Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN : 2348-4470
p-ISSN : 2348-6406

International Journal of Advance Engineering and Research Development

Special Issue on Recent Trends in Data Engineering

Volume 4, Special Issue 5, Dec.-2017

Prevention of Cross Site Scripting (XSS) and securing web application at client side

Shikha Agarwal[1], Akhil Nair[2], Shital Gaikwad[3], Pallavi Chame[4], Swapnil Ethape[5]

Department of Computer Engineering
AISSMS's Institute Of Information Technology, Pune, India

Abstract — *Cross Site Scripting attack (XSS) is a code injection based web security threat in which a website or a web application can be compromised by the attacker to get access over the sensitive information. These kind of attacks are possible when a malicious JavaScript code is injected into the web application using an insecure input or a malicious XSS link. These kind of attacks can be performed on any site which has no input validation and has a poor security implementation over users input. The existing systems fall short for major code injection based attacks. These attacks are increasing day by day as the ratio of vulnerable sites are very high and many a developers don't feel the need for such Security implementations. We presented the design and implementation of a web based API which can be used to protect any website or web application against cross site scripting (XSS) based attack using content security policy (CSP). This API is created in a portable format so that it can be easily implemented on any web based platform and deployed quickly as possible to decrease development time to a greater extent. With the API we also developed an android based administrative control which will help the owner or administrator to efficiently manage security features of his/her website or web application.*

Keywords- *web application security, cross-site scripting vulnerability, content security policy, input validation, application programming interface*

I. INTRODUCTION

All protection based services for securing the web application or a website from cross site scripting based vulnerability is sever side and very hard to implement. And this does not ensure protection from every kind of script those are specially designed for exploiting the vulnerability at the client side. This attacks are targeted and specially designed to interfere with user interface and steal sensitive data. In this paper we presented the design and implementation of a web based API which can be used to protect any website or web application against cross site scripting (XSS) based attack using content security policy(CSP). This API is created in a portable format so that it can be easily implemented on any web based platform and deployed quickly as possible to decrease development time to a greater extent. With the API we also developed an android based administrative control which will help the owner or administrator to efficiently manage security features of his/her website or web application.

All the major security systems available out there lets the data to enter the system and then examines it and then produces the desired output. This is again flawed because as stated above the attacker may enter the content or malicious code in an encoding scheme that only the browser can interpret there by rendering the entire security system useless.

As a result we researched about different security features that can implemented to prevent the malicious content from entering the system itself. That is blocking the entry of such codes or scripts at the client side that is the browser side itself. That is where the CSP (Content Security Policy) gives as a great advantage over the other systems [3].

More over none of the prevention techniques notify the user about these attacks in real time. Here we have implements a model where all the admins will have a special developers account linked to this sites which will be active on their android devices. Upon occurrence of such an attack all the admins will be notified about the attack with the following details.

a. IP from which the attack originated
b. CSP violation report
c. Information of the account active at that
d. Agent information
e. Information of the service provider
f. Location info if available

The CSP (Content Security Policy) implementation is simple but efficient. It is a simple frame work where different rules can be specified to secure our web application / website. In our system we have implement a security feature using this CSP framework were a nonce value or a SHA-256 key is generated for every request. All scripts matching this values or keys are allowed to execute at the client side. Any dynamically added code which does not match this key or value will not be allowed to enter the system and cannot be executed on the system.

II. MOTIVATION

The non-profit international organization Open Web Application Security Project (OWASP) published the data regarding web security breaches in 2016. This data had some alarming information about the kind of attacks which were being performed.
According to their report 1 out of every 8 websites were vulnerable to a code injection based attack i.e. an XSS based attack and according to the adjacent graph almost 43% attacks out of the total where code injection (XSS) based.

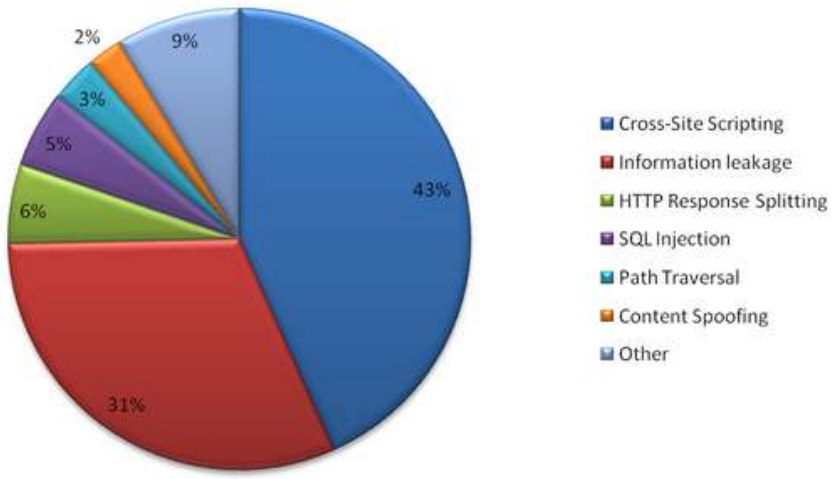

Fig: OWASP Attacks Graph

This gave us the motivation to create a working model of a web security API for prevention of such attacks at the client side itself and make the web applications or web sites completely secure from these kind of attacks.

III. LITERATURE SURVEY

Mukesh Gupta, et al [1] proposed an identification and prediction scheme for Cross Site Scripting based attacks. Also the classification of XSS attacks have be briefly explained. Cross Site Scripting is a security bug that can affect web applications. This bug allows an attacker to inject their own malicious code into HTML pages that are displayed to the users. On successful execution of the malicious code, the system or website action or behavior can be completely changed. It also can steal user's private data or can be performed on behalf of the user and specifically speaking one of the most application layer web attacks, it targets scripts which are dynamically included in a page which executes on the client-side rather than executing on the server-side.
Ankit Shrivastava, et al [2] proposed assessment and prevention mechanism's for prevention of XSS in web applications. This papers has additional information about the things to consider while creating a secure web application. XSS is a threat that occurs because of security flaws of client side scripting languages like JavaScript and HTML. The model of XSS is to handle client-side scripts of a web-app to execute in the order preferred by the malicious manipulator. These kinds of manipulations can embed a script in a page that could be executed each and every time when it is loaded, or whenever an associated event is executed.
Samer MhanaThe, et al [3] proposed a system for generating dynamic Content Security Policy (CSP) at run time to avoid Cross Site Scripting vulnerabilities. Implementation of security is based on CSP (Content Security Policy) which implements Nonce values and this value is uniquely generated for each request within the server and is included with the response to the server. Also the concept of SHA- 256 hashing algorithm is used for static key generation.
Punam Thopate, et al [4] proposed a dynamic detection system for prevention of Cross Site Scripting (XSS). Additionally this paper consist of all the information regarding various different kinds of XSS attacks and also

technical information related to the already existing system to prevent XSS. Automated detection mechanisms or tools are preferred to prevent cross site scripting, such as web security scanners. Detection rate of certain kind of cross site scripts are very low as compared to others. In general, scanners are unable to detect code injection attacks which are based on persistent code which resides in the server itself. Class of Scripting is injected into dynamic pages of trusted sites foe transferring sensitive data of third party. And it avoids same origin policy or cookie protection mechanisms to allow attackers to access confidential data.
Mohit Dayal, et al [5] proposed a very interesting approach for inspection of scripts to identify XSS attacks. Cross Site Scripting attack (XSS) is the computer security threat which allows the attacker to get access over the sensitive information, when the JavaScript, VBScript, ActiveX, Flash or HTML which is embedded in the malicious XSS link gets executed. Here they have discussed about various impacts of XSS, types of XSS, checked whether the site is vulnerable towards the XSS or not.

IV. SYSTEM DESIGN

The system consist of the following modules:

4.1. XSS Security API.

This module is the primary component of the entire project. This module itself manages the security implementations like generation of nonce value and SHA-256 keys and appending it to the response header. It is also responsible for generating a CSP header for every request dynamically and generating new secret keys for every single request. Upon any violation of the rules specified in CSP this module is responsible for notifying all the admins to real time notifications and emails. This module is further divided into sub modules.

- **4.1.1** Key generation module
- **4.1.2** Mobile Application Interfacing Module
- **4.1.3** Access Control Module

4.2. Client Web Application.

This module is the actual web application on top of which the other modules have been implemented to protect this module. In our case this module is the Paper Checking Management System we developed. Any application which implements this web security API can be termed under this module.

4.3. Administrative Module.

Administrative module is another important module in our projects. This module helps the administrator to remotely manage the security API and the web application itself. This modules allows the admins to block or unblock certain IP addresses. Also allows the admins to communicate with each other. It also helps the admins in managing the web application by changing the current state of the web application. This module consist of multiple sub modules for various tasks.

- **4.3.1** Access Modifier
- **4.3.2** Attack Report Manager

V. PROPOSED SYSTEM & ARCHITECTURE

Generally the attack occurs at client side and the data migrates to the server and then the data is validated and inspected for any type of malicious content. This system is completed flawed as data can be in any encoding scheme which the security systems fail to identify or verify [1]. Thus the code migrates and resides in the data base. Upon next request this code again migrates to other client computers and thus making this attack even more devastating.
As the above diagram illustrates all the defense system is client side based in our API implementation. As stated above the API we implemented generates a new nonce value for every new request and assigns this key to all the script included on that page. Now the implementation of CSP helps us to define rules that only the scripts with this values will be executed once [3].
As soon a script or malicious content is entered by the user and it does not match the nonce value provided by the server the code is rejected at the client side itself. A report is send back to server to notify all the administrators and to take adequate action against this attack.
Another implementation with works with CSP is that to implement a SHA-256 value which is generated by evaluating the script that is to be included on the page [3]. Any script that does not match this values is rejected at client side itself.

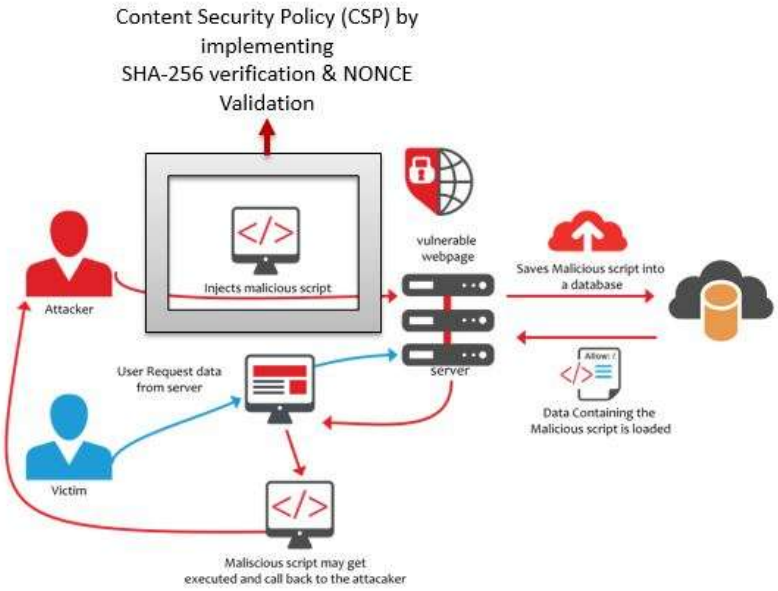

Table 1. System Architecture

VI. EXPECTED OUTCOME

Any unwanted XSS script should be denied access and should not be allowed to enter the system and execute. On attempts to attack the web application / web site notification should be send to all the admins or owners of the site through real time notification to their android devices and email attack report.

VII. CONCLUSION & FUTURE WORK

Cross Site Scripting is a security bug that can affect web applications. This bug allows an attacker to inject their own malicious code into HTML pages that are displayed to the users. On successful execution of the malicious code, the system or website action or behavior can be completely changed. It also can steal user's private data or can be performed on behalf of the user and specifically speaking one of the most application layer web attacks, it targets scripts which are dynamically included in a page which executes on the client-side rather than executing on the server-side. After examining all these prevention and detection mechanisms, we came to the conclusion that designing a security API which will work very fine in validating and removing stripping the legitimate cross-site scripts (XSS), XSS worms and virus as well. The implementation of security is based on CSP (Content Security Policy) which implements Nonce values and this value is uniquely generated for each request within the server and is included with the response to the server. Also the concept of SHA-256 hashing algorithm is used for static key generation.

REFERENCES

[1] Mukesh Gupta, Mahesh Govil, Girdhari Singh. "Predicting Cross-Site Scripting (XSS) Security Vulnerabilities in Web Applications", Malviya National Institute of Technology, IJCSSE, 2015

[2] Ankit Shrivastava, Santosh Choudhary,Ashish Kumar. "XSS Vulnerability Assessment and Prevention in Web Application", Manipal University Jaipur, INGCT, 2016

[3] Samer Mhana, Jamilah Din, Rodziah Atan. "Automatic Generation of Content Security Policy to Mitigate Cross Site Scripting", Universiti Putra Malaysia Serdang, ICSITech, 2016

[4] Punam Thopate, Purva Bamm, Snehal Kunjir. "Cross Site Scripting Attack Detection & Prevention System", IJARCET, Vol 3 Issue 11, 2014

[5] Mohit Dayal, Nanhay Singh Ambedkar. "A Comprehensive Inspection Of Cross Site Scripting Attack", Institute of Advanced Communication Technologies and Research, New Delhi, ICCCA, 2016

[6] Mahmoud Mohammadi, Bill Chu, Emerson Murphy-Hill. "Automatic Web Security Unit Testing: XSS Vulnerability Detection, NC State University Raleigh", IEEE/ACM, 2016

[7] Vikas K. Malviya, Saket Saurav, Atul Gupta. "On Security Issues in Web Applications through Cross Site Scripting (XSS)", Computer Science & Engineering PDPM IIITDM Jabalpur, APSEC, 2013

[8] M. Ridwan Zalbina, Deris Stiawan, Ahmad Heryanto. "Payload Recognition and Detection of Cross Site Scripting Attack", College of Computer Science & IT Albaha University, IEEE, 2016

[9] Imran Yusof, Al-Sakib Khan Pathan, "Preventing Persistent Cross-Site Scripting (XSS) Attack by Applying Pattern Filtering Approach", International Islamic University Malaysia, IEEE, 2013

[10] Rahul Johari, Pankaj Sharma. "A Survey On Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection", Ministry of Communications and IT Govt. of India, ICCSNT, 2012

[11] Guowei Dong, YanZhang, Xin Wang, Peng Wang. "Detecting Cross Site Scripting Vulnerabilities Introduced by HTML5", Renmin University of China, IJCSSE, 2014

[12] Hiroya Takahashi, Kenji Yasunaga, Masahiro Mambo. "Preventing Abuse of Cookies Stolen by XSS", Kanazawa University Japan, AJCIS, 2013

[13] Prof. Piyush A. Sonewar, Prof. Sonali D. Thosar. "Detection of SQL Injection and XSS Attacks in Three Tier Web Applications", Prawara Rural Education Society, Loni, IJARCET, 2015

