RESEARCH ARTICLE OPEN ACCESS

Click jacking Vulnerability Analysis and Providing Security against WEB Attacks Using White listing URL analyzer

D.Kavitha
Department of Computer Science and Engineering
Valliammai Engineering College
dkavitha2005@gmail.com

S.Ravikumar
Department of Information Technology
Valliammai Engineering College
vijayravikumaran.rs@gmail.com

----------**********----------

Abstract:

In recent years publicly reported vulnerability have strong growth in web applications. These security vulnerabilities continue to infect the web applications can cause vast security problems. Organizations are taking their businesses online, attacker can steal confidential data of the organization with these attacks resulting loss of market value of the organization. Securing the websites against these vulnerabilities is very difficult and challenging task as day to day new techniques for attacks are invented, so the study of various types of vulnerabilities, detecting the attacks and providing solution for these vulnerabilities is essential part in internet world. This paper focuses on clickjacking attack and providing security mechanism as whitelisting URL analyzer to overcome the clickjacking attack. The security mechanism has the features and standards to measure the attack and how much the vulnerability is being exposed with respect to the context of application in the social networking environment. The Proposed iframe tag checking algorithm and DNSlookup checking algorithm is based on regex. Regex handles both the internal and external fault efficiently and reduces the load time of iframe tag checking and DNSlookup using simple patterns of regex. Thus the proposed algorithm overcomes clickjacking attack efficiently than existing defenses. The vulnerability of the attack can be measured by the deviation of the system state with expected state. This deviation can be overcome by the security mechanism.

Key words — Clickjacking, security mechanism, iframe, DNSlookup, URL analyzer

----------**********----------

I. INTRODUCTION

Technology development in web applications coupled with a changing business environment mean that web technology are becoming more prevalent in corporate, public and Government Services. Almost all business relay on websites to deliver content to their customers, interact with customers, and sell products. Web security is possibly today's most overlooked aspect of securing the enterprise and should be priority in any organization. Applications are often riddled with vulnerabilities that are used by attackers to gain user information. There are number of new security attack, which could potentially pose significant risks to an organization's information technology infrastructure [1]. Most of the attacks are targeting security flaws in the design of web applications, such as click jacking. Click jacking is a malicious attack where the attacker hijacks user clicks in order to perform undesired actions which are beneficial for the attacker ssss[2]. It is performed through iframes in which, the user unknowingly clicks on a malicious page that sits on top of user page. In technical term to an invisible iframe is placed above a clickable component on the page and instead of doing the action that was intended, the attackers iframe is executed resulting in a completely different action than the one intended by the user. Many defenses have been suggested for click jacking but they have all been bypassed by malicious users. This project focuses

on providing, an efficient countermeasures for the click jacking security attack. Click jacking attack is also known as UL readdress attack whereas the attacker uses multiple transparent layers to trick the user to click a button or to link to another page with this the user is hijacked while clicking by this the attacker owns the control of the intended user[3]. The server didn't return an X-Frame-Options header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid click jacking attacks, by ensuring that their content is not embedded into other sites. A typical click jacking attack uses two nested iframes to crop and position an element from a target website[4]. The inner iframe contains the target page and must be large enough to display it. The outer iframe is much smaller and acts as a window on to the page loaded in the inner frame. Frame busting refers to the code provided by the web page intended to prevent the web page from being loaded in a sub frame. Frame busting code typically consists of a conditional statement and a counter action that navigates the top page to the correct page [5]. The key solution to prevent clickjacking is to improve web browser functionality to detect and defend against hidden iframe and malicious java script. Web filters can block users from accessing dangerous sites that may contain click jacking techniques. Web application firewall can scrub all content from malicious scripts and deny attackers from injecting click jacking scripts on to your web sites. Application firewall can also validate from parameter inputs to prevent malicious input from being sent to the web servers [6].

II .RELATED WORK

Denial of service is a common attack in the Internet which causes significant problems for both users and service providers[7]. Distributed attack sources can be used to enlarge the attack in case of distributed denial of service. Defending against DoS/DDoS attacks generally involves 3 different phases: prevention, detection and response. Detection is one of the key steps for defending against DoS/DDoSattacks and proper response will be linked to the detection alarm. A good detection technique provides short detection time, low false positive rate, and low computational overhead [8]. This Paper presents a novel technique which detects TCP based flooding attacks by using the TCP congestion window which is analyzed using the cumulative sum Network Simulator NS2 which is used to validate the proposed technique. CUSUM statistic uses a sequential change point detection algorithm to detect TCP based flooding attack and it filter the traffic using different criteria's like address, port and protocol. It requires less computational and memory resources than other change point detection algorithms. The congestion window detection technique is a particularly attractive metric as it is commonly available as part of the TCP/IP stack. TCP protocol cannot be manipulated by the attacker as it directly related to the load in either intermediary network connections or the load at the server. In this paper congestion window will be affected by the packet loss in the network. Using the TCP/IP stack extra computational overhead will be added to the target and the TCP/IP stack will need modification to perform the CUSUM and report the measurement. In this paper buffer-overflow protection: the theory proposes a framework for protecting against buffer overflow attacks the oldest and most pervasive attack technique [9]. The malicious nature of buffer-overflow attacks is the use of external data as addresses or control data. Using this observation, a sufficient condition for preventing buffer overflow attacks is provided and it proves that it can create a secure system with respect to buffer-overflow attacks. The underlying concept is untrustworthy, and should not be used as return addresses and function pointers [10]. If input can be identified, the buffer-overflow attacks can be caught. This framework is used to create an effective, hardware, buffer-overflow prevention tool. It focused on control data and local variables. In the vast majority of attacks, control data is the target so prevention schemes have focused on control data[19]. Control data can be divided into several types as return addresses, function pointers, and branch slots. Return addresses have been the primary target since their location can easily be guessed [20]. The draw

backs are it Focus on the length of a buffer, there is an added cost to checking a static value before each and every procedure completes. Depending on the implementation of the variables and how frequently procedural calls are made, the overhead can range from trivial to significant To maintain its integrity, the created address is locally can be signed when it is created and is validated by associated instructions. But here the signature must not be passed across domains. In this paper hijacking spoofing attack and defense strategy based on internet TCP sessions The rapid development of network applications, network security issues become a priority need to consider a variety of network applications in a variety of spoofing attacks. The focus of network intrusion is to prevent spoofing TCP session hijacking [11]. The attacker acts as a third party to participate in this type of attack that is the injection of additional information on TCP based sessions. Passive hijack of the session attack first observe and record all possibility to send and receive data at the rear. This paper first explains the concept of TCP session hijacking and its principle and the harm caused by it. It helps the attacker to find the defense for session hijacking attack process. Encryption technology can be one of the few ways prevent session hijacking attacks. Session hijacking attacks is very difficult to defend. Therefore, any critical connections are used to transmit sensitive data must be encrypted [13]. Telnet protocol secure SSH (Secure Shell) can make the system against session hijacking attacks. In this paper the principles arising from session hijacking expounds the process of the attack and defense strategies, although currently there is no effective way to prevent or eliminate a fundamental, but can be reduced accordingly to prevent the occurrence of such an attack to some extent. It possible to predict the serial number, as long as the attacker access to the session packet.TCP protocol which also brings security threats cannot be avoided [12]. In this paper survey on botnet: its architecture, detection prevention and mitigation Robot Network or Botnet is the biggest network security threats faced by home users, organizations, and Governments. Botnet appeared two decade ago, but even now the threat caused by it is estimated due to its robustness and dynamic nature, which results in a user blockage by ISP or stop the normal usage of certain applications. Botnet challenges IT Community in detection, prevention and mitigation from Botnet attacks [18]. In this paper analyze and suggest various testbed which gratify these requirements would enable a range of experimental study. The aim of these experiments on new method and tools is to characterize, compare, identify and prevent botnets [17]. It is to analyze the protocols being used by the Supervisor-bots and how they evolved with the passage of time. Analyzed how cyber defenders proposed and work for the detection of a cyber-attack from known and unknown BOTNETs and given ideas and techniques for its prevention and mitigation [14]. Honey pots based defense is so popular and used mostly; it is predicted and possible that one day supervisor-bots will have a defense mechanism for detection of honey pots in their bots. New testbeds are required to be developed which allow testing in large-scale network either open or closed environments. Getting of Botnet sample code is required for analyzing but criminals don't want to examine their malware as well as cyber defender also feels hesitation with un-trusted ones. Supervisor-Bot ability is used for DNSBL lookup before attack [15]. However this has two problems; first having high false positive and second it cannot detect distributed inspection. However, application of all these strategies is the most expensive solution and it is not realistic to expect that all the Internet users will apply them. The attacks can be identified when they exceed thresholds. The thresholds are not dynamically adjusted depending on the part of day, day of the week etc, in accordance with the stored statistics. After the protocol is completely decoded, analysis engine cannot detect strange behavior, i.e. unexpected values in the packages and reject the packages [16]. Follow the traffic in real time and not compared it with the stored statistics. The problem lies in the fact that this issue is not treated in the same way in many countries and that's why appropriate legislative solutions haven't been defined.

III. SYSTEM ARCHITECTURE

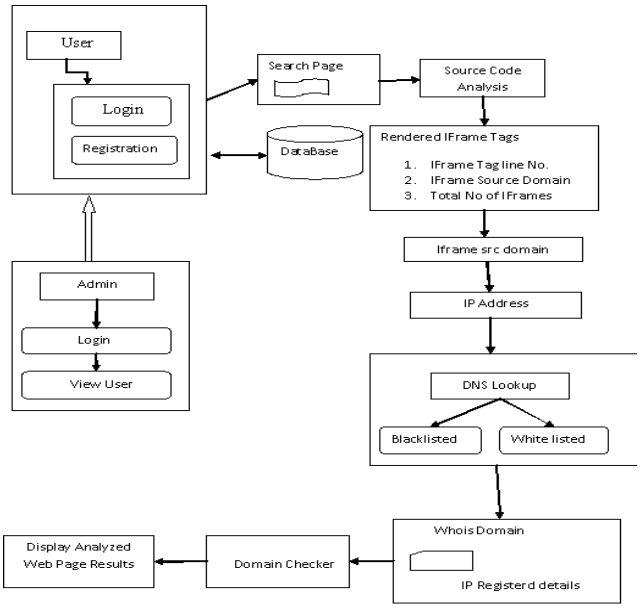

Fig 1: Overall system architecture

Fig 1 describes the overall system architecture whereas the user creates the user login page and the authorized user registration is done. The user authentication is needed to allow the authorized user to enter the corresponding websites. In login or logon is the process by which individual access to a computer system is controlled. User can log in to a system to obtain access and can then log out or log off when the access is no longer needed

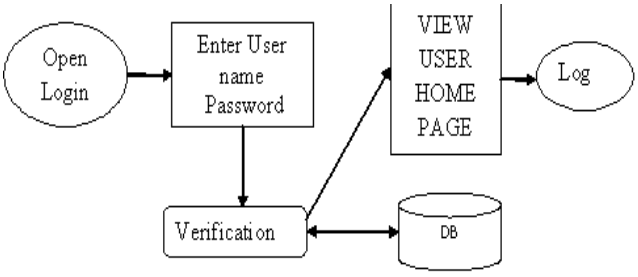

Fig 2: login page

Fig 2 describes the verification of the authorized user in the database. This module can be placed on any Module Tab to allow users to login to the system. It acts as a interface between the user and the search page. The username and password of this module is stored in the database while registration. Whenever the user log in, it automatically checks the database and provide the access to the user. Search module allows users search for specific URL what they intended. And it is designed to search for information on the World Wide Web. It retrieve the information about user entered URL. The actual input URL is placed in this module. When the user clicks the submit button, it retrieve the next module. When the user specified an inappropriate URL it shows the URL is Invalid. The admin module role which has all available permissions for manipulates and views the user actions. Login into admin page it will display the registered user details. In this module the tag checker analyses the source code of corresponding websites and retrieve the iframes in the HTML tags it gets the input URL from the user and analyze the source code for iframes using regex. First it checks the characteristic of input URL using preg match function of regex. Regex is also used to match the input URL with Iframe URL.

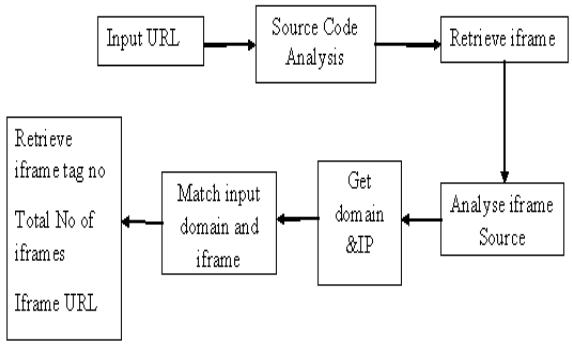

Fig 3: iframe tag checker

Fig 3 describes the iframe tag checker whereas the input URL is given the source code is analyzed in the source code iframe tag is retrieved and it is analyzed with its domain and its IP address with its matching domain. Then it finally displays, iframe contained tag line number, number of iframes in the tag iframes contained source URL. In this module retrieve the iframe source domain from iframe tag checker and finds the IP address retrieved domain and it analyzed whether this module is listed in DNS lookup. If this IP address is not listed in DNS lookup it shows the results as IP is in Whitelisted .If the input URL is a Listed in DNS Lookup, then the results shows the websites is vulnerable and its blocked by the administrator. Usually domain stored in dnslookup as an array. Reverse IP function is used to reverse the IP of input URL.

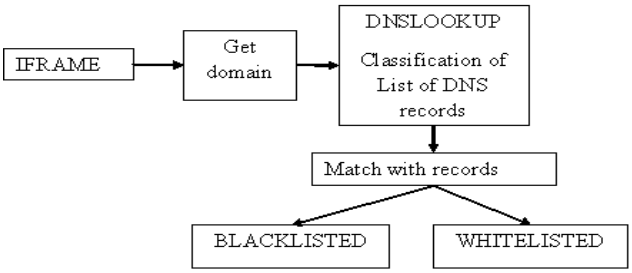

Fig 4:DNSLOOKUP record matched with whitelisted

Fig 4 describes that this Module get the input from IP of DNS Lookup domain and retrieve the IP registered user details and it displays, Registry Domain ID ,Registrar URL, Registrar Creation and Updated Date, Registrar Organization and Registrar Address Details like Street, Country ,Fax and Contact details. It will display whether the domain is vulnerable or not based on the analysis of previous modules it shows the results as website is secured or vulnerable or may be secured.

IV ALGORITHM FOR IFRAME TAG CHECKER USING REGEX

This is the algorithm for iframe tag checker using regex and the algorithm as follows:

Input: Input URL
Output: Retrieve the content of iframe
Step 1: Get the input URL from User
Step 2: Analyze the URL Pattern of input URL using regex
Step 2.1: If the regex of URL matched with the input URL
Step 2.1.1: Get the domain from the matched pattern of regex
If the match is not found return invalid URL. **Step 3**: Get the source file of input URL
Step 4: Get the domain of input URL
//Search source code of input URL for iframes
Step 5: Match regex of iframe tag with the source file
Step 5.1: Initialize frame count is equal to zero
If Match found
Step 5.2: Extract the iframesrc using regex
Step 6: Get the domain of iframesrc
Step 7: Compare the domain of iframe with input domain
If not matches
Step 7.1: Increment the frame count by one
//get the content of iframe
Step 7.2: Get line number of iframe.
Step 7.3: Get iframe domain.
Step 7.4: Get total iframe count.
If **Step 5** is false
Return no iframes

V ALGORITHM FOR WHITELISTED AND BLAKLISTED USING DNSLOOKUP

This is the algorithm for separating the blacklisted and whitelisted using DNSLOOKUP and the algorithm as follows:

Input: iframe domain
Output: Whitelisted or Blacklisted
Dnslookup=>Predefined DNSlookup contained array (“”, ””)
Type A=>”Not secured”
Step 1: Get IP of iframe domain
Step 2: For each IP Perform reverse IP
Step 2.1: Explode of IP using regex
Step 3: Reverse of Explode using regex
Step 4: Implode of Reverse using regex
Step 5: Check the DNS records of DNSLOOKUP with the
Output of Step2 If found
Step 6: Matched with “Type A” records of DNSLOOKUP
If match found IP is in Blacklisted Display vulnerable
Else
IP is in whitelisted

VI EXPERIMENTAL RESULTS

Fig 5 shows the experimental result of IP checker for vulnerable URL, Fig 6 shows the experimental results of whos domain for vulnerable URL and the Fig 7 represents the experimental results of domain checker for vulnerable URL.

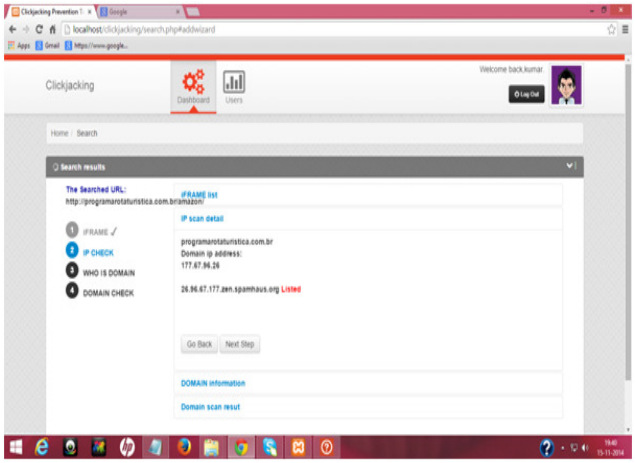

Fig 5: IP Checker for Vulnerable URL

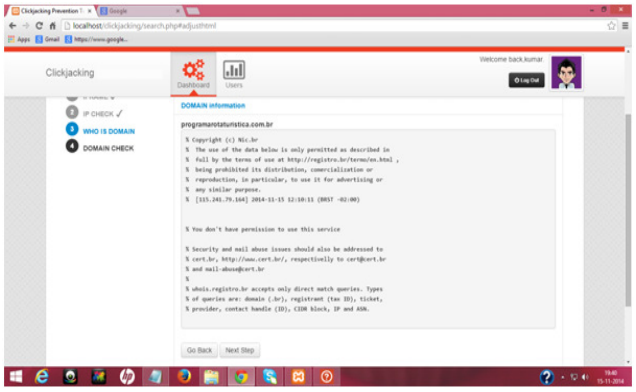

Fig 6:Whos domain for Vulnerable URL

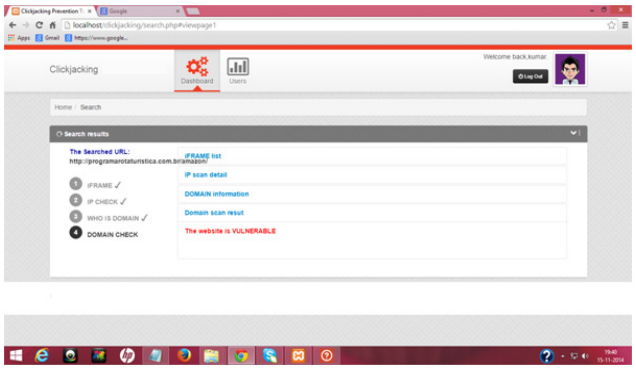

Fig 7: Domain Checker for Vulnerable URL

VI CONCLUSION AND FUTURE WORK

The Analysis of clickjacking security attack has been done by means of using whitelisting URL analyzer security mechanism. The vulnerability of clickjacking and the deviation of the system is, from expected to the actual level of security. The security mechanism incorporates the features to measure the deviation of the system. The proposed algorithm of iframe tag checker using regex focuses on analyzing and retrieving the iframe and the Dnslookup algorithm focuses on analyzing whitelisting and blacklisting. These algorithm performed using regex will reduce the time spent on coding, reduces effort, increase the processing speed and it provides efficient solution for clickjacking. Thus the algorithm ensures the security of clickjacking and its vulnerability using whitelisting URL analyzer. Reasearchers needed to overcome the future web attacks Using this paper protection for any organization from clickjacking is possible but when the organization is surrounds with new attack, our solution is not enough, Preventing more than one new attack, updates will be needed for regular maintenance. It will lead to new security mechanism.

REFERENCE

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2]J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[5] M. Young, The Technical Writer's Handbook. Mill Valley, CA:University Science, 1989.

[6] Marcus Niemietz, Authentication Web Pages with Selenium: Vulnerability Analysis and Exploita- tion of Authentication Web Pages with Selenium . AVM, 2010, p. 35.

[7] Himanshu Dwivedi, Chris Clark, David Thiel Professional Penetration Testing: Creating and Operating a Formal Hacking Lab
. Syngress Media, 2009, p. 353.
[8] Paul Stone, 2010. Next Generation Clickjacking, White Paper . Context Information Security Ltd.
[9] Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, ChristoperKruegel, 2010. A Solution for the Automated Detection of Clickjacking Attacks. ASIACCS.
[10] A. Moshchuk, T. Bragin, S. D. Gribble, and H. M.Levy. A crawler-based study of spyware in the web. In Proceedings of the Network and Distributed System Security Symposium, NDSS 2006, San Diego,California, USA, 2006.
[11] D. Balzarotti, M. Cova, V. Felmetsger, N. Jovanovic,E. Kirda, C. Kruegel, and G. Vigna. Saner:Composing static and dynamic analysis to validate sanitization in web applications. In IEEE Symposium on Security and Privacy , pages 387–401, 2008.
[12] Y. Xie and A. Aiken. Static detection of security vulnerabilities in scripting languages. In USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium , Berkeley, CA, USA, 2006. USENIX Association
[13] Marco Balduzzi, Manuel Egele, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. A solution for the automated de- tection of clickjacking attacks. In ASI- ACCS'10 , 2010.
[14] Adam Barth, Collin Jackson, and John C.Mitchell. Robust defenses for cross-site re- quest forgery. In In proc. of 15th ACM Con- ference on Computer and Communications Security (CCS 2008) , 2008.
[15] Adam Barth, Collin Jackson, and John C. Mitchell. Securing frame communication in browsers. Communications of the ACM
(CACM 2009) , 2009.
[16] Daniel Bates, Adam Barth, and Collin Jackson. Regular expressions considered harm-ful in client-side xss lters. In Proceedings of the 19th International World Wide Web Conference (WWW 2010) , 2010.
[17] D.Kavitha, Rani S.K, Review of Botnet Attacks and its detection Mechanism , International Journal of Innovative Research in computer and Communication Engineering volume 3, pp 2377-2383, March 2015 .
[18] D.Kavitha,S.Ravikumar, A Survey of different software Security attacks and risk analysis based on security threats, International Journal of Innovative Research in Computer and Communication Engineering, volume 3, pp 3452-3458, April 2015.
[19] Rani S.K, D.Kavitha , Comparative study of major phishing targets , International Journal of Applied Engineering Research , volume 10, pp 122699-12706, March 2015.
[20] Rani S.K, D.Kavitha , A study of major phishing targets and their anti-phishing solutions , International Journal of Technical Research and Applications , volume 3,pp 45-51, March 2015.