International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

A Review on Clickjacking Attack and its Defense Mechanism

Puneet Kour[1]

[1]*M.Tech. Student, Dept. of Computer Science & IT, University of Jammu, Jammu & Kashmir, India*

Abstract – *The tremendous growth of cyber attacks have become most prevalent in the past few years. Every time attackers have discovered new web vulnerabilities to carry out malicious activities on the internet. One such attack is a clickjacking attack and that have been adopted by the attackers to deceive the innocuous internet users to initiate some action. A clickjacking attack is a user interface redress attack in which the attacker creates a malicious page of tricking a user into clicking which is something different from the user awareness, thus it revealing confidential information of the user and allowing others to take control of their computer while clicking on some unoffending objects including web pages. This paper focuses on describing the implementation of various possible clickjacking methods and describing the hazards in various prevention techniques of clickjacking attack and so we suggest some mitigation mechanisms to prevent this vulnerability.*

Key Words: Web Vulnerabilities, Clickjacking, Web framing attack.

1. INTRODUCTION

In this modern era, information sharing over the Internet is everyday business. So, securing the data means preventing misuse due to unauthorized access. But on the other hand, attackers always keep looking for flaws, vulnerabilities, and bugs in the web applications, browsers, servers, and the other components of the web to carry out malicious activities. So information should be protected from unauthorized access, used, modified, etc is always a challenging issue for researchers. The exploitation of one such web vulnerability is clickjacking attack (also known as User Interface redress attack, UI redress attack, UI redressing). In this attack, the victim is tricked into interacting with a User Interface (UI) element that they do not see. The attacker designs positioned visual elements which are actually malicious pages. The user is lured into clicking on these elements but, in the reality, unknowingly clicks on an element on a different page.

In the year 2008, the term 'Clickjacking' was coined by Jeremiah Grossman and Robert Hansen. According to them, Clickjacking is known as cross-domain attacks because the attack is performed by hijacking user-initiated clicks [1]. A clickjacking attack is a technique that tempts the victim into clicking on a specific element of a webpage, while the victim intends to interact with the content of a different website. The victim clicks on an element of the attacker's choice under the misconception which is apparently a harmless page. In order to perform the clickjacking attack, the attacker loads a malicious page from the target website inside an iframe with the help of Cascading Style Sheets (CSS) then the attacker can hide everything except the targeted region of the page. The attacker has loaded an iframe with zero opacity and when the user clicks on the interactive page it actually clicks on the invisible frame. The targeted region made fully transparent and placed on top of another element on the attacker's page can either be displayed as a part of the attacker's page, known as User Interface redressing, or made fully transparent and placed on top of another element on the attacker's page [2]. All most of the web browsers like Opera, Google Chrome Firefox, etc have been the victims of this attack. Social networking websites have been one of the most attacked groups of web applications. Apart from the Adobe Flash, the "Twitter bomb" is known to be one of the worst clickjacking attacks. There are many bank websites, web applications, and some open-source web applications have no defense against clickjacking attack in 2012 [3]. Fig. 1 depicts the architecture of the clickjacking attack.

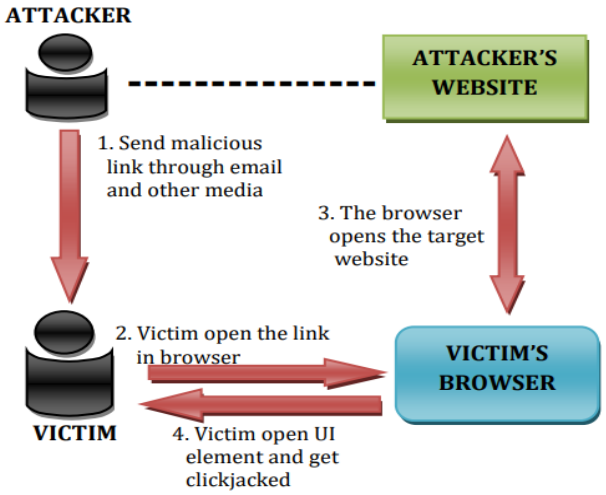

Fig- 1: Architecture of Clickjacking attack

2. CLASSIFICATION OF CLICKJACKING

The clickjacking attack aims at both user and the web application in which the attacker tricks the victim to perform an action which victim doesn't intend to. Existing clickjacking attacks are classified as [4]:

2.1 Compromising target display integrity

Hiding the target element- The hacker hides the target element using HTML/CSS styling sheets, but mouse events still work. The attacker can carry out this attack by making the target element transparent which is of zero opacity.

Partial overlays- The attacker confused a victim by making only the target element opaque. The partial overlaying is done with the help of the z-index property of CSS and thus compromising target display integrity.

Cropping- Hacker can crop the target element to show only a small piece of the target element.

2.2 Compromising pointer integrity

Attacker compromises the pointer integrity by displaying a fake cursor by hiding the actual default cursor, known as cursorjacking. This way victim confounds a click's target.

2.3 Compromising temporal integrity

The attacker changes a user interface element after the user decides to click. When the actual click occurs the attacker could move the target displayed element on top of a button shortly after the victim hovers the cursor over the button, in expectation or prediction of the click.

Attackers have continued to compromise the context integrity of web applications in the past few years and continue to do so; providing terms variants of the attack such as likejacking, web framing, cursorjacking, mouseJacking, strokejacking, drag-and-drop clickjacking, etc.

3. LITERATURE WORK

The goal of a clickjacking attack is to trick the victim into performing unwanted actions by hijacking the victim's clicks. Various researchers have worked in the field for the detection and the prevention of clickjacking attacks. Researchers are still carrying out their studies on this domain. Some of them are listed as:

Jawwad A. Shamsi et al [5] proposed Clicksafe tool, which is a browser-based tool to provide security against clickjacking attacks. It consists of three major components one is a detection unit that detects malicious components in a web page one is a mitigation unit that provides interception of user clicks and gives educated warnings to users. The tool also incorporates a feedback unit that 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 that utilizes detection of malicious web components and incorporating user feedback.

G. Rydstedt and et al. [6] surveyed frame busting code the Alexa Top-500 websites including networking, banks, social networks, online merchandise, trading, and gaming websites sites and proposed a JavaScript-based defense against clickjacking attack.

Stamm et al [7] proposed a content security policy (CSP) to protect the web page 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.

Dipti Pawade and et al. [8] proposed a JavaScript-based browser extension called "ClickProtect" for Google Chrome which provides a solution for multiple clickjacking attacks. It alerts the user before proceeds towards an unsafe action with a pop-up message.

R. P. Seenivasan and K. Suresh Joseph [9] proposed a system anti-clickjacking technique with security and cost metrics to get prevent clickjacking attacks.

Yusuke Takamatsu and Kenji Kono [10] proposed a "Clickjuggler" tool for checking defenses against visual clickjacking during development. They implemented the clickjuggler tool as a plug-in for Firefox 20.0.1 and 3.6.8 using Firefox plug-in interface. They applied the juggler tool to four real-world web applications including Joomla, WordPress, MediaWiki, and Roundcube.

A. Sankara Narayanan [11] provided a clear view of numeric figures for the clickjacking google results. He compared the numeric figures of clickjacking attacks with other browser-based attacks on the basis of google results. He using various online tools to implemented a clickjacking attack and for the mitigation of the attack using available anti-clickjacking tools.

Brigette Lundeen and Jim Alves-Foss [12] proposed a plug-in module for Browser Exploitation Framework. It provides a tool designed to help professional penetration testers easily demonstrate the impact of client-side security vulnerabilities of a clickjacking attack.

4. EXISTING DEFENSE MECHANISMS

Some defense policies are to b adopted by users and the web developers to get rid of such type of attack to some extent. The following defense mechanisms are:

I. Content-Security Policy- Content-Security Policy (CSP) is used via HTTP response header. But for clickjacking attack it includes an anti-clickjacking frame directive. CSP provides a method to declare approved origins of the content that browsers should be allowed to embed on that page. CSP

covers the following tags live frame tag, iframe tag, object tag, embed tag, meta tag, etc.

II. X-Frame-Option- X-Frame-Option (XFO) is also an HTTP response header thatis introduced by Microsoft Internet Explorer. XFO is the server-side approach for clickjacking attack. XFO may used directly with the HTTP header. It supports three main directive values:

a. Deny- If the web application use deny value then the response header cannot allow any domain to display the page within a frame.

b. Sameorigin- If the web application use sameorigin value then it can be displayed from the web pages with the same origin.

c. Allow- From URI- If the web application use allow- from URI value then it allow only the current displayed frame with only specific <uri>.

III. Frame Busting- Frame busting is also known as frame breaking which is a client-side technique. In this technique, there is no use of the HTTP header. In frame busting there is only to modify the HTML web page code.

5. DISCUSSION

In the above discussion, the conclusion is that there is still a need for some relevant prevention solutions to get rid of this clickjacking attack because every time hackers find out another approach to revivify the attack successfully. Many authors study to overcome the clickjacking attack, also the above work can be extended to provide better and powerful solutions for clickjacking attack by means of parsing techniques to identify the attacking spots before the hacker attack.

6. CONCLUSION

Clickjacking is one such vulnerability and the probability of such attack is quite higher when lacking awareness among a large group of web users. There have been a variety of mitigation techniques, but the hacker has already bypassed those techniques in one way or the other. Users should be aware of such vulnerabilities on the internet. Existing solutions and defense mechanisms to mitigate clickjacking attacks do not provide complete protection. The above work can be extended to provide some better solutions against clickjacking attacks or maybe developing efficient techniques for both client-side and server-side to protect the web users from clickjacking attacks in the future.

REFERENCES

1. Robert Hansen and Jeremiah Grossman, “Explanation of Clickjacking”. Available: http://www.sectheory.com/clickjacking.htm

2. Context Information Security Ltd, Next Generation Clickjacking, London, 2010. Available: http://www.contextis.co.uk

3. Dingjie Yang, Clickjacking: An Overlooked Web Security Hole, 2012. Available: https://blog.qualys.com/securitylabs/2012/11/29/clickjacking-an-overlooked-web-security-hole

4. Lin-Shung Huang, Alex Moshchuk, Helen J. Wang, Stuart Schechter and Collin Jackson, Clickjacking: Attacks and Defenses, Proc. USENIX Security Symposium, Bellevue, WA, 2012, 413-428.

5. Shamsi, Jawwad A., et al. "Clicksafe: Providing security against clickjacking attacks." Proc. of IEEE 15th International Symposium on High Assurance Systems Engineering, 2014.

6. G. Rydstedt, E. Bursztein, D. Boneh and C. Jackson, “Busting frame busting: a study of clickjacking vulnerabilities at popular sites”, Proc. IEEE Web 2.0 Security and Privacy, Oakland, CA, 2010, 1-13.

7. Stamm, Sid, Brandon Sterne, and Gervase Markham. "Reining in the web with content security policy." Proc. In 19th international conference on World Wide Web ACM, 2010.

8. Dipti Pawade, Era Johri, Divya Reja and Abhilasha Lahigude, Implementation of Extension for Browser to Detect Vulnerable Elements on Web Pages and Avoid Clickjacking, Proc. in 6th IEEE International Conerencef on Cloud System and Big Data Engineering, Noida, India, 2016, 226-230.

9. R. P. Seenivasan and K. Suresh Joseph, A Survey of Clickjacking Attack and Countermeasures in Web Environment, International Journal of Advanced Research in Computer Science and Software Engineering, 6(12), 2016, 206-213.

10. Yusuke Takamatsu and Kenji Kono, “Detection of Visual Clickjacking Vulnerabilities in Incomplete Defenses”, Proc. of IEEE Journal of Information Processing, 23(4), 2015, 513-524.

11. Sankara Narayanan, “Clickjacking Vulnerability and Countermeasures”, International Journal of Applied Information Systems, 4(7), 2012, 7-10.

12. Brigette Lundeen and Jim Alves-Foss, "Practical Clickjacking with BeEF", Proc. IEEE Conference on Technologies for Homeland Security (HST), Massachusetts, USA, 2012, 614-619.

13. Clickjacking attack. Wikipedia. Available: https://en.wikipedia.org/wiki/Clickjacking
14. Clickjacking attack. Acunetix. Available at: https://www.acunetix.com/blog/web-security-zone/defend-against-clickjacking-attacks/

15. Clickjacking UI redressing. Available at: https://portswigger.net/web-security/clickjacking

16. Clickjacking attack and its prevention measures. Available at: https://owasp.org/www-community/attacks/Clickjacking