesh are getting health related tips and information with in a minimal time and effort with the help of web application. Now days, most of the medical organizations like hospital, pharmaceutical company, diagnostic center etc. are using web application rapidly for providing best services of their clients. Usually the web applications are used for exchanging sensitive information about patients, doctors, medical resource and the organizations. During our study, it has been observed that the top management of these sectors is not having much knowledge on information technology especially on security issues and its impact in case of any security breaches. Thus, the web applications are currently running with different vulnerability where cross site scripting and session hijacking is the common of them. Cross site scripting occurs when developers develop a web application just complete raw coding or main features/ functionalities for any specific web application. That time they don't take any security issues as a big deal. On that case they don't use any kind of functions for validating user's inputted data. A web application is open for all users and all users are untrusted for web site and they can do any malicious in website. So users can input any data into the input fields which can be malicious. In this situation if developer not use any input validation then server take input data as trusted data and got executed on the web server. Cross site script has different impact on web applications session hijacking is one of them. Session hijacking occurs when the attacker can compromise the session token by using malicious code or programs running at the client-side. If an attacker sends a crafted link to the victim with the malicious JavaScript, when the victim clicks on the link, the JavaScript will run and complete the instructions made by the attacker. The XSS attack to show the cookie value of the current session; using the same technique it's possible to create a specific JavaScript code that will send the cookie to the attacker.In this study we are briefly discuss about different exploitation technique of cross site scripting like Persistent XSS, Non Persistent XSS, DOM Based XSS and its impact on medical sector web applications in Bangladesh.

An assessment and analysis on Cross Site Scripting (XSS) vulnerability, with its three major exploitation techniques, are discussed in this paper. We have also tested the three given techniques on the educational websites in Bangladesh.

This paper is organized in five sections. Literature Review 2.XSS and its exploitation types along with its working process are explained in section 3. After performing data analysis, result and statistics are furnished in section 4. Conclusion is provided in section 5.

II. LITERATURE REVIEW

In our study time we found several numbers of researches on SQLi, XSS, CSRF,RCE,LFI,LFD Code

Injection and Buffer Overflow[1][2][3][4][5][6][7][8][9][14][15][]. In Bangladesh region different researchers also worked on SQL injection[14][15], Cross Site Scripting, CSRF,LFD and LFI. Some researchers provide review and different analysis on SQL injection in education sectors and .bd domains in Bangladesh [1][2][3][4][5][9][10]. We also got a few number of case study on XSS and CSRF in Bangladesh, but we did not find any researches on XSS vulnerability in medical sector of Bangladesh all though XSS is a one of the most critical and OWASP ranked vulnerability[ ]. In this paper we are focusing on XSS vulnerability exploitation techniques and its impact on medical sector in Bangladesh.

III. XSS AND ITS EXPLOITATION TYPE

XSS (Cross Site Scripting) is a web vulnerability, stands for Cross Site Scripting regarded as the second biggest web loophole gaining second place after SQLi. This vulnerability is occurs when the website didn't filter some special characters in input and URL parameters fields such as ();>=<'"or not properly sanitized malicious symbols. For these type of vulnerability where an attacker use JavaScript and HTML injecting on the client side into the web page exploit this vulnerability. After that these malicious script will be executed on the vulnerable web server or any targeted remote machine. Attacker can exploit this vulnerability by submitting queries into URL, input fields, text boxes, message boxes and comments fields. Search terms or even perimeter after URL may be affected by caused of XSS.

Why Occurs XSS:

When developers develop a web application they just complete raw coding or main features/ functionalities for any specific web application. That time they don't take any security issues as a big deal. That's why they don't use any kind of functions for validating user's inputted data. A web application is open for all users and all users are untrusted for web site and they can do any malicious in website. So users can input any data into the input fields which can be malicious. In this situation if developer not use any input validation then server take input data as trusted data and got executed on the web servers.

```
Line01: <?php
Line02: $var = $_GET['firstvariable'];
//getting inputs without validations
Line03: echo $firstvariable;
Line04: $var1 = $_POST['secondvariable'];
Line04: echo $secondvarable;
Line05:echo$_SERVER['HTTP_USER_AGENT'];
//executing malicious codes
Line06: ?>
```

Code 0: Coding pattern of XSS vulnerability

In these following code at Line02 and Line04 is taking inputs from out fields and at Line05 that inputs are executed on the host system directly. So if an user manipulate the inputted data with his malicious codes these may causes harm to the server system.

A. Exploitation Techniques:

Generally, there are three types of XSS exploitation techniques. These are discussed as follows:

i) Persistent XSS
ii) Non Persistent XSS
iii) DOM Based XSS

i) Persistent XSS (Stored XSS):

Persistent XSS is also cognizant as Stored XSS. It is the most devastating variant of cross site scripting flaw. It occurs when an attacker inject a malicious script in the input field and this malicious script will saved by the server permanently.

```
Line01: <?php
Line02: if(isset($_POST['btnSign']))
Line03: {
Line04: $message=trim($_POST['mtxMessage']);
Line05: $name=trim($_POST['txtName']);
Line06:$message=stripslashes($message);
// Sanitize message input
Line07: $message = mysql_real_escape_string($message);
Line08:$name=mysql_real_escape_string($name);
// Sanitize name input
Line09: $query = "INSERT INTO guestbook (comment,name) VALUES (
'$message','$name');";
Line10:$result=mysql_query($query)or die('<pre>'.mysql_error().'</pre>');
Line11: }
Line12:?>
```

Code 01: Vulnerable Coding Pattern of Persistent XSS

Here the Code 01 describes the persistent xss vulnersbility. Line06 takes message input and Line08 takes name input from the user without any proper validation, and that malicious input is sending to the server through guestbook e.g(Line09). Line10 is executing the entered query to the specific database that creates Persistent XSS vulnerability.

Exploitation Process:

First an attacker targets a web application which is vulnerable to XSS. This website is like a forum, peoples visit this site for making New Post/New Thread/New Topic. Then attacker find that kind of fields where his given will be executed and stored on the server e.g. (Message Box, Visitor Log, Comment Field, Login Field). First he may check that his inputs will be saved or not then if he found that he may include some malicious scripts like, <script>document.body.background="http://www.mywebsite.com/imageid=16";</script>. when It will be executed and saved by the server the given image file will be displayed on the client site of the vulnerable

web application. If he wants something like cookie he may use, "><svg /onload=prompt=document.cookie);>. When the procedure perform correctly it leaked all the session cookies to the attacker with a pop-up box.

ii)Non-Persistent XSS (Reflected XSS):
It is also known as Reflected Cross Site Scripting. These vulnerability can redirect users from that site to any malicious site. When a user visits any specific web application which is vulnerable to reflected XSS he may click malicious links or posts posted by the attackers.

```
Line01: <script type="text/javascript">
Line02: function learnMoreClicked(link) {
Line03: link.style.visibility = 'hidden';
Line04:document.getElementById('explanation').styl
e.visibility = 'visible';
Line05: }
Line06: function followLinkClicked() {
Line07: location.href = "javascript:alert(1)";
Line08: }
Line09: </script>
```

Code 02: The vulnerable java script code
Code 02 is the example of Non-Persistent XSS. Only a set of java script code has been used to demonstrate the vulnerability. At Line07 location.herf is used to enter the location name with a pop-up menu and these creates vulnerability by redirecting the vulnerable web application to any specific application.

Exploitation Process:
For these kind of attack an attacker send a special kind of payloads will be sent to the server via HTTP Request and the server will execute that without any filtering. Attacker may send it through imputable parameters e.g. http://www.refvulnerablesite.com/followlink/notify?u=javascript:alert(yes) . Now an attacker may use this kind of codes to redirect victims to a malicious page.
http://www.refvulnerablesite.com/followlink/notify?u=http://www.malicioussite.com/signupFB.php

iii)DOM Based XSS (Document Based XSS):
Document Based XSS vulnerability is generally known as DOM Based XSS. This vulnerability occurs in when an application is at the stage of processing any documents by the users.

```
Line01: <script>
Line02: document.write("<OPTION
value=1>"+document.location.href.substring
Line03: (document.location.href.indexOf
("default=")+8)+"</OPTION>");
Line04:document.write("<OPTION
value=2>English</OPTION>");
Line05: </script>
```

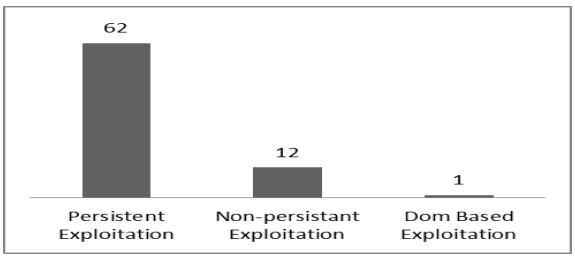
Code 03: DOM Based XSS vulnerable code

Exploitation Process:
Document Based XSS occurs in the processing or decision making pages. For that confirmation he will check that with normal XSS payloads. If the attacker confirmed that with DOM Based Cross Site Scripting Vulnerability then he may change or add malicious queries after the perimeter of that vulnerable page. E.g.
http://www.domxssvulnsite.com/registration.php?pageID=6
http://www.domxssvulnsite.com/registration.php?page=<script><h1> alert ("These Page is Vulnerable to XSS") </h1> </script>
When these query is executed by the server it will block the full web page with our inputted message. Attacker may also steal the session cookies or login credentials from session hijacking of these kind of vulnerability exploitations.

IV. RESULTS AND STATISTICS

In this study, we have formulated the sample size mechanism using universal calculator provided by G.Power 3.1.9.2. In this research, we examined 143 educational websites of Bangladesh. Only 13 sites were found free from the weakness of LFD. We found four types of LFD vulnerability in 130 websites in the sample. We had chosen manual black box testing approach to collect data for this study. We analyzed this data set, based on LFD exploitation type, level of access in host system, and level of risk created by LFD vulnerability. The analysis is discussed below:

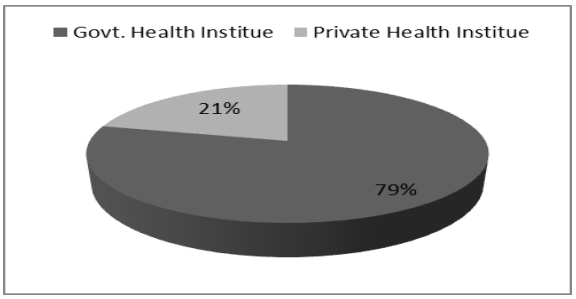
A. Analysis based on LFD exploitation type:
Graph 01 shows that among 133 XSS vulnerable educational websites, 90 sites can be exploited by the general XSS exploitation. Base64 and Long Directory Traversal Exploitations type were found in 10 and 30 websites respectively. Finally, we found, only three websites could be exploited by HTTP POST Base Exploitations. After analysing the data, it is observed that a good number of educational websites in Bangladesh have been developed without adequate protection against XSS exploitation.



Graph 01: XSS Exploitation Types

B. Analysis based public and private sector:
The risk level is identified as the impact of compromising, critical information that could be retrieved from the websites, through XSS exploitation. We have categorized the levels of risk as

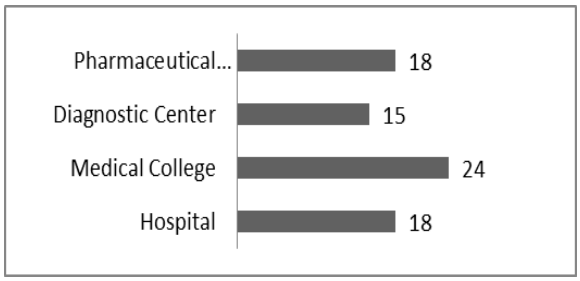
critical, high, medium, and low. The statistics of the four risk levels are shown in Graph 02.



Graph 01: Sector wise exploitation output

C. Analysis based on pharmaceutical, diagnostic, medical college and hospital:

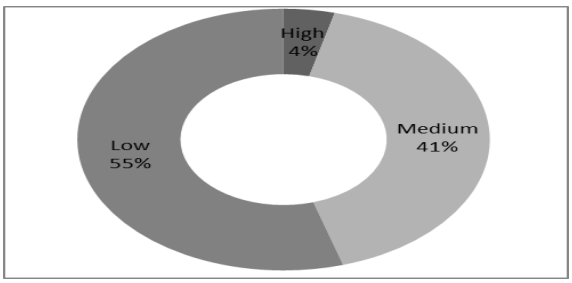
The risk level is identified as the impact of compromising, critical information that could be retrieved from the websites, through XSS exploitation. We have categorized the levels of risk as critical, high, medium, and low. The statistics of the four risk levels are shown in Graph 03.



Graph 03: Organization type wise exploitation

D. Analysis based on the level of risk resulting from XSS vulnerability:

The risk level is identified as the impact of compromising, critical information that could be retrieved from the websites, through XSS exploitation. We have categorized the levels of risk as critical, high, medium, and low. The statistics of the four risk levels are shown in Graph 04.



Graph 04: Level of Risk caused by XSS Vulnerability in Health Sector web apps

In this dataset, 31% of the risk is considered to be critical, since the intruder can get the full access to and full control of the host server and of the web application's admin panel. Then, 46% of websites are marked as high risk sites which can be exploited easily by the outsiders. Attackers can cause serious harm to the host server. In this review, 15% and 8% sites are considered as medium- and low-risk sites respectively, in terms of risk involved if LFD vulnerability is exploited.

CONCLUSION

Changes of time and technology people are more dependent on web application they complete their daily life important job using web application on the other hand attacker uses that an advantages because of different web vulnerabilities. . In this initiative, we have evaluated 100+ health sector websites in Bangladesh where the presence of XSS vulnerability found in 75 cases. In this paper shown that different XSS exploitation type, exploitation process. Cross site scripting vulnerability is high ranked weaknesses in the web application of health sector by which attacker can easily theft the user cookies and by using that cookie attacker can manipulate user data, records, password, and information. The effects of these vulnerabilities are in a high risk that every sector of Bangladesh can be facing a thread in their near future.

REFERENCES

[1] T. Farah, M. Shojol, M. Hassan and D. Alam, "Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF," 2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP), Konya, 2016, pp. 74-78.

[2] I. Yusof and A. S. K. Pathan, "Preventing persistent Cross-Site Scripting (XSS) attack by applying pattern filtering approach," The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), Kuching, 2014, pp. 1-6.

[3] R. Wang, X. Jia, Q. Li and S. Zhang, "Machine Learning Based Cross-Site Scripting Detection in Online Social Network," 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), Paris, 2014, pp. 823-826.

[4] I. Yusof and A. S. K. Pathan, "Mitigating Cross-Site Scripting Attacks with a Content Security Policy," in Computer, vol. 49, no. 3, pp. 56-63, Mar. 2016.

[5] Chun, S., Jing, C., ChangZhen, H., JingFeng, X., Hao, W. and Raphael, M., 2016. A XSS Attack Detection Method based on Skip List. International Journal of Security and Its Applications, 10(5), pp.95-106.

[6] T. S. Rocha and E. Souto, "ETSSDetector: A Tool to Automatically Detect Cross-Site Scripting Vulnerabilities," 2014 IEEE 13th International Symposium on Network Computing and Applications, Cambridge, MA, 2014, pp. 306-309.

[7] G. Dong, Y. Zhang, X. Wang, P. Wang and L. Liu, "Detecting cross site scripting vulnerabilities introduced by HTML5," 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chon Buri, 2014, pp. 319-323.

[8] Ding Lan, Wu ShuTing, Ye Xing and Zhang Wei, "Analysis and prevention for cross-site scripting attack based on encoding," 2013 IEEE 4th International Conference on Electronics Information and Emergency Communication, Beijing, 2013, pp. 102-105.

[9] M. K. Gupta, M. C. Govil and G. Singh, "Predicting Cross-Site Scripting (XSS) security vulnerabilities in web applications," 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), Songkhla, 2015, pp. 162-167.

[10] B. Mewara, S. Bairwa and J. Gajrani, "Browser's defenses against reflected cross-site scripting attacks," 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, 2014, pp. 662-667.

[11] G. Shanmugasundaram, S. Ravivarman and P. Thangavellu, "A study on removal techniques of Cross-Site Scripting from web applications," 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Chennai, 2015, pp. 0436-0442.

[12] V. K. Malviya, S. Saurav and A. Gupta, "On Security Issues in Web Applications through Cross Site Scripting (XSS)," 2013 20th Asia-Pacific Software Engineering Conference (APSEC), Bangkok, 2013, pp. 583-588.

[13] M. T. Louw and V. N. Venkatakrishnan, "Blueprint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers," 2009 30th IEEE Symposium on Security and Privacy, Berkeley, CA, 2009, pp. 331-346.

[14] D. Alam, M. A. Kabir, T. Bhuiyan and T. Farah, "A Case Study of SQL Injection Vulnerabilities Assessment of .bd Domain Web Applications," 2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec), Jakarta, 2015, pp. 73-77.

[15] D. Alam, T. Bhuiyan, M. A. Kabir and T. Farah, "SQLi vulnerabilty in education sector websites of Bangladesh," 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec), Cape Town, 2015, pp. 152-157.

★★★