IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

nCORETech 2017

LBS College of Engineering, Kasaragod

Vol. 6, Special Issue 3, March 2017

A Study on Clickjacking Attack Detection Techniques

Karthika K[1], Divya B[2]

PG Student, Computer Science & Engineering, Vimal Jyothi Engineering College, Kannur, India[1]

Associate Professor, Computer Science & Engineering, Vimal Jyothi Engineering College, Kannur, India[2]

Abstract: The tremendous growth of World Wide Web made the world to joined as together. It paved the way to minimize the distance of each other. Even though it provides greater advantages to internet users, the same way it poses great insecurity to information. Web security is an important issue of concern nowadays. Web attacks are increasing day by day. Clickjacking attack is a web framing attack that has recently received wide media coverage. In a clickjacking attack, the attacker creates a malicious page such that it tricks into clicking on an element of a page that is only barely visible. It redresses the users click for malicious purposes. By stealing user's click, they force the users to perform unintended actions, thus posing a significant threat to the Internet users. Although clickjacking attack has been the matter of discussion and disturbing reports, it is currently unclear what extent it clickjacking is used by attackers. But still, it is a problem incurred by typical internet users. This paper focuses on various clickjacking defenses on different web browsers.

Keywords: Information Security, Web Security, Clickjacking, Web Browser, Web framing attack.

I. INTRODUCTION

Internet users are increasing day by day. Using the Internet the users can do anything right from business, information exchange, shopping, entertainment, education, banking and so on. With the growth of Internet usage, security issues have also increased. Cyber-attacks are increasing day by day and so web security is always a challenging issue for researchers. Information should be protected from unauthorized access, use, disclosure, modification, etc.

Web framing attacks are getting popularity nowadays.These attacks focus on the web browser and displays a malicious page on victims page. It displays a page that is different from the original one. Clickjacking attack comes under the domain of web framing attacks. Clickjacking is an emerging web treat nowadays.It was first reported in June 2002 by Ruderman who noted that transparent iframes can lead to security problems in the Mozilla bug tracking system. The term Clickjacking was coined by Jeremiah Grossman and Robert Hanseen in 2008. It is also known as "cross-domain attack" , because the attack is performed by hijacking user-initiated mouse clicks.

Most prominent example of Clickjacking attack was an attack against "Adobe Flash Plugin settings page." An invisible iframe was created, and by loading the settings page into this, the attacker could trick the user into alerting the security settings of flash, giving permission for any flash animation to utilize the computer's microphone and camera.

Clickjacking attacks can be done in various ways.For example, imagine an attacker who builds a website with a button that says "click here for a free trip." The attacker has loaded an iframe with opacity set to zero. When a user tries to click on "free trip" it actually clicks on the invisible frame.The user won't see that he or she is actually about to be charged purchases to their credit card.

Clickjacking may sees easy to execute at first because it uses some basic HTML and JavaScript features to frame a page or create layers on a single page. But, the detection and prevention of a clickjacking attack are tough because there are no established procedures to test it like any other bug or vulnerability. Also uses avoidable features of style sheets and scripting languages [1].

There are various clickjacking defensive methods such as client-side defenses or server-side defenses. Even though techniques are there to mitigate this attack, none of the methods provide a complete solution. In this paper, various clickjacking prevention techniques are discussed.

A. LITERATURE REVIEW

A. Frame busting

Frame busting method proposed by Elie Bursztein et al [2] consists of a script in each page that should not be framed. The main aim of this technique is to prevent the web page from being loaded in a sub frame[3]. It includes JavaScript code snippets in pages that need protection against clickjacking. The code snippet checks if a loaded web page is at the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
nCORETech 2017
LBS College of Engineering, Kasaragod
Vol. 6, Special Issue 3, March 2017

top window or not. If the current page is not in the top window, then it is loaded on the top window, thereby stopping the page loading in the iframe [4] [5].

B. X-Frame-OPTIONS HTTP Response Header

It is a server-side approach to control clickjacking [6]. HTTP Header set on HTTP responses. It takes any of the given three possible values.

a. DENY: If this value is assigned to the response header no domain can frame the content.
b. SAMEORIGIN: Only the current domain can frame the content if this value is used.
c. ALLOW-FROM <uri>: Only the specified <uri> is allowed to frame the content of the web page.

It has the limitations such as Per-Page Policy Specification, Problems with the multi-domain site, Proxies.

C. Content Security Policy

Stamm et al [10] proposed a content security policy to protect the web from security attacks. It provides a standard method to declare approved origins of content that browsers should be allowed to load on that page. CSP is delivered via HTTP response header. But it is more general than X-Frame-options.

D. No-Script :ClearClick

No-Script [8] is a free, open-source extension for Mozilla, Firefox, sea monkey, and other Mozilla based web browser. It allows the executable web content such as javascript, java, flash, silver light and other plugins only if the site hosting is considered trusted by its user and previously added to its whitelist.This provide an anti-clickjacking method called clearclick [9] to address clickjacking attack

E. In Context

The main cause of Clickjacking is that an attacker application presents a sensitive UI element of a target application out of context to the user, and the user gets tricked to act out of context. InContext enforces target display integrity by comparing the OS-level screenshot of the area that contains the sensitive element and the bitmap of the sensitive element rendered in isolation at the time of user action. If these two bitmaps are not the same, then the user action is canceled and not delivered to the sensitive element. This design is resilient to new visual spoofing attack vectors because it uses only the position and dimension information from the browser layout engine to determine what the user sees [3].

F. Trueclick

Ali Osman Ulusoy et al [10] implemented a tool called TrueClick, which uses image processing and machine learning techniques to build a classifier based on these features to automatically detect the trick banners on a web page. This tool assist Internet users in automatically detecting malicious trick banners and helping them identify the correct link based on their visual properties such as image size, color, placement on the enclosing web page, whether they include animation effects, and whether they always appear with the same visual properties on following loads of the same web page. It is implemented as a browser extension that runs on demand when the user visits a file sharing website containing trick banners and clicks on a button to activate the system. Once the analysis of the banner images is complete, TrueClick can either mark the detected trick banners as such or block them entirely.

G. Proclick

Hisham Haddad et al [12] proposed a proxy level framework to detect Clickjacking attacks. ProClick examines the content of requests and response pages at the proxy level to detect Clickjacking attacks. Proclick analyzes web pages before being rendered to the browser for detecting Clickjacking attack symptoms. The framework is a client side proxy which can intercept the incoming requests and response pages. It analyzes the parameter values of request pages and HTML JavaScript code of response pages and performs systematic checking to decide if attack symptoms are present or not. The framework can also be extended to detect new Clickjacking attacks. It also incurs negligible performance overhead and does not break up the source code of legacy web applications. It does not depend on specific HTTP header or the enabling/disabling of JavaScript at the browser. ProClick is the core module that has the capability to analyze both requests and response pages. It performs a number of checks based on the specified policy for detecting Clickjacking attacks.

H. Adaptive User Interface Randomization

Providing the same web page layout to each user gives the attacker an easy chance to size and position the attack elements inside the victim page so that the user clicks on the exact target region and his click can be hijacked to be used for the attacker's advantage. Hill et al [7] proposed a randomization technique to mitigate the clickjacking attack.

Randomizing the placement of a button and recording and analyzing the missed clicks, can prove to be an effective approach to determine clickjacking attempts. But it has the drawback of having poor user experience and also the attacker can induce the victim to send multiple clicks to target application thus exhausting randomizing possibilities.

I. ClickProtect

ClickProtect proposed by Dipti et al [13] provides a secure way to use the internet without the problem of web security. It is a browser extension that runs on the browser while the user is surfing on the internet. This method extracts the clickable elements and stores it in an array. For these elements, it checks whether the visibility has opacity value less than 1. If it is true then it will make those elements visible and will notify the user. It also checks for cursor properties. If cursor style is set to none, It will notify the user that cursor is customized and it the makes the original cursor visible.

J. ClickSafe

Jawwad A. Shamsi et al [17] proposed Clicksafe, a browser-based tool to provide security against clickjacking attacks. It consists of three major components. The detection unit detects malicious components in a web page which redirect users to external links. The mitigation unit provides interception of user clicks and gives educated warnings to users who can then choose to continue or not. Clicksafe also incorporates a feedback unit which records the user's actions, converts them into ratings and allows future interactions to be more informed. It is predominant from other similar tools as the detection and mitigation are based on a comprehensive framework which utilizes detection of malicious web components and incorporating user feedback.

III. CONCLUSION

Clickjacking is one of the dangerous attacks. Users should be aware of such attacks. A user's browser can come under the control of an attacker who can steal confidential information. These attacks can be prevented to an extent by using the latest and up-to-date software products and by following rational security practices.

REFERENCES

[1] Shamsi, Jawwad A., et al. "Clicksafe: Providing security against clickjacking attacks." 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering. IEEE, 2014
[2] Rydstedt, E. Bursztein, D. Boneh, and C. Jackson, "Busting Frame Busting: A study of clickjacking vulnerabilities at popular sites", IEEE Oakland Web 2.0 Security and Privacy (W2SP 2010).
[3] L.S. Huang, A. Moshchuk, H. J. Wang, S. Schechter, and C. Jackson, "Clickjacking: Attacks and defenses," in USENIX Security Symposium. USENIX Association, 2012.
[4] M. Balduzzi, M. Egele, E. Kirda, D. Balzarotti, and C. Kruegel, "A solution for the automated detection of clickjacking attacks," in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '10. New York, NY, USA: ACM, 2010, pp. 135-144.
[5] Wang, H. J., Grier, C., Moshchuk, A., King, S. T.,Choudhury, P. and Venter H. 2009. "The Multi-Principal OS Construction of the Gazelle Web Browser." In Proceedings of the 18th Conference on USENIX Security Symposium.
[6] Hill, Brad. "Adaptive user interface randomization as an anti-clickjacking strategy." May (2012).
[7] Maone, Giorgio, http://noscript.net/
[8] HMaone G. 2008. "Hello ClearClick, Goodbye Clickjacking!" http://hackademix.net/2008/10/08/helloclearclick-goodbye-clickjacking/. Last accessed 16th, Dec 2015.
[9] Stamm, Sid, Brandon Sterne, and Gervase Markham. "Reining in the web with content security policy." Proceedings of the 19th international conference on World wide web. ACM, 2010.
[10] Duman, Sevtap, et al. "Trueclick: automatically distinguishing trick banners from genuine download links." Proceedings of the 30th Annual Computer Security Applications Conference. ACM, 2014.
[11] Narayanan, A. S. 2012. "Clickjacking Vulnerability and Countermeasures," International Journal of Applied Information Systems (IJAIS) Foundation of Computer Science FCS, New York, USA Volume 4-No.7, December 2012, pp. 7-10.
[12] Shahriar, Hossain, Vamshee Krishna Devendran, and Hisham Haddad. "ProClick: a framework for testing clickjacking attacks in web applications." Proceedings of the 6th International Conference on Security of Information and Networks. ACM, 2013.
[13] Pawade, Dipti, et al. "Implementation of extension for browser to detect vulnerable elements on web pages and avoid Clickjacking." Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference. IEEE, 2016.
[14] G. Rydstedt, E. Bursztein, D. Boneh, and C. Jackson. "Busting frame busting: a study of clickjacking vulnerabilities at popular sites," in IEEE proceedings of the Web 2.0 Security and Privacy, 2010.
[15] S. Lekies, M. Heiderich, D. Appelt, T. Holz, and M. Johns, "On the fragility and limitations of current browser-provided clickjacking protection schemes," in Woot 2012, USENIX Security Symposium. USENIX, 2012.
[16] Brigette Lundeen, Dr. Jim Alves-Foss, "Practical Clickjacking with BeEF" published under IEEE 2012 and accessed from http://beefproject.coml [14] Ubaid Rehman, Waqas Ahmad Khan, Nazar Saqib, Muhammad Kaleem, "On Detection and Prevention of Clickjacking Attack for OSNs", 11th International Conference on Frontiers of Information Technology, 2013.
[17] Shamsi, Jawwad A., et al. "Clicksafe: Providing security against clickjacking attacks." 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering. IEEE, 2014.