# Client Side Solution for Protection of Web Applications from Cross Site Scripting Attack

**P.Risha Hebiya**#1**,J.Ganesh**#2

*P.G. Student, Dept. of Computer Science and Engineering, Arasu Engineering College, Kumbakonam, India*[1]

*Assistant Professor, Dept. of Computer Science and Engineering, Arasu Engineering College, Kumbakonam, India*[2]

*rishahebi@gmail.com*#1*,jaygan86@gmail.com*#2

*Abstract:* **Cloud computing is an important model for delivering valuable business services to the cloud users. Cloud service consider as online Storage where the data is remotely stored, managed and backed up. The user to store their personal files and business oriented data in online and access them from anywhere via the internet. Cross Site Scripting (XSS) is a widespread breed of web vulnerabilities which allows attacker to inject malicious code from their untrusted websites that are being accessed by the multiple user. When the user want to access those affected website at the time the malicious script will be automatically downloaded into the each user web browser and then the script will be executed. Then the attacker to collect the user credential information from that malicious code execution. So that, to prevent XSS vulnerabilities (First Order attack) by using cloud cross site scripting filter approach which contains multiple function. The cloud audit server performs the user authentication and monitors the stored data in cloud. The results shows proposed mechanism to enhanced the data storage security in cloud and efficiently detect the scripting vulnerabilities.**

*Keywords:* **cloud storage, XSS, security and scripting vulnerabilities.**

## I. INTRODUCTION

As the use of the Internet has grown, so the number of attacks which attempt to use it for different purposes like personal information gathering and collect business oriented financial data. Ahmed Elhady Mohamed et al [3], one vulnerability which has become commonly exploited is known as cross-site scripting. This class of vulnerability attack occurs when an attacker insert malicious code into a web application and gain access from unauthorized information. In such case, the victim is unaware that their information is being transferred from a site to another site controlled by the attacker.

The scripting languages allowed webpages to become more dynamic. These scripting languages allow users to exchange a quantity of information with web servers, and this information is often sensitive. Cross site scripting is a widespread breed of web vulnerabilities which allows hackers to inject malicious code from their untrusted websites into the webpages that are being viewed by unknown victims. S.Krishnaveni [10], XSS is an attack on the privacy of clients of a particular web site which can lead to a total breach of security when customer details are stolen or manipulated.

The XSS attack involves three parties – the attacker, a client and the web site. The goal of the XSS attack is to steal the client cookies, or any other sensitive information, which can identify the client with the web site.

## II. RELATED WORK

Adam Kie zun et al [1], proposed a technique for finding security vulnerabilities in Web applications. To consider SQL Injection (SQLI) and cross site scripting (XSS) attacks are widespread forms of attack in which the attacker crafts the input to the application to access or modify user data and execute malicious code. To implemented a tool Ardilla evaluated the web vulnerabilities. Ardilla's taint propagation track the flow of tainted data through the database. The user of Ardilla needs to specify the type of attack (SQLI, first order XSS, or second-order XSS), to analyze and the initial database state. Tejinder Singh Mehta et al [2], focused the possibilities of securing web applications on client side as well as on server side. To minimize theft space for taint contents by using QualysGuard (WAS) tool.

In cloud computing scenario QualysGaurd WAS acts as a software as-a-service (SAAS). QualysGuard WAS automates the websites scanning process QualysGuard WAS tool is the best tool to check browser with three options, Basic Scan: (browser issues). Intermediate Scan (browser issues and system settings) Advanced Scan (browser issues and system settings to reflect the issues with fix the issue). Cong Wang et al [4], to ensure the correctness of user's data in the cloud. To detect any unauthorized data modification and corruption in stored data in cloud environment. So, they provide efficient mechanisms for dynamic data verification operation to achieve their goals such as Storage correctness and Fast localization of data error. Jin Li et al [5], describe the integrity of data storage in cloud computing. To reduce the computational cost at user side during the integrity verification of their data. They proposed OPoR scheme is proved security against reset attacks and reduces computation overhead for tag generation on client side. C wang et al [6] describe the cloud data storage security with Privacy Preserving Public Auditing with watermarking for data Storage security in cloud computing.

The watermarking processes achieve the very high level of security so the data or images cannot be identified by the attacker in the cloud. It supports dynamic data where the user can perform various operations on data like insert, update and delete. Mohammad Reza et al [7] explore the possibility of using the selective monitoring approach instead of the exhaustive checking method. In particular, they proposed a study of potential selective monitoring schemes. In a selective monitoring scheme, they do not monitor every read/write post or every user in the OSN (Online Social Network).

## III. OVERVIEW OF CROSS SITE SCRIPTING

The two most common risks in the Web environment, injection namely SQL injection, which lets attackers alter SQL queries sent to a database and cross-site scripting (XSS), are also two of the most dangerous Nuno Antunes et al [8]. XSS permits the attacker to inject malicious code (Java Script) into the user system or Targeted cloud storage server. The reason for that the user trusts the remote storage server to be fully secured. A. Duraisamy et al [9], once the attacker to import the malicious code into the targeted cloud server means it will be spread onto the entire users whose are all to access that cloud server. Then the malicious code will be automatically running under the each one web browser until it was removed from that. The client browser execute the malicious code and collect the user information. To consider the following different kind of cross site scripting vulnerabilities and their functions.

### A. TYPES OF CROSS SITE SCRIPTING

The introduction of scripting languages allowed webpages to become more dynamic. Server-side scripting languages such as PHP and ASP enabled web developers to interact with resources that reside on the server such as files and databases. Client-side scripting languages, e.g. JavaScript, execute in the user's web browser and provide similar functionality on the client computer. To consider the following two types of cross site scripting vulnerabilities in online environment.

#### a) FIRST ORDER XSS

The common type of XSS is Reflected XSS attack. In Reflected XSS, the attacker's payload script has to be part of the request which is sent to the web server and reflected back in such a way that the HTTP response includes the payload from the HTTP request. Using Phishing emails and other social engineering techniques, the attacker attract the victim to unintentionally make a request to the server which contains the XSS payload and ends-up executing the script that gets reflected and executed inside the user browser. In Reflected XSS attack, the attacker needs to deliver the payload to each victim – social networks are often conveniently used for the distribution of Reflected XSS attacks.

#### b) SECOND ORDER XSS

The dangerous type of XSS is Stored (Persistent) XSS attack. Stored XSS attacks involves an attacker injecting a malicious script (referred to as the payload) that is permanently stored on the target storage server (within a database). When a victim navigates to the affected web page in a browser, the XSS payload will be served as part of the web page. This means that victims will unintentionally end-up executing the malicious script once the page is viewed in a user browser. At last attacker collect the user credential information from that malicious code running under the user browser.

## IV. SECURITY MODEL

In this case to provide the security to the cloud storage server by avoiding the persistent XSS attacks as well as to provide the web application security by eliminating the reflected XSS attack. The User want to store valuable business data and their personal data in cloud storage server. The Cloud Audit Server responsible for user authentication and to eliminate the scripting vulnerabilities by using anti XSS framework. The Cloud Storage Server used to store the user data in remote location wherever user want to upload or download those stored data via internet. The attacker is a malicious user of the website who intends to launch an attack on the victim by exploiting an XSS vulnerability in the website.

The following diagrammatic representation of XSS attack function in cloud environment. The user want to access the cloud storage means first to register into the cloud audit server after that to access those cloud storage.

## V. PREVENTION OF XSS ATTACKS

The best way to prevent Persistent XSS is to make sure that all user input is properly sanitized before it gets stored permanently on the web server, and as a second line of defense, make sure that the static content presented to users is also sanitized. As malicious scripts can be encoded in various ways, sanitization parsers should take encoding into consideration, as well as various ways to inject code, when searching for payloads in the content to be stored.

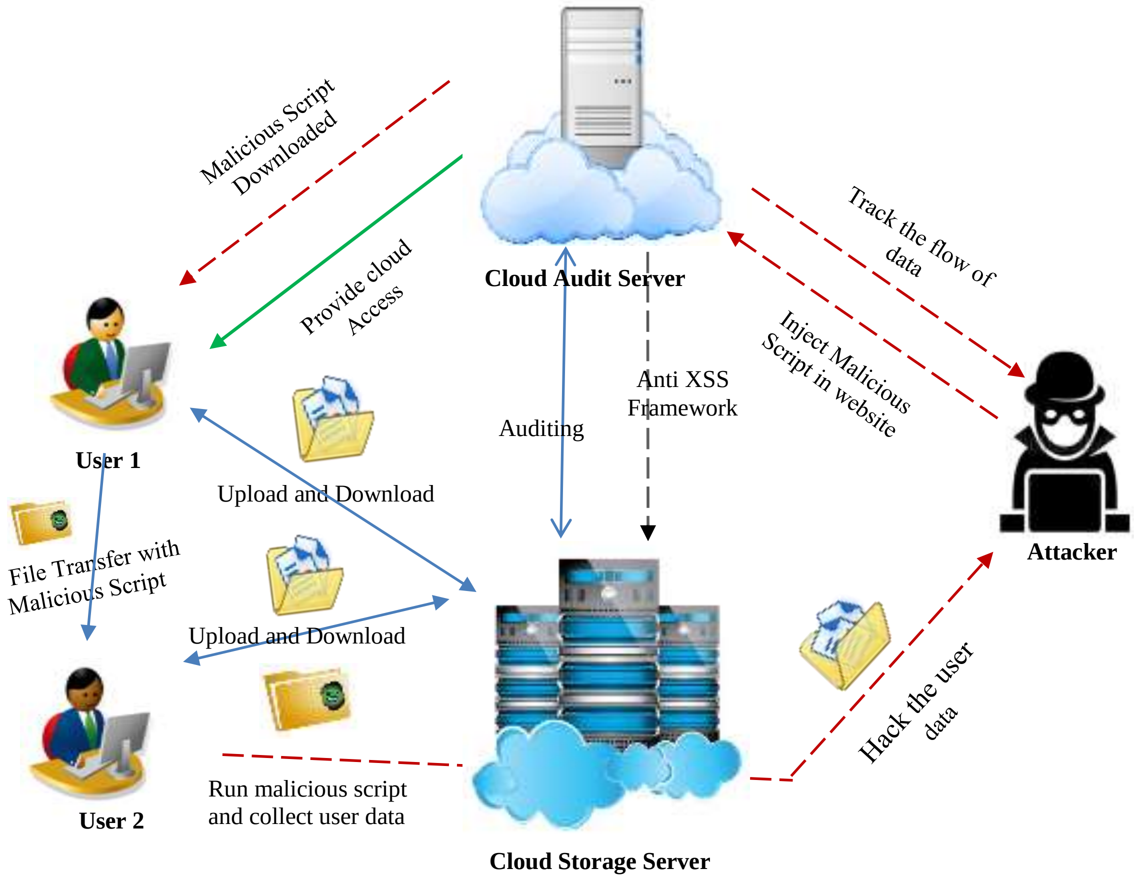

**Fig.1 Cross Site Scripting attack in cloud**

### A. Anti XSS Framework

A web security mechanism point of view the whole HTTP request sent to a application must be considered as input and not only the parameter values that are served by users into HTML input fields. Cross-Site Scripting filtering for JavaScript means scanning a data stream for specific string patterns considered dangerous and then take appropriate actions like transformation or deletion. There are many character encodings schemes available that are used to represent foreign language characters. The character encoding schemes of the input data for a Web application is normally indicated in the request header generated by the client browser. The first step is to normalize the input data to specific character encoding. An important concept is that all links that are statically embedded in a web page can be considered safe with respect to Cross Site Scripting attacks. The attacker does not directly use static links to encode sensitive user data. The reason is that all statically embedded links are composed by the server before any malicious code at the client can be executed.

A Cross Site Scripting attack, on the other side, can only succeed after the page has been completely retrieved by the browser and the script interpreter is invoked to execute malicious code on that page. When the user visiting that page at the time the malicious script will be automatically downloaded into the user system. Then to collect the user credential or original information using that malicious script execution on user browser.

## VI. PERFORMANCE ANALYSIS

The given analysis graph represent the better performance results compared to the existing classifier/filter. To consider highest level of security comparison with the existing Noxes tool and proposed Anti XSS. Engin Kirda et al [11], Noxes used for detecting scripting vulnerabilities in web application. We implemented anti XSS framework for preventing the cross site scripting vulnerabilities. The comparative analysis (Table I), it is found that the predictive accuracy shown by Anti XSS filters with Basic Noxes, the Anti XSS provide higher accuracy than the Basic Noxes. Table I Show the comparison of performance analysis between the existing and proposed approaches.

TABLE I Performance of Anti XSS

| **Technique** | **Evaluation Criteria** | | | |
|---|---|---|---|---|
| | **Training time (sec)** | **Testing time (sec)** | **Training Accuracy (%)** | **Testing Accuracy (%)** |
| Anti XSS | 0.2043 | 0.4432 | 0.6330 | 0.8204 |
| Noxes Tool | 0.3322 | 0.5433 | 0.7784 | 0.9976 |

## VII. CONCLUSION AND FUTURE WORK

In this paper we analysis different types of cross site scripting attacks and their variety of functionality. The proposed Anti XSS framework to protect cloud data storage server against cross site scripting attacks (Non Persistent attack) and improve the storage security. In our paper the cloud audit server performs the auditing operation and monitors the cloud storage server simultaneously. The Anti XSS framework to detect and prevent the scripting vulnerabilities in server side by cloud audit server. To remove the cross site scripting vulnerabilities from malicious webpages by Anti XSS.

## ACKNOWLEDGMENT

**P.Risha hebiya** received her B.Tech degree in Information Technology from Adhiparasakthi Engineering College, Melmaruvathur in 2014 and she is currently pursuing M.E Computer Science and Engineering in Arasu Engineering College kumbakonam, Tamilnadu, India. Her areas of interest are: Web Application Security, Cloud security and big data.

**J.Ganesh** received his B.E degree in computer science and engineering from Anjalai Ammal Mahalingam Engineering College, Kovilvenni and M.E Computer Science and Engineering from Kumaraguru College of Technology, Coimbatore. He is currently pursuing PhD in Anna University, Chennai and currently working as a Teaching Fellow in Arasu Engineering College, kumbakonam, Tamilnadu, India.

## REFERENCES

[1] Adam Kie˙zun, Philip J. Guo, Karthick Jayaraman and Michael D. Ernst, ”Automatic Creation of SQL Injection and Cross-Site Scripting Attacks”, IEEE International Conference on Software Engineering,Vol.10,pp.199-209,August 2010.

[2] Tejinder Singh Mehta , Sanjay Jamwal, ”Model To Prevent Websites From XSS Vulnerabilities”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol.6, pp.1059-1067, 2015.

[3] Ahmed Elhady Mohamed,” Complete Cross-site Scripting Walkthrough”, website: www.infosec4all.tk , 2008.

[4] Cong Wang, Qian Wang, Kui Ren and Wenjing Lou,” Ensuring Data Storage Security in Cloud Computing”, IEEE Transactions on Computers, Vol.3, pp.1-9, Jul 2009.

[5] Jin Li, Xiao Tan, Xiaofeng Chen, Wong D.S and Fatos Xhafa "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource- Constrained Devices", IEEE Transactions on Cloud Computing, Vol.3, pp.195 – 205, April 2015.

[6] Wang.C, Chow.S.M, Wang.Q, Ren.k and Lou.W, “Privacy-Preserving Public Auditing for Secure Cloud Storage”,IEEE Transaction on Computers,Vol.62, pp.362-375, 2013.

[7] Mohammad Reza Faghaniand and Uyen Trang Nguyen,” A Study of XSS Worm Propagation and Detection Mechanisms in Online Social Networks”, IEEE transactions on information forensics and security, Vol.8, pp.1815-1826, November 2013.

[8] Nuno Antunes and Marco Vieira,” Defending against Web Application Vulnerabilities”, IEEE Computer Society, pp.66-72, February 2012.

[9] Duraisamy.A, Sathiyamoorthy.M and Chandrasekar.S,”A Server Side Solution for Protection of Web Applications from Cross-Site Scripting Attacks”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278 - 3075, Volume-2, Issue-4, March 2013.