# Implementation of Preventing CSRF and XSS Security Attack by Generating Multiple Tokens for a Session and Filtering Special Characters by K-BAG Filter

**D.Kavitha[1] M.R.Akshaya[2] M.Karthick[3] K.Baghya[4] K.Gomathi Raja Eswari[5]**
[1]Assistant Professor [2,3,4,5]UG Student
[1,2,3,4,5]Department of computer science and Engineering
[1,2,3,4,5]Valliammai Engineering College

*Abstract*— Cross Site Request Forgery is a security attack force the user to perform state changing request like fund transferring, getting password. It inherits the user credentials and privileges of the victim to perform an undesired function on behalf of the victim and this attack focus on the state of the session. The CSRF attack is prevented by generating unique encrypted token for each state in a session and the token is of 8-bit. The token encrypted with MD5 hashing algorithm, In order to secure the token i.e., the token becomes 128 bit value. The session is not been compromised unless the MAC address matches. The XSS attack is prevented by filtering the special characters using K-BAG filter. The malicious functions are removed by means of pattern matching

***Key words:*** CSRF attack, XSS attack, K-BAG filter, Token generation, session ID

## I. INTRODUCTION

Now-a-days it is rare to see the webpage without having any attacks. According to the OWSAP (open web application security project), 87% of the web page [2] consists of CSRF (Cross site Request Forgery) attack. CSRF is a kind of attack in which the attacker tries to capture the request packet send from the user. CSRF attack is also known as hostile linking, session riding [14], and one click attack [12]. Another common attack in the webpage is XSS [1],[10], [15] (Cross Site Scripting) attack. XSS is a kind of attack in which the attacker tries to inject the malicious script in the text box of the webpage where webpage is vulnerable [3],[8].CSRF attack is server side attack. In CSRF attack, the role of the attacker is the man-in the middle. When the user sends the web request to the server, the attacker will capture the request and get the important information about the user. Many techniques are available to prevent this attack. Nenad jovanovic, et.al introduced server side proxy to prevent this attack. In that system there is one proxy server, which is used to detect the CSRF attack. These detection and prevention is transparent to the user and web application itself. The main advantage of the system is that requires only minimal manual effort. The limitation in this system is that generates some false XSRF alarms. Yin-chang sung,et.al introduced content box to prevent standard and multi stage CSRF attacks. The content box is used to differentiate between the trusted site and untrusted site. By this we can find which webpage was affected by the CSRF attack. The main advantage of this system is that it have some builtin method and built in properties to reduce the computation overhead.Bayan Chen, et.al introduced CSRF guard to prevent the CSRF attack. CSRF guard is a token or key to differentiate between the legitimate request and forged request. In order to prevent the CSRF attack, they inject the token.

If the attackers guess the pattern of the token then attacker will change the content in that request because the browser will inject only one token as CSRF guard. In our proposed system, we are going to prevent the CSRF attack by generating the token for each and every step in a session. The tokens generated by the server are unique and random. In addition to the token generation, we also encrypt the token generated by the server. By encrypting the token, the attacker will not able to guess the pattern of the token. Additionally we introduced a new feature in the server to verify the MAC address of the user.

XSS attack is client side attack. It is similar to SQL injection [3],[10] attack. There are two type of XSS attack are available. They are stored (persistent) and reflected (non-persistent). Reflected XSS attack means the attack will be performed until that webpage get refreshed. Stored XSS attack means the attack will be performed every time even if the webpage get refreshed. Anastasio Stasinopoulos, et.al introduced XSS auditor [1]. XSS auditor is a filter used to remove the script in the input. XSS auditor doesn't use regular expression to filter. Instead they examine the DOM tree created by the HTML parser. The main advantages of this XSS auditor is to identify whether the input contains the script or not. In our proposed system, we are going to introduce a filter called K-BAG filter. This K-BAG filter is used to remove the tags and special characters. If the attacker tries to inject the malicious script in the textbox then K-BAG filter will remove all the tags and special characters. Pattern matching is used to check whether the input given in the textbox contains any executable function or not.

## II. OVERALL SYSTEM ARCHITECTURE

### A. *CSRF*

Figure 1 describes that the user will the web request to access the server through the browser. Then the server will generate token and response. The generated token is sent to encryption algorithm to generate the encrypted token. Then the server will send the encrypted token and response to the user to continue the application.

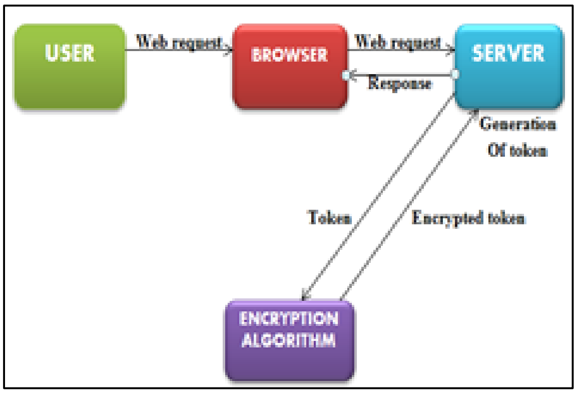


Fig. 1: Overall architecture diagram for CSRF

### B. *XSS*

Figure 2 describes that the input entered by the user will send to HTML tag filter in order to filter the HTML tag. The output from the HTML tag filter is send to K-BAG filter which is used to filter the special character. The output from the K-BAG filter is send to pattern matching in order to check whether the given input is in correct pattern or not. If the input is correct pattern then input will store in database. If the input is not correct pattern then the server will block the access to continue the application.

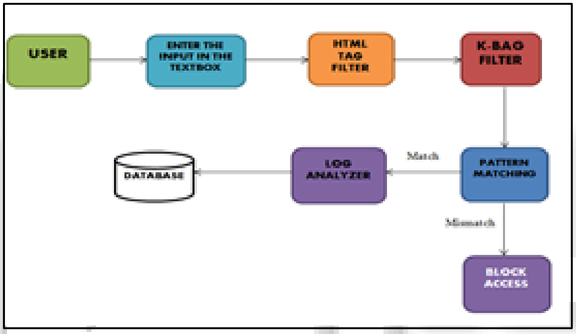


Fig. 2:

## III. System Description and Design

### A. *Web request to access server*

Initial step is to send the web request to access the server. CSRF is a server side attack so we are going use PHP to create the webpage. PHP is a server scripting language. PHP code is mixed with HTML code. PHP code is processed by the PHP interpreter which is implemented as Common Gateway Interface (CGI) executable. The general syntax for writing PHP code is

```
<?php
………
?>
```

This PHP code is embedded with HTML code. HTML is a markup language used to create the webpage. HTML was defined by Tim Berners-Lee in 1990 when he was working at a European high-energy physics research center (CERN). HTML use HTTP protocol in order to provide the communication between the server and client. Hypertext Transport Protocol (HTTP) is a form of communication protocol which provides the detail about the communication between the server and client. HTTP protocol is also known as request-response protocol. If the users want to access the server then users should the web request to the server through the browser. This web request is in the form of HTTP request message. Every HTTP request message consists of some basic elements like start line, header field, and followed by the message body which is optional. Every start line consists of three parts. They are request method, request -URI portion of web address and HTTP version. Example for start line is GET / HTTP/1.1

This GET method will request the server to return the resource specified in the request -URI portion of web address. After sending web request to the server, the server will verify that whether the user has rights to use the webpage or not. If the user has no rights to access that webpage means then the server will send error message to the user. If the user has rights to access the webpage means then the server will send the response message to the user. This response message is in the form of HTTP response [9] message. The HTTP response message consists of a status line, header field and message body which is optional. The status line has three fields. They are version used by the server software, a numeric status code to indicate the type of the response and a text string to present the information in the numeric status code in human-readable form. Example for status line is HTTP/1.1 200 OK

Here 200 is a numeric status code. All status codes are three digit decimal numbers. The first digit in the status code represents the general class of the status code. That is 100 series is used to provide the information to client before the request processing has been completed. 200 series is used to represent that request has been successfully processed. 300 series is used to represent that client needs a different resource to fulfill the request. 400 series is used to represent the client error. 500 series is used to represent the server error. Therefore web request should send from the user in order to access the server.

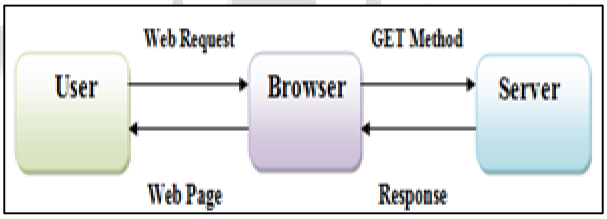


Fig. 3: Web request to access server

Figure 3 describes that user gives the request for the web page in browser by entering its domain name. Browser forwards the request to server in the form of GET method. Server sends the respective response packets to the browser.

### B. *Token generation and response from the server:*

CSRF attack is performed by capturing the web request from the user. To prevent the CSRF attack, we are going to generate the token [11]. Token is a 8-bit value generated by the server to identify the authorized user. The web request from the user will send to the server. The server will check whether the user has rights to access the webpage or not. If the user has rights to access the webpage then the server will generate 8-bit token. Then this token is encrypted using an encryption algorithm. This encrypted token is send to hashing algorithm in order to generate the unique hash value. We are going to use MD5hashing algorithm [6], [7] to generate hash value. Message digest is used to generate the message authentication code (hash value). The message integrity is measured by MD5. MD5 consist of hash function that deals with security features. The output generated by MD5 algorithm is always 128 bit. It will not bother about the number of bits in the input. Therefore the attack required 2128 bit operation to see original message. Generally MD5 algorithm is less secure than SHA (Secure Hash Algorithm) because SHA produce output of 160 bit and it required 2160 bit operation to see original message. The main advantage of MD5 algorithm is faster than SHA because MD5 requires sixty four iterations to generate the hash value whereas SHA requires eighty iterations to generate the hash value. After generating the token, the response and token from server is sending to the user through browser. Then the user will send the next web request to the server along with that the token. Then the server will check whether the token sent by the user is correct or not. If the token is correct then the server will send the response and another fresh token to the user. If the token is not correct then the server will block the access.Generation of token is done for each and every step in a session. Therefore it is difficult for attacker to identify which token is generated for which step. Additionally the token is in encrypted format so he may not see the original token.

In PHP, MD5 hash value is generated by function called md5 (string, raw). The parameter string is used to specify the string for which the hash value has to be generated and the parameter raw is used to specify the output either in a hex or binary format.

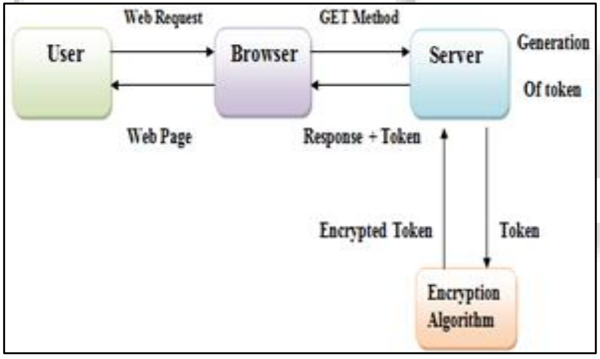
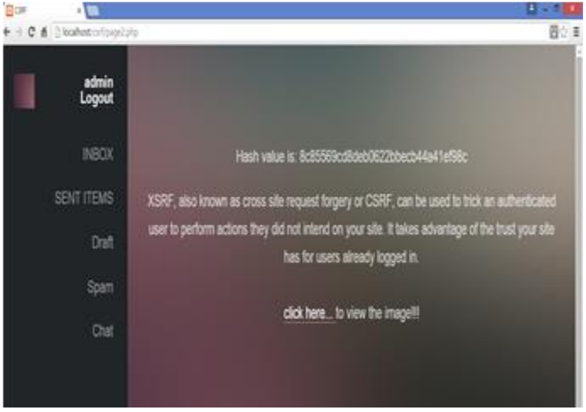


Fig. 4: Generation of token

Figure 4describes that user sends web request to the server. At once, the server receives the request packet it generates the token. Then the token is being encrypted with an encryption algorithm along with the user text. Finally the encrypted token is send to the user. For each session new token will be generated.

1) *Implementation of preventing CSRF attack*



Hash value is: 8c85569cd8deb0622bbecb44a41ef98c

XSRF, also known as cross site request forgery or CSRF, can be used to trick an authenticated user to perform actions they did not intend on your site. It takes advantage of the trust your site has for users already logged in.

click here... to view the image!!!

Fig. 5:

C. *Mac address validation in the server*

After sending the response and token to the user, the user will continue the application by sending the next request and token to the server. If the attacker analyzes the pattern of token for long time then the attacker may guess the pattern of the token. By the guessing the pattern of the token, it is easy for the attacker to decrypt the token. Then the attacker will do whatever he wants. To avoid this kind of attack, we are going to add the additional feature to the server. The new feature in the server is to check the MAC address of the user. MAC address is unique identifier assigned to network interfaces for communications. MAC address is also known as Ethernet Hardware address (EHA), hardware address or physical address. The server will have the MAC address of the user. MAC address cannot be changed. When the user sends the web request to the server, the server will check the MAC address of the user. If the MAC address mismatches then the server will not provide the access to the server. If the MAC address matches then the server will provide the access to the server.

Fig. 5: MAC address validation

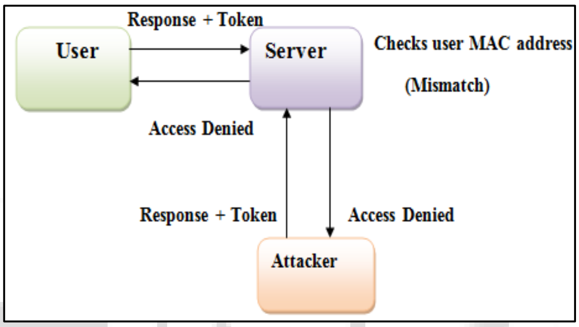
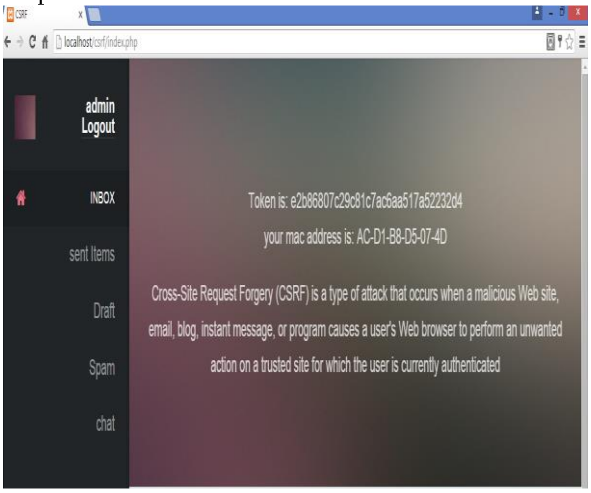


Figure 5 describes that when the user is send the web request to the server, the server will verify the MAC address of the user to order to find the authorized user. If the MAC address is mismatched then the server will block the access.

1. Implementation of mac address validation:



Token is: e2b86807c29c81c7ac6aa517a52232d4

your mac address is: AC-D1-B8-D5-07-4D

Cross-Site Request Forgery (CSRF) is a type of attack that occurs when a malicious Web site, email, blog, instant message, or program causes a user's Web browser to perform an unwanted action on a trusted site for which the user is currently authenticated

D. Filtering of special character using k-bag filter:

XSS is a client side attack. XSS is occurred due to badly written PHP [4], [5]. This attack is performed by injecting the malicious script in the textbox of the webpage. To prevent this kind of attack, we are going to filter [13] the tags and special character. The input entered by the user is send to the HTML tag filter. HTML tag filter is used to filter the tags in the input given by the user. For example if the input given by the user contains the <script> tags means then the HTML tag filter will remove the <script> tag. After removing the HTML tag, the output from the HTML tag filter is send to the K-BAG filter. K-BAG filter is used to remove the special character in the input given by the user. For example if the input given by the user contains the &lt script &gt then the K-BAG filter is used to remove the special character like &. K-BAG filter is used to remove not only the special character like & but also the special character like @, <,>. The output from the K-BAG filter is free from the tags and special character.The K-BAG filter is also used to remove some functions like alert and functions to get the cookies.

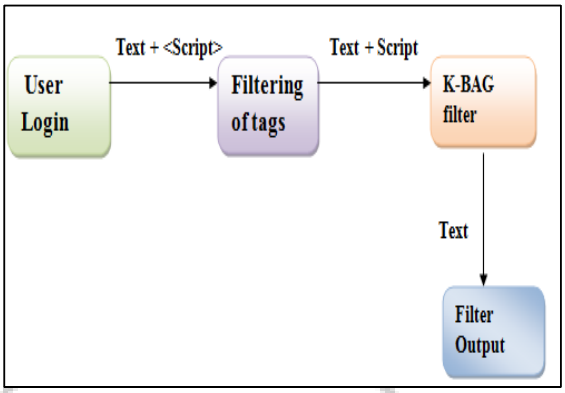
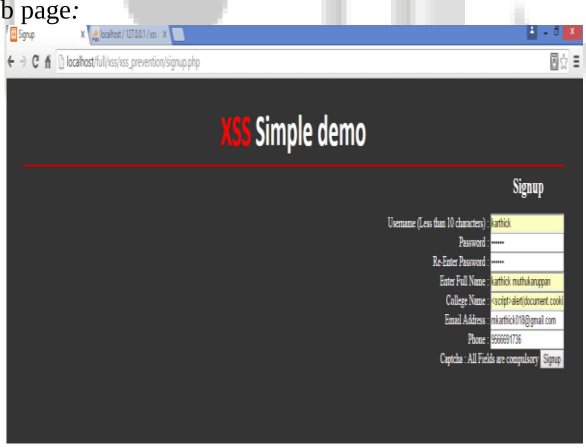


Fig. 6: Filtering of tags and special character

Figure 6 describes that in user login, the unauthorized user enters the text along with the malicious script tags. The tags are removed in the filtering phase. The special characters are removed in the K-BAG filter.

1) Implementation of insrting script tag in text box of a web page:

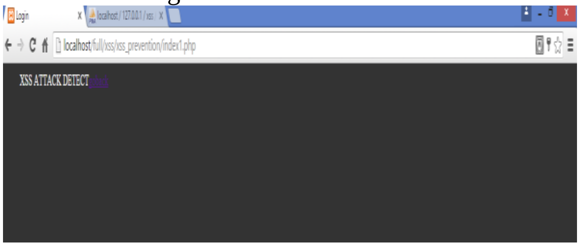

Signup
localhost / 127.0.0.1 / xss
localhost/full/xss/xss_prevention/signup.php

XSS Simple demo

Signup

Username (Less than 10 characters) : karthick
Password : ••••••
Re-Enter Password : ••••••
Enter Full Name : karthick muthukaruppan
College Name : <script>alert(document.cooki
Email Address : mkarthick018@gmail.com
Phone : 9566691736
Captcha : All Fields are compulsory Signup

Fig. 7: Detected xss attack:


Login
localhost / 127.0.0.1 / xss
localhost/full/xss/xss_prevention/index1.php

XSS ATTACK DETECTgoback

E. Verification through pattern matching:

After removing the tags and special character from the input given by the user. It is necessary to check the input is in correct pattern. It is verified by using pattern matching. Therefore the pattern matching is used to check whether the input given by the user is in correct pattern or not. Pattern matching is often described by the regular expression. If the pattern is mismatched then the server will block the user and the user may not continue the application. If the pattern is matched then the server will give the response message in the log analyzer. After that the input given by the user is stored in the database.

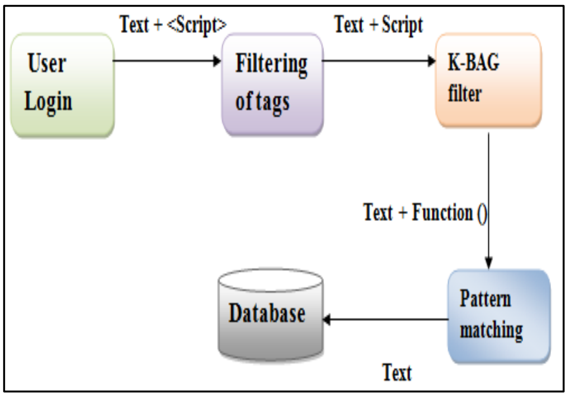


Fig. 7: verification through pattern matching

Figure 7describes that the texts from K-BAG filter may contain executable function. The pattern matching is used to filer the function in the input text. If text matches access will be denied else result is stored in the database.

## IV. Conclusion

The prevention of CSRF attack can be implemented by means of generating unique tokens to each state in a session in order to secure the token the MD5 hash algorithm is used to encrypt the token with its hash value results to form a 128 bit encrypted token so it is not compromised when a hacker hacks the session and in addition to that the Mac address is not matches and it is not validated. The XSS attack is prevented by means of the proposed K-BAG filter whish filters the special characters and it prevents in injection of malicious code inside the script tag. Pattern matching removes the malicious executable function by means of matching the input text with existing default functions in the log.

## References

[1] Bypassing XSS Auditor: taking advantage of badly written PHP code by Anastasios Stasinopoulos in 2014 IEEE transaction.
[2] Protecting websites from attack with secure delivery networks by David Gillman in 2015.
[3] Analysis of field date on web security vulnerabilities by jose fonseca in IEEE transaction on dependable and secure computing vol 11 NO:2 March/April 2014.
[4] Defending against cross site scripting attacks by Lwin Khin Shar in march 2012 IEEE.
[5] Security vulnerabilities in the same- origin policy: Implications and alternatives by Hossein Saiedian in September 2011 IEEE.
[6] Design and Performance Analysis of a Unified, Reconfigurable HMAC –Hash Unit by Esam Khan in IEEE vol.54,No.12,DECEMBER 2007.
[7] Research for the application and safety of MD5 algorithm in password authentication by X.Zheng in 9th International Conference on Fuzzy system 2012.

[8] Threat modeling for CSRF attacks by Xiaoli Lin in 2009 IEEE.

[9] An HTTP Extension for secure transfer of confidential data by Masaru Takesue in 2009 IEEE.

[10] New threats and attacks on the world wide web by Thorsten Holz in 2006 IEEE

[11] Light-weight CSRF protection by labeling user-created contents by Yin- chung sung, in 2013 7th International conference on software security and reliability.

[12] A study of effectiveness of CSRF Guard by Boyan Chen in 2013 IEEE.

[13] A study of XSS worm propagation and detection mechanisms in online social networks by mohammad reza faghani in November 2013 IEEE.

[14] Preventing cross site request forgery attacks by Nenad Jovanovic in 2006 IEEE.

[15] Automating Isolation and Least Privilege in Web Services by Aaron Blankstein in 2004 IEEE.