MITIGATION AGAINST CROSS SITE SCRIPTING (XSS) ATTACK – A SURVEY

Laxman Khokhar[1], Snehal Sathwara[2]

[1] Student, Department of Computer Engineering, Marwadi University, Gujarat, India
[2] Assistant Professor, Department of Computer Engineering, Marwadi University, Gujarat, India

ABSTRACT

As we know the most of web applications are used by an organizations to grow their business. The security of web application is most essential concern for the continuous business without interruption of any cyber attack. The Cross Site Scripting often known as XSS attack is common injection vulnerability which is also specified in OWASP (Open Web Application Security Project) Top 10 Web Application Risk in 2010, 2013, 2017. By exploiting the Cross Site Scripting vulnerability, an attacker may capture sessions of users and also able to change, view and delete the web application data, execute the malicious script over the web application, and manipulate the victim to attack other targeted servers. This paper describes the types of Cross Site Scripting and demonstrate the exploitation of the Cross Site Scripting vulnerability on DVWA (Damn Vulnerable Web App) website. Also, the paper include a mechanism to apply the security against Cross Site Scripting vulnerabilities by implementing the Web Application Firewall (WAF) such as Modsecurity. We can deploy Web Application Firewall (WAF) to safeguard web applications. It protect the web applications from various types of attacks such as cross-site-scripting, SQL injection, and cross-site request forgery etc. A Web Application Firewall is providing the security against various vulnerabilities on OSI (Open Systems Interconnection) Layer 7 which is Application Layer. This firewall will work effectively when it is configured with proper rules. By writing the strong custom rules, it ensures the Web Application Firewall (WAF) correctly detects attacks and also block them if any rule violated or takes appropriate action.

Keyword: - *XSS vulnerability, Exploitation, Web Application Firewall, Modsecurity.*

1. INTRODUCTION

The Cross Site Scripting (XSS) is the commonest application layer vulnerability in which an attacker can executed the malicious script in victim's browser instead of server side and display the malicious content in victim's browser. The Cross Site Scripting vulnerability often found in various web application specially where the user provide the input to the web application such as search bar, various forms and feedback forms. In the Web application, there is a cross-site scripting vulnerability which enables an attacker to take benefits of web server scripts which inject JavaScript or HTML code to hijack user sessions in the victim's browser. Cross-site scripting has also been used to steal cookies, session id and also it can lead to Cross Site Request Forgery (CSRF) attack. There are mainly three types of cross site scripting:

- Stored XSS
- Reflected XSS
- DOM Based XSS

1.1 Stored XSS

Stored XSS attacks happens when the injected malicious script will get saved in the backend database like comment fields, Message forums, etc. and whenever any external user tried to access the web application functionality which required to fetch the data from mentioned fields then this malicious script get executed on victim machine[2].

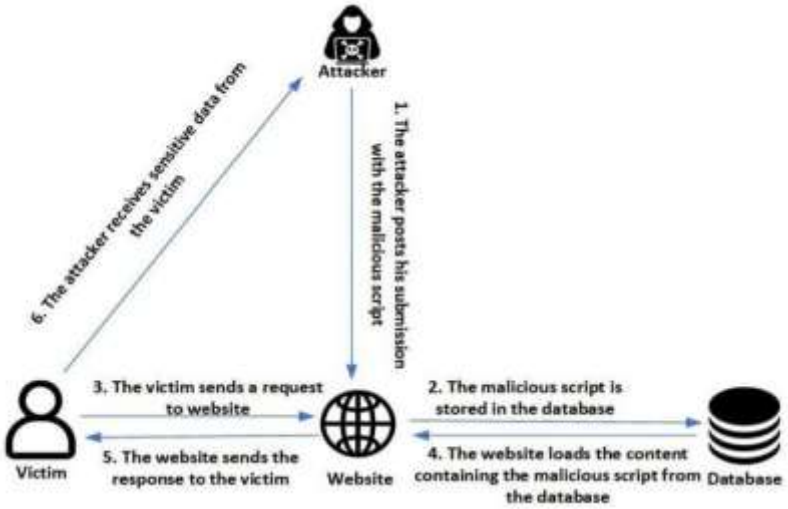

Fig -1: Process of Stored XSS

1.2 Reflected XSS

Reflected XSS attacks also called non-persistent attacks which occur when the malicious script is getting executed when user clicks on link which in turn execute malicious script on user machine [2].

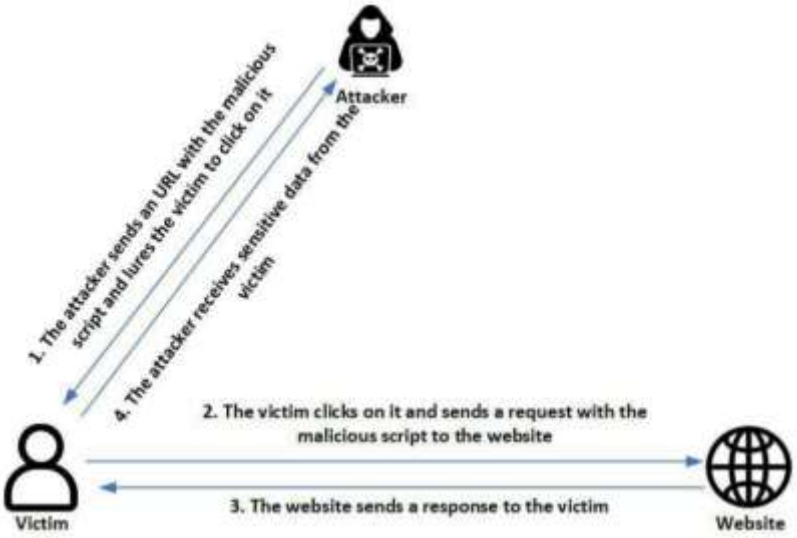

Fig -2: Process of Reflected XSS

1.3 DOM-Based XSS

DOM-Based cross site scripting attack is form of Advance XSS attack where user-side malicious code change the data in document object model. The web application reads these malicious data from the DOM and exposes results to the user. If these data are not treated correctly then the attacker will insert the malicious code that will be saved in the DOM and will get run whenever browser will try to fetch that data for processing [3].

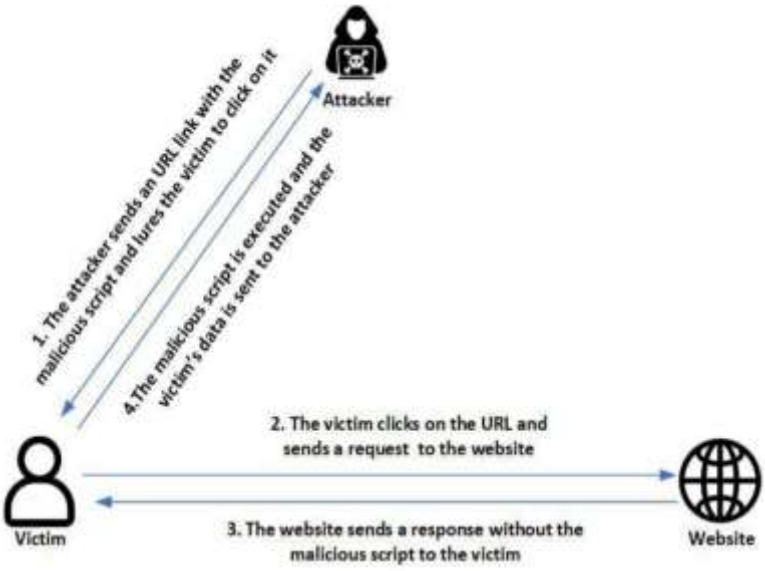

Fig -3: Process of DOM-Based XSS

2. OBJECTIVE

The aim of this survey paper is to provide protection against Cross Site Scripting (XSS) attack by implementing Web Application Firewall. The Modsecurity is a Web Application Firewall which can block the XSS attacks and keep website safe from such attacks.

3. RESEARCH METHODOLOGY

3.1 What is web application firewall (WAF):

The Web Application Firewall can be used to secure the web application by filtering the HTTP request and also monitoring the HTTP request-response flow over a web application. It protect the web applications from various types of attacks such as cross-site-scripting, SQL injection, and cross-site request forgery etc. A Web Application Firewall is providing the security against various vulnerabilities on OSI Layer 7 which is Application Layer [6]. There are several different types of Web Application Firewalls on the competition such as Barracuda, Fortinet, Imperva, Bee Ware, Breach Security, ModSecurity, Citrix, and F5. The ModSecurity is one of the most common Web Application Firewall. The ModSecurity is popular among LAMP environment as it is open source and works with Apache server [7].

3.2 Implement WAF based Architecture:

Now we are going to implement the WAF based architecture as show below:

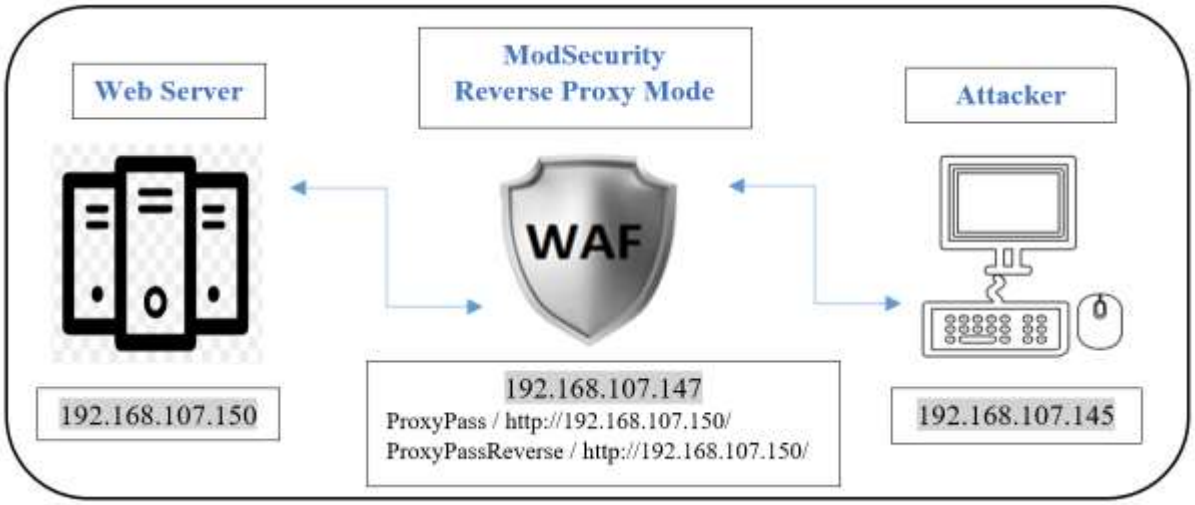

Fig -4: Proposed Methodology to apply security against XSS attack

On the web server, we have hosted a Deliberately Vulnerable Web Application (DVWA) which acting as our targeted website and having vulnerable code that can be exploited using XSS attacks [10].

On the firewall, we have installed ModSecurity with Mod-Proxy, so whenever an attacker trying to access the web server, then all the HTTP traffic first filtered at firewall and then further it will forwarded to the webserver [7][9].

On attacker's machine, we have installed Kali Linux as operating system and used Burp Suite to preform attacks [11]. The attack vectors are summarized in Table 1.

4. LITRATURE REVIEW

Victor Clincy and Hossain Shahriar in Web Application Firewall: Network Security Models and Configuration [16] presented the two types of WAF rule policy such positive and negative. A positive policy-based WAF contains the whitelisting rules which will only allow traffic that matches with the rules and blocked all other traffic which does not match. A negative policy-based WAF contains the blacklisting rules that generally allow all traffic through waf but only block the traffic if there is any malicious pattern that matches the blacklisting rule. Basically, many firewalls usually use either positive or negative rules but use both in rare cases. In a positive policy-based WAF, the HTTP(S) traffic are checked to ensure that the content is allowed. The whitelisting rules policy requires high level planning and in-depth experience of the web application and also requires correct and agreed whitelist input values. Fine tuning of the WAF can take a huge amount of resources, time and also detailed information of the system. In a negative policy-based WAF, the HTTP(S) traffic are reviewed, if any data/argument is detected as malicious then such HTTP(S) traffic is blocked. It may not be satisfactory to use this model only as hackers are often determining new ways to bypassing security policies. For instance, blacklisting certain input characters can not be protected as much as hackers may use complicate characters to bypass input validation during an attack.

Teddy Mantoro in Log Visualization of Intrusion and Prevention Reverse Proxy Server against Web Attacks [17] proposed WAF based architecture, in which Modsecurity is running with reverse proxy mode. The aim of the paper is to identifying the SQL Injection attack and providing the effective mitigation from the SQL injection attack. Also in the paper they have describes the log analysis of the traffic which is served by the Modsecurity. In log analysis they have observed that few Rules are violated against the SQL injection attacks and Modsecurity has blocked such HTTP requests.

5. EXPLOITATION OF XSS VULNERABILITY

We have taken the below attack vectors and trying to exploit the Cross Site Scripting vulnerability with Burp suite tool. We have successfully exploited the Cross Site Scripting vulnerability and performed the XSS attack over DVWA web application.

Table -1: List of XSS Attack Vectors

Sr. No.	Attack Vector
1.	<script> alert(‘XSS’)</script>
2.	<body onload="alert(11111)">
3.	<svg onload=prompt('XSS')>
4.	<sCrIpT>alert(11111)</sCrIpT>
5.	<b onmouseover=alert(‘XSS testing!‘)>
6.	click me!
7.	<img src=x onerror=alert(11111)>
8.	<frameset><frame src onload=alert(11111)>
9.	<svg/onload=alert(11111)>

6. FINDINGS

We have observed that the Cross Site Scripting (XSS) vulnerability can be patched via Web Application Firewall. The Modsecurity allows to write the custom rules which will block such malicious HTTP requests and keep website safe from the XSS attack [17].

7. CONCLUSIONS

The Cross Site Scripting vulnerabilities are mainly caused by unsuitable filtering of user inputs, such as the incorrect order in which the encoding method is applied. This paper explains the types of Cross Site Scripting and exploitation of the XSS vulnerability on DVWA website. Also observed that the Cross Site Scripting vulnerability can be patched via Web Application Firewall such as Modsecurity. The Modsecurity can monitor and filter HTTP requests and blocked the malicious HTTP requests. The Modsecurity allows to write the strong custom rules which will block such malicious HTTP request and keep website safe from XSS attack.

8. REFERENCES

[1]. Liu, M., Zhang, B., Chen, W., & Zhang, X. (2019). A Survey of Exploitation and Detection Methods of XSS Vulnerabilities. IEEE Access, 7, 182004-182016.
[2]. OWASP, “Cross-site Scripting (XSS)”, 2018. [Online] Available: https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) [Accessed 30-Dec-2019].
[3]. Acunetix, “Types of XSS: Stored XSS, Reflected XSS and DOM-based XSS”, 2019. [Online] Available: https://www.acunetix.com/websitesecurity/xss/ [Accessed 30-Dec-2019].
[4]. Jain, T., & Jain, N. (2019, March). Framework for Web Application Vulnerability Discovery and Mitigation by Customizing Rules through ModSecurity. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 643-648). IEEE.
[5]. Veracode, “CROSS-SITE SCRIPTING (XSS) TUTORIAL: LEARN ABOUT XSS VULNERABILITIES, INJECTIONS AND HOW TO PREVENT ATTACKS”, 2019. [Online] Available: https://www.veracode.com/security/xss [Accessed: 30-Dec-2019].
[6]. Cloudflare, “What is a WAF? | Web Application Firewall explained”, 2019. [Online] Available: https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/ [Accessed: 30-Dec-2019].

[7]. Jesin A, "How To Set Up ModSecurity with Apache on Ubuntu 14.04 and Debian 8", 2015. [Online] Available: https://www.digitalocean.com/community/tutorials/how-to-set-up-modsecurity-with-apache-on-ubuntu-14-04-and-debian-8 [Accessed: 30-Dec-2019].
[8]. Justin Ellingwood, "How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 14.04", 2014. [Online] Available: https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04 [Accessed: 30-Dec-2019].
[9]. Mateusz Papiernik, "How To Use Apache as a Reverse Proxy with mod_proxy on Ubuntu 16.04", 2017. [Online] Available: https://www.digitalocean.com/community/tutorials/how-to-use-apache-as-a-reverse-proxy-with-mod_proxy-on-ubuntu-16-04 [Accessed: 30-Dec-2019].
[10]. Hacking Tools, "Damn Vulnerable Web Application (DVWA)", 2019. [Online] Available: http://www.hackingtools.in/free-download-damn-vulnerable-web-application-dvwa/ [Accessed: 30-Dec-2019].
[11]. PortSwigger Ltd., "Burp Suite", 2019. [Online] Available: https://www.portswigger.net/burp [Accessed 30-Dec-2019].
[12]. Singh, J. J., Samuel, H., & Zavarsky, P. (2018, April). Impact of Paranoia Levels on the Effectiveness of the ModSecurity Web Application Firewall. In 2018 1st International Conference on Data Intelligence and Security (ICDIS) (pp. 141-144). IEEE.
[13]. OWASP, "OWASPTop10-2010-PressRelease", 2010. [Online] Available: https://www.owasp.org/index.php/OWASPTop10-2010-PressRelease [Accessed 30-Dec-2019].
[14]. OWASP, "Category:OWASP Top Ten Project", 2013. [Online] Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#OWASP_Top_10_for_2013 [Accessed 30-Dec-2019].
[15]. OWASP, "Top 10-2017 Top 10", 2017. [Online] Available: https://www.owasp.org/index.php/Top_10-2017_Top_10 [Accessed 30-Dec-2019].
[16]. Clincy, V., & Shahriar, H. (2018, July). Web Application Firewall: Network Security Models and Configuration. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 835-836). IEEE.
[17]. Mantoro, T. (2013, September). Log visualization of intrusion and prevention reverse proxy server against Web attacks. In 2013 International Conference on Informatics and Creative Multimedia (pp. 325-329). IEEE.
[18]. Github, " PayloadsAllTheThings / XSS Injection/ ", 2015. [Online] Available: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20Injection[Accessed 30-Dec-2019].