© 2016 IJSRST | Volume 2 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Analysing and Detection of Clickjacking Attack

K.Gokul, P. J. Gowtham, S. Jaffar Ahamed, G. Abirami

Department of Information Technology, Dhanalakshmi College of Engineering, Chennai, Tamiladu, India

ABSTRACT

In a clickjacking attack, a lot of effort has been put into researching client-side attacks, including such as cross-site scripting and request forgery, and more recently, clickjacking. Similar to other client-side attacks, clickjacking attacks can use the internet browser to utilize weaknesses in cross domain isolation and the single origin policy. It tricking the clients to click on something that is actually not what the user perceives they are clicking on. In the most severe cases, this vulnerability attack can cause an unsuspecting user to have their account compromised with an only a single click. Although there are some protections available for clickjacking attack, the web applications implementing these mitigations are too far and in middle cases. Additionally, although the possibility for an attacker to frame a page is easy to detect, it is more difficult to demonstrate or assess the impact of a clickjacking vulnerability than more traditional client-side vectors.

Keywords: Clickjacking, Internet Protocol address, Uniform Resource Locator, iFrame, Antivirus, Web Vulnerabilities

I. INTRODUCTION

By the turn of the century, information, including access to the Internet, will be the basis for personal and social economic and political advancement. The Internet supplements the traditional tools you use to gather information, Data, News and other resources correspond with other people .Improvements in the internet also leads to increase a threats and vulnerabilities. When a user wants to surf a live site they should be aware of the website whether it is secure or not. For that we created a tool to check whether the website is secure or not. If the websites contains any iframe tag then it is easy for attackers to attach web vulnerabilities to grab the information of the user. A websites can contain one or more number of iframe tags. If a website can show in the framed content means, it has the operations which lead a clickjacking attack.so we proposed a system which detect the clickjacking attack. Our tool consists of number of antivirus program that will detect the iframe tags and vulnerabilities present in the websites.

Objective

There are two main objectives for this project. First is to point out and illustrate some of these new threats that are accompanied with the implementation of new web standards. The second objective is to describe the derived protection mechanisms and explain how it could help defending against these threats.

II. METHODS AND MATERIAL

Despite wide-range of discussions and articles, click jacking attack still lacks a formal and actual definition. That manipulating the frame is also way of clickjacking attack. A click can also be stolen the details of the user frame. In filtration process easily get is IFRAME is existing in URL or not. If IFRAME is exist in URL that means that particular URL can be affected via Clickjacking attack. Although clickjacking has been the concept of many discussions and alarming reports, it is presently unclear to what extent clickjacking attack is being used by attackers.

In our study, we conducted the following experiment to assess the live web sites that implement called frame

IJSRST162238 | Received: 29 March 2016 | Accepted: 03 April 2016 | March-April 2016 [(2)2: 77-80]

busting technique. First of all, we have been prepared a web page that accepts only the single parameter denoting a URL that should be embedded in an IFRAME. Once the page contents (i.e.IFRAME) are finished loading and rendering, we verified that the IFRAME was still present or not. So it performs a frame busting technique that would alternate the all content in the browser window, thus removing the I-FRAME. To start this experiment, we implemented a Firefox extension that takes a list of URLs to be visited. Once a page is loaded, the extension waits for a few seconds and then verifies the presence of the IFRAME. If the IFRAME is not part of the documents DOM-tree any-more, we conclude that the embedded page performed frame-busting technique and it survive among 500 live websites. Using both known and novel attack techniques, we found that clickjacking defenses have been encountered could be circumvented in one way or another.

The search engine can search the URL of particular website, which needs to scan, and will call the Anti-Malware engines to perform the operations. A URL is a reference to a resource that specifies the resource location of the computer network and a mechanism for retrieving it. Here we are checking weather user giving valid URL or not .Here we are viewing the source code of given url and find the iframe tag is available or not.

I-FRAME: It is an HTML document enclosed inside another HTML document on a website. The HTML element is used to insert the code, such as an advertisement, into a Web Page. We are assimilating the I-frame tag from given URL and find out the domain name.

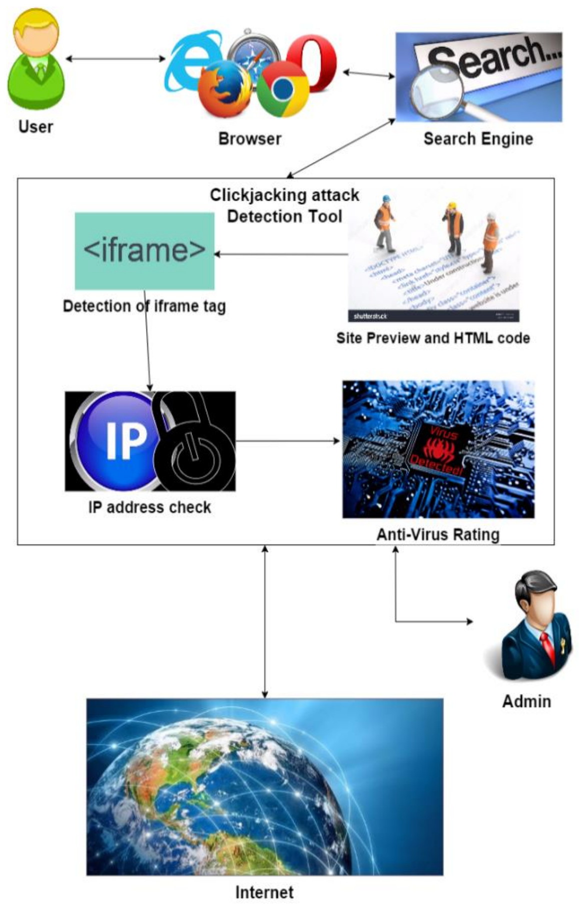


Figure1. System Architecture

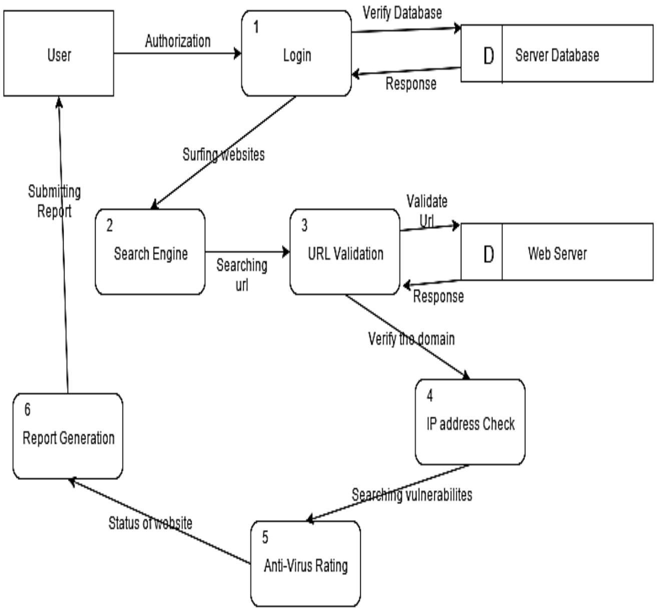


Figure 2. Data Flow Diagram

An Internet Protocol address is assigned to each of the device associate in a computer network that uses the Internet Protocol for communication. Here we are checking IP address because every domain having the ip address it was registered when they were launching the website. So we scan the IP address whether it was in block list or white list. Then the website was pushed into 29 different kinds of anti-virus program for scanning and generating the report. We have advanced how we assign web reputation to control pace with new types of attacks that can attack very quickly, or try to stay hidden. If we getting the Anti-virus Rating then well get surety of the domain is 100 percentages safe. We also provide protection against DDOS attack which is based on the blocking of particular IP address.

III. RESULT AND DISCUSSION

In our accession we encrypt the user's authentication information from other users by using md5/hash algorithm. We are analyzed the top websites for detection of clickjacking attack and other clickable elements. From our results, most of the website having IFRAME tag for representing their website has look and feel user interface.(i.e)IFRAME tag is used to view the images or videos at perfect scale position.

First we authorize a user for security purpose and also establish a authority to access this tool for particular user.

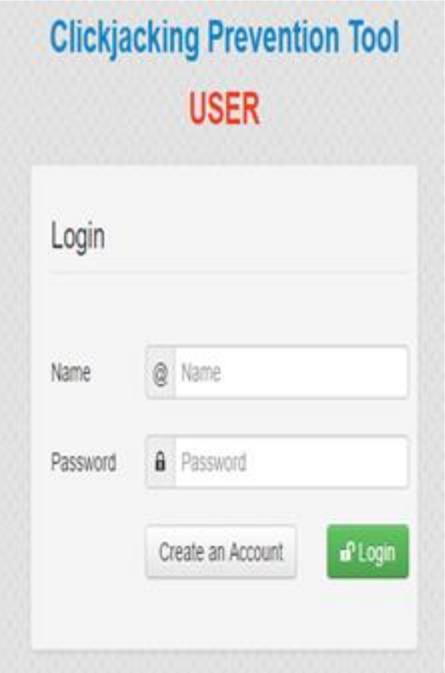


Figure 3. User Authentication

Here we attach a search bar for entering the URL for particular domain or websites, then it process the URL validation. It checks the URL which is presents in the internet or not. If the URL is presents in the internet, then it processes a IFRAME tag counts and IP address check.

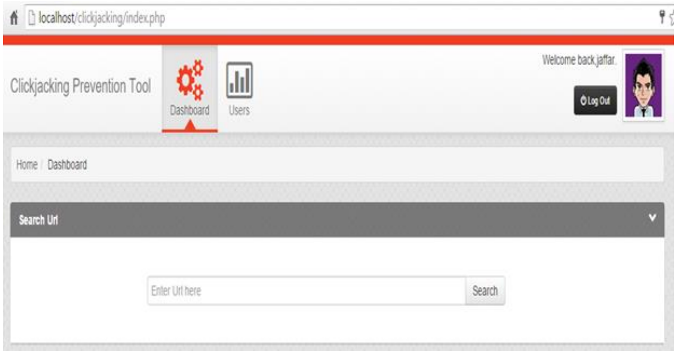
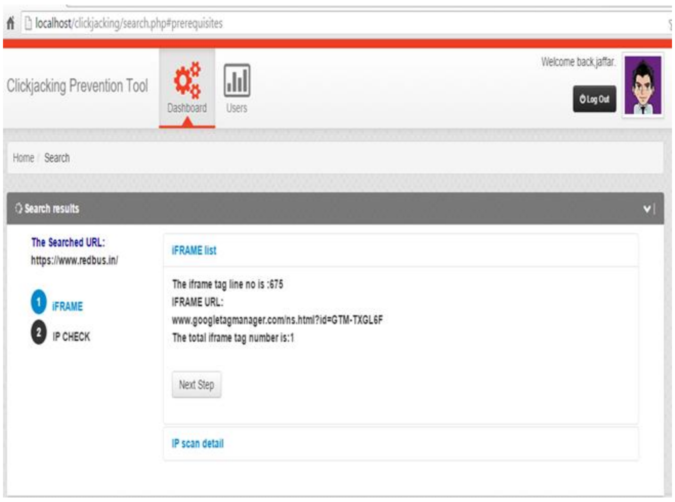


Figure 4. URL Validation & Search Bar

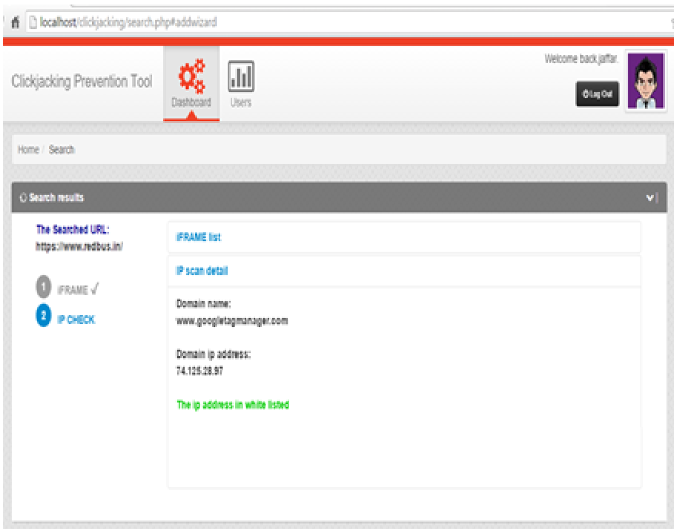
IP address check capturing the domain IP addresses which are specified in the URL. Then it verifies the domain IP address for blacklist or whitelist. If the IP address is in blacklist means that it shows the report deny for user. Otherwise it will push forward to running the anti-virus program.



The Searched URL:
https://www.redbus.in/

The iframe tag line no is :675
IFRAME URL:
www.googletagmanager.com/ns.html?id=GTM-TXGL6F
The total iframe tag number is:1

Figure 5. iFrame Tag Counts



The Searched URL:
https://www.redbus.in/

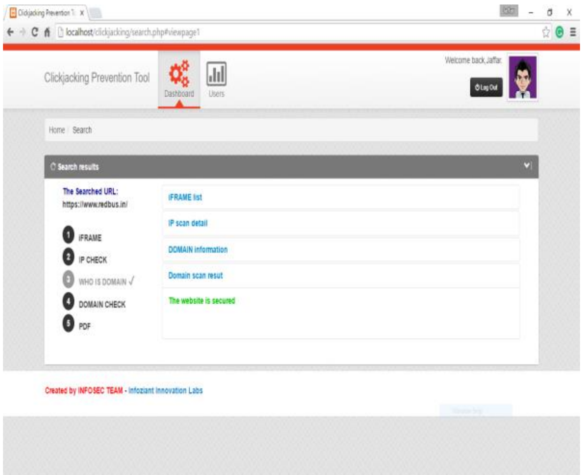
Domain name:
www.googletagmanager.com

Domain ip address:
74.125.28.97

The ip address in white listed

Figure 6. IP Address Check

Finally, the anti-virus program rating generates a report for the particular domain as safe or not. Report describes all the information about the domain and risk factors. From the report user can aware about the domain and threats in the internet.



The Searched URL: https://www.redbus.in/

The website is secured

Figure 7. Anti-Virus Check

IV. FUTURE ENHANCEMENT

Click jacking attack is in its initial stage and focuses only on detection technique. This tool can be upgraded in the future for preventing click jacking attack. In future, this tool can also implement in the server side to prevent the web vulnerabilities and also improve the security for their users' information.

V. CONCLUSION

Clickjacking is a web attack that has newly gathered wide media coverage. There have been many news items, discussions, and forums on the topic. In this paper, we conferred our system able to detect the clickjacking attempts on web pages. We validated our tool and we conducted empirical experiments to estimate the prevalence of such attacks on the Internet by automatically testing more than one million web pages that are likely to contain malicious content and to be visited by Internet users. By distributing the analysis on multiple virtual machines we were able to scan up to 15,000 web pages per day. Even though the pages containing these clickjacking attacks have been posted as examples on security-related websites, we found them automatically. Furthermore, in our analysis, we also detected several other interesting cases that we call borderline attacks. Such attacks are difficult to accurately classify as either being real attacks, or false positives.

VI. REFERENCES

[1] Y.-W. Huang, S.-K. Huang, T.-P. Lin, and C.-H. Tsai. Web application security assessment by fault injection and behavior monitoring. In WWW 03: Proceedings of the 12th international conference on World Wide Web, pages 148159, New York, NY, USA, 2003. ACM.

[2] S.Kals, E.Kirda, C.Kruegel, and N. Jovanovic. Secubat: a web vulnerability scanner. In WWW 06: Proceedings of the 15th international conference on World Wide Web, pages 247256, New York, NY, USA, 2006. ACM.

[3] N. Jovanovic, C. Kruegel, and E. Kirda. Pixy: A static analysis tool for detecting web application vulnerabilities (short paper). In IEEE Symposium on Security and Privacy, pages 258263, 2006.

[4] G. Wassermann and Z. Su. Sound and precise analysis of web applications for injection vulnerabilities. SIGPLAN Not., 42(6):3241, 2007.

[5] Y.Xie and A. Aiken. Static detection of security vulnerabilities in scripting languages. In USENIX-SS06: Proceedings of the 15th conference on USENIX Security Symposium, Berkeley, CA, USA, 2006. USENIX Association.

[6] Y.-W. Huang, S.-K. Huang, T.-P. Lin, and C.-H. Tsai. Web application security assessment by fault injection and behavior monitoring. In WWW 03: Proceedings of the 12th international conference on World Wide Web, pages 148159, New York, NY, USA, 2003. ACM.

P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna. Cross site scripting prevention with dynamic data tainting and static analysis. In Proceedings of the Network and Distributed System Security Symposium, NDSS 2007, San Diego, California, USA, 28th February - 2nd March 2007, 2007.

[7] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser analysis of web-based malware. In HotBots07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pages 44, Berkeley, CA, USA, 2007. USENIX Association.

[8] Y.-M. Wang, D. Beck, X. Jiang, and R. Roussev. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In IN NDSS, 2006.