Security Measures to Prevent Cross Site Scripting Attacks from a Bangladeshi Perspective

S M Ishraq ul Islam
ID: 1935362050
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
ishraq.islam01@northsouth.edu

Md. Abdul Hai Al Hadi
ID: 1835148050
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
hadi.mbstu@gmail.com

Syed Shahir Ahmed Rakin
ID: 1925417050
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
rakin.ahmed2000@gmail.com

Abstract— Website hacking is well known term all over the digital world. So, security is an essential thing on websites or online application. As various process or technique for security on website increasing day by day, but internet networks fail to provide security for websites for some vulnerabilities. In our research study, we find some vulnerabilities, such as Cross site scripting, SQL Injection Attack, Denial of service and distributed denial of service attacks, Threat from Key Logging, IP Spoofing, Non-binding Spoofing, Binding Spoofing, Buffer Overflow, Format String attack, Virus and worms, Password cracking, Phishing & Pharming. This paper provides two potential way to help secure websites from Cross site scripting.

KEYWORDS—WEBSITE HACKING, WEBSITE VULNERABILITIES, CROSS SITE SCRIPTING.

I. INTRODUCTION

Security is an essential and more important thing on websites. The demand for communication made the web technology an essential one. The importance of websites and its security increasing day by day, but internet networks fail to provide security for websites. There are two categories of web security namely web browser security and web application security [5]. Web browser security doesn't the major problem comparing to web application or websites security. Web servers are one of the most targeted objects of an organization [6]. Securing a web server is as important as securing the website or web application itself and the network around it. If we have a secure web application or site and an insecure web server or vice versa, it still puts our business at a huge risk. It is not an impossible task [6]. We will discuss some of the vulnerable online attacks commonly occurs in websites and try to provide solution for preventing such attacks by using some tools.

II. EXISTING USE

All the web pages we access are stored or hosted in a Web Server somewhere on the planet. All these hosting servers have been interconnected so that they can call web pages very easily from each other. This interconnecting is called as the Internet and all the web pages that reside on these hosting servers build up the Word Wide Web (WWW). To improve or enhance the versatility of the World Wide Web it became clear that a technique or procedure was need to enable better exchange of information between the client and the hosting server. The web hosting server can pass information to another application record and respond base on the information entered by the end user [7]. It makes use of what is called the Common Gateway Interface (CGI). All Web pages containing HTML forms normally use this technology to pass information entered by the end user to the hosting server.

Web hosting servers have default technique or procedures to fetch web pages [7]. We can alter these default technique or procedures very easily adding a Server-Side Script. When a web hosting server first receives a request, if the requested web page calls a server-side script then the server executes the script and attach the result to the web page which is to be sent to the end user. These server-side scripts are logical and sophisticated manner. They can analyze end user's behavior or motive from the data he has sent via the request and if required they can interact with a database to fetch some stored results or to store what the user has sent. There are some common scripting languages are PHP (Hypertext Pre-processor), ROR (Ruby on Rails), ASP (Active Server Page) and CFM (Cold Fusion). Of these, PHP and ROR are the only languages that are Open Source. This means that the engine that drives the hosting server is freely available and has been collaboratively developed by the people who use it frequently.

III. WEAKNESS

We find some vulnerabilities; online attacks commonly occur in Web sites.

1. Denial of service and distributed denial of service attacks.
2. Threat from Key Logging.
3. IP Spoofing.
4. Non-binding Spoofing
5. Binding Spoofing
6. Buffer Overflow
7. Format String Attack.
8. SQL Injection Attack.

9. Cross site scripting.
10. Virus and worms.
11. Password cracking
12. Phishing & Pharming

IV. CROSS SITE SCRIPTING VULNERABILITIES

A. *What is XSS?*

The Cross-Site Scripting (XSS) is a method of attack that injects malicious code into existing web pages or sites which is then run on the client-side computer of the visitors to that website. This method of attack targets the users instead of the webpage. The webpage is used as an attack vector to reach the victim which is the visitor to that webpage.

It works as follows. A bad actor infiltrates a webpage or a website and places malicious code in some part of the website. When accessing that webpage, the visitor's browser is tricked into running that piece of code, thus executing the attack.

It is one of the most exploited vulnerabilities in the internet [1]. For large websites, it is difficult to plug all XSS vulnerabilities and therefore it makes it easy for attackers to find attack vectors and exploit them before they are found out.

The Cross-Site Scripting (XSS) is a method of attack that injects malicious code into existing web pages or sites which is then run on the client-side computer of the visitors to that website. This method of attack targets the users instead of the webpage. The webpage is used as an attack vector to reach the victim which is the visitor to that webpage.

B. *Potential Harm?*

XSS can result in a range of harm. Since it can potentially allow the running of arbitrary code, it can lead to attacker having access to many functions and data from the users' machines. This includes, but isn't limited to stealing user cookies and use it to impersonate the user, modify the page's looks and functionality, connect to another website or server in the background and steal data.

An XSS attack can be conducted using JS, Flash, Visual Basic, CSS and HTML. There are two steps to a successful XSS attack

- The planting of malicious code on a website. This usually relies on the website having a user input.
- The victim visiting the website. In some cases, custom websites are used to target specific victims to snatch the data to be used against them in the foreseeable future.

HTML code, which is used to build a website contains tags. Tags contain blocks of information that are interpreted based on the nature of the tag. One such tag is the <script> tag which is interpreted by the browser as code to be run. In this manner, malicious code snippets could be inserted into any user input fields on a website. If this is stored as is, when this HTML snippet is served to another visitor to the website, that script tag is read by the browser as code to be run and it thus executes the malicious code, thereby harming the user.

C. *How Does It Affect Bangladesh?*

Bangladesh is currently seeing a boom in the tech sector. With many industries now looking to digitize their operations, both internal and external demand for web-based technologies has skyrocketed. Many companies now provide all or a portion of their services through online portals and websites. Many companies also use web-based technologies for internal functions such as email and CRM. This growth of demand has led to a rapid rise in web development and the launch of many new websites and services.

But with this new wave of web-based services comes the risk associated with websites, the key of which is the XSS. This is particularly harmful for the Bangladeshi populace due it's relative less familiarity with technology. XSS attacks can be extremely difficult to spot. Bangladeshis are also currently performing more and more complex and important tasks online, such as business transactions, banking and information sharing, which makes the threat of XSS attacks even more dangerous.

The lack of technological familiarity of the Bangladeshi populace combined with the increasing dependency on technology could lead to the severe loss of user data and bring financial harm. This in turn could reduce the trust people place on internet-based systems, thus stunting growth in the sector.

D. *How To Prevent It?*

Since XSS involves the use of malicious code to which enables the attacker/hacker to gain access to many of the functions of the site. However, there are cases where the attacker/hacker puts the code but it is still in doubt whether the attacker is operating from the system itself or not.

Validating input is the process of ensuring an application is rendering the correct data and preventing malicious data from doing harm to the site, database, and users. While whitelisting and input validation are more usually related with SQL injection, they can also be used as an extra method of prevention for XSS. Whereas blacklisting, or disallowing certain, predetermined characters in user input, disallows only known bad characters, whitelisting only allows known good characters and is a better method for preventing XSS attacks as well as others.

Input validation is especially helpful and good at preventing XSS in forms, as it prevents a user from adding special characters into the fields, instead refusing the request. However, as OWASP maintains, input validation is not a primary prevention method for vulnerabilities such as XSS and SQL injection, rather it is a first-stage prevention method in case any of the weaknesses is discovered by the attacker and take counter-measures to tackle it.

In JavaScript, there are methods such as checkValidity() method and setCustomValidity() which checks whether the input is valid or not. If the input is found to be valid, then the progression goes, else it gives a validation message in the case of an invalid data (given that the attackers entered invalid data).

Rao et al. (2017) had proposed a strategy in order to prevent XSS attacks by the use of code filtering algorithm which works by sanitization of user input that may have any scripting tags or codes, as a result, preventing the scripting tags or codes from being stored or executed from the database of the web application. Results showed that an attacker normally checks whether a web application is vulnerable or not before the attacker tries to inject malicious code into the server. [2]

In the observations of the researchers, it has been found that while other forms of attack affect the server side; the XSS makes impacts of the client side (user) i.e. the real manipulation is inside the target's web browser while the server side is where the vulnerability is housed upon. It was also noted that web applications take in multifaceted Hypertext Markup language (HTML) input from operators and the inconsistent web browser performances serving as vectors for fruitful XSS attacks made the encounter very problematic to unravel. [3]

Persistent XSS attacks are basically involved with the message boards web applications which houses weak input validation mechanisms. This second category, which is termed as non-persistent XSS attack takes advantage of the vulnerability that appears in a web application when it makes use of the information given by the user for the purpose of generating an outgoing or landing page for that user. As a result, instead of storing the malicious code embedded into a message by the attacker, the malicious code itself is directly thrown back to the user with the assistance of a third-party mechanism. [4]

E. Preventive Measures For Bangladesh

Since existing technical solutions already exist for preventive measures, what needs to be implemented are strong policies for the application of the following measures. In this paper we propose the following measures

- **Method 1:** Stipulate the usage of the aforementioned safety measures during the creation of contracts for website development.
- **Method 2:** Require websites made for handling sensitive consumer data pass a system audit to check for XSS vulnerabilities.

Method 1 requires any contracts made to design websites to include safety checks for XSS attacks. This way any websites designers will be legally bound to provide the necessary safety precautions. It adds an extra layer of incentive for website developers to be aware about XSS attacks. Standard templates of such contracts could be designed which can then be used as a baseline for newer contracts.

Method 2 is implemented once website development has neared completion and is ready for deployment. At this point, if the website has been designed to handle sensitive user data, it must pass an audit of XSS safety precautions carried out by some regulating board.

These steps can be implemented in two phases. Phase one involves conducting nationwide awareness campaigns that inform people willing to launch websites about the necessity of XSS safety precautions. It provides guidelines on necessary preventive measures and what to look out for when launching new websites.

Phase two consists of setting up rigid audit systems to ensure proper safety measures have been taken and used. This involves employing security experts who are able to review a website and check for potential XSS exploits and only certify a website for launch once all safety criteria have been met.

V. Conclusion

XSS attacks are one of the most common forms of attack on the internet. It's potential harm to a growing internet user base such as Bangladesh is huge. Which is why it is critical to take policy level measures to ensure such attacks are prevented. This paper proposes two such measures to help secure websites from such attacks.

References

[1] Grossman, J., 2007. Whitehat website security statistics report. Retrieved March, 8, p.2010.

[2] Vonnegut, S., "3 Ways to Prevent XSS," Checkmarx, 09-Oct-2017. [Online]. Available: https://www.checkmarx.com/2017/10/09/3-ways-prevent-xss/.

[3] John-Otumu, Adetokunbo & Imhanlahimi, A & Akpe, R. (2018). Cross Site Scripting Attacks in Web-Based Applications: A Critical Review on Detection and Prevention Techniques. 25-35.

[4] Garcia-Alfaro, Joaquin & Navarro-Arribas, Guillermo. (2007). Prevention of Cross-Site Scripting Attacks on Current Web Applications. http://hdl.handle.net/10363/606. 4804. 10.1007/978-3-540-76843-2_45.

[5] Mirdula, S. and D. Manivannan, Security Vulnerabilities in Web Application-An Attack Perspective [J]. International Journal of Engineering and Technology, Vol.5, No.2, pp. 1806-1811, 2013.

[6] Yaashuwanth .C, Dr. R. Ramesh," Attacks in WEB Based Embedded Applications", IJCSI International Journal of Computer Science

Issues, Vol. 7, Issue 6, November 2010

[7] Li, L. and Lee, G. DDoS attack detection and wavelets. Telecommunication Systems, pp 435-451.,(2005).

[8] Jayamsakthi Shanmugam1, Dr. M. Ponnavaikko2," Cross Site Scripting-Latest developments and solutions: A survey", Int. J. Open

Problems Compt. Math., Vol. 1, No. 2, September 2008.

[9] Joe, M. M., & Ramakrishnan, D. B. (2014). A survey of various security issues in online social networks. *International Journal of Computer Networks and Applications, 1*(1), 11–14.

[10] Lorenzo Lamberti,Analysing and protecting against existing cyber attacks.

[11] Kharb, Dr. Latika. "International Journal of Engineering and Management Research." Cyber Crimes Becoming Threat to Cyber Security, vol. 7, no. 2, pp. 48–51., www.ijiris.com.

[12] Muragendra Tubake," Cyber Crimes: An Overview", Online International Interdisciplinary Research Journal, {Bi-Monthly}, ISSN2249-9598, Volume-III, Issue-II, Mar-Apr 2013

[13] Agrawal, Manish, Kailash Patidar, Rishi Kushwah, and Sudesh Chouhan. "Computation analysis and review based on cross-site scripting attack." (2019).

[14] MacDermid, Kenny. "Prevention of cross site scripting attacks using automatic generation of content security policy headers and splitting of content to enable content security policy." U.S. Patent 10,318,732, issued June 11, 2019