ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

A Survey on Clickjacking and Tapjacking Solutions Provided by Different Browser

Shreya Bapat[1], Sayali Kandarkar[2], Neha Gupta[3], Prof. Dipti Pawade[4]

B. E. Student , Dept. of I.T., K. J. Somaiya College of Engineering, Mumbai, India[1,2,3]

Assistant Professor,Dept. of I.T., K. J. Somaiya College of Engineering, Mumbai, India[4]

ABSTRACT: Today's computers face so many malicious attacks more than ever. There has been a huge increase in such attacks in the past 2 to 3 years so that even the security researchers are finding keeping their heads above water tough especially in the times when malware can be easily delivered by advertisements on the legitimate sites like yahoo, Google etc. This paper describes UI redressing attacks like clickjacking and tapjacking. Clickjacking is basically an attack where an attacker using various social engineering practices deceives the user into clicking on a button or a link by making apt use of transparent layers. Tapjacking is a contemporary form of clickjacking where mobile phones are the targets of malevolent attackers. This paper also presents the latest and up to date solutions to the clickjacking attacks with respect to various browsers**.**

KEYWORDS: Clickjacking, tapjacking, Google chrome, Firefox, Internet Explorer.

I. INTRODUCTION

With boom in use of internet the security issues and cyber fraud cases are intensified to great extends. Earlier very few phishing attempts were there. But now clickjacking has emerged as one of the biggest challenge to the security researcher. In clickjacking users mouse clicks are stolen or these clicks are used to redirect the user to the nasty page through which phishing can be performed easily [1]. Security researchers had come up with various preventive techniques for clickjacking. Section II discuss about various solution provided by web browser like Google chrome, Firefox, Internet Explorer. When everyone was thinking that clickjacking is somewhat in control; its contemporary version called "Tapjacking" is introduced. In tapjacking, attacker has changed their target to mobile phones. Section III covers the concept of tapjacking and its solution.

II. BROWSERWISE SOLUTION FOR CLICKJACKING

In this section we have discussed different solutions provided by various web browsers.

A. *Solution in Google Chrome:*

Clickjacking is a very nasty attack. There are two options to prevent Clickjacking in Google chrome. The first solution is to use JavaScript framebusters to have your website break out of iframe, and the other solution is to set the X-FRAME-OPTIONS header. X-frame-option is the better solution as it solves the problem completely by blocking framing. X-frame-options are server-side set http headers. They are segment of the security protection against malicious attacks like Clickjacking. Chrome will not let you ignore or modify it [2].

Moving towards JavaScript framebusters, there are some problems with JavaScript framebusters. They are, most of them attempted to 'break out' of the frame, substituting the main window with their file. However the attacker file in the parent window can simply hold out from being navigated and though there are various methods for overcoming a problem, they can always be bypassed. If your script detects that, you have been framed; the preferable solution from a security perspective is to make the whole page unusable. For instance, by putting a document.body.innerHTML to an error warning. The parent-navigation approach occurs because framebusters emanated as a means for webmasters to get away from undesirable framing from various sites.

The next issue is, framebusters can be averted from running the frame in an iframe just because the user has JavaScript turned off. One can demystify this by having the page work only when JS is enabled, but that has clear negative and adverse detrimental accessibility impact [2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

B. *Solution in Mozilla Firefox:*

In order to achieve the desirable solution so as to prevent clickjacking, we need to enhance the functionality of web browsers so that we can perceive the hidden iframes and spiteful javascript and further take suitable,well judged and timely action against them.The solutions are as follows [4]

Solution 1: Disable scripting and plugins

By selecting tools>add ons>plugin>disable, we can keep the malicious attackers from creating transparent layers and luring the users into opening harmful sites.Once we disable the plugins we need to shut down the browser and delete adobe flash and remove it from the system.

Solution 2: By downloading the latest version of No Script Firefox plugin

Once we download the latest version of NoScript plugin, NoScript plugin pre-emptively blocks all the malign sites and let the user access java, javascript from only those sites which are trusted and authorized.Once we install NoScriptplugin,we need to restart the system then select the NoScript icon located at the bottom of the right status bar>options>forbid[iframe]>ok. Thus, we can be assured of the fact that users will be able to gain access to only safe and trusted sites.

Solution 3: X-Frame options

One of the best solutions to clickjacking is X-Frame server response header. We need to set the X-Frame options header to same origin or deny to prevent the target web application from being loaded with an i-frame hence the clickjacking attacks will be averted successfully.

C. *Solution in Internet Explorer*

The Internet explorer was launched in August 16, 1995.It has various versions from IE1 to the latest that is IE11.Security threats have always been a major debacle in these versions. With the advent of the threat of click jacking in the year 2008, it became necessary to find a solution against it for the various IE versions. The solution that was devised to tackle this attack was by using Xframe options response header. It is used to check whether the page that will be displayed is safe to be displayed in a frame or no. There are three values for this [3]:

- Deny: the page would not get displayed in the frame even if the site tries to do so.
- Same origin: Page A will get opened in a frame if and only if the site that contains the frame is same as the site which will be serving that Page A in the frame.
- Allow from URI: The page gets displayed in a frame on the origin that is specified.

III. TAPJACKING

Clickjacking is an attack where an attacker entices users to click on a button or a link by making tricky use of opaque or transparent layers. Just like Clickjacking on the web, Tapjacking takes place when a malicious application is created by developer that lures users into tapping on a pop-up window (called toast window), making it a gateway for a lot of threats. This technique is not much complicated but can cause serious threats to the Android users. Using the techniques of clickjacking and tapjacking, an attacker could deceive the users into clicking on advertisements, wiping the entire data from the computer/mobile, granting private information etc.

On Android, temporary notifications called toasts are used to pop up a message on the surface of a window. Simple toast notifications only take up small part of the screen however a malicious user can customize a toast to take up the entire window. Toasts have a lifespan of only 3.5 seconds so the attacker has to always make sure that the user only sees the fake user interface and not the targeted one. In this way, the attackers try to obscure the real user interface. However there is a short period of time between the end of one toast and start of another one where the real user interface is visible for that period of time. But if in that period of time a legitimate activity having the same view is displayed but the attacker, toasts can be launched repeatedly.

In order to prevent the interaction events, Android 2.3 added the ability for views so that they can obscure another views. Tapjacking is basically a technique wherein the background layer is hidden behind a foreground layer .When the user actually taps the foreground layer, the background layer gets triggered. Depending on various permissions that an application gets while installing, it can get an access to various things in the cellphone. So it is better not to download applications from unknown sources. Here are a few examples:-If the wallpaper of your device gets suddenly changed

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 12, December 2015

then you should be careful because as the malicious developers can get an access to your phone's settings then he may also extract other important information from your phone. The permission that the application needs in this scenario is set_wallpaper.

Next type is getting access to user's credentials by GET_ACCOUNTS permission. From this the malicious developer can get access to really confidential information. Another type can be where the developer can get an access to user's settings and may change the password of the cell-phone and the user would not be able to unlock it. With the advent of social networking and smartphones, we now use phones to access various social networking sites like twitter, Facebook. We often perform status updates. With the various applications getting an access to all these accounts themalicious developer can perform random status updates without user's consent. Some of the solutions for it are:

- Sandbox detection:
  In this technique a copy of applications virtual machine is made and applications are kept isolated from each other.
- Machine learning detection:
  These are algorithms used to detect malware in the applications and codes by analyzing the pattern of the codes.
- Static analysis detection:
  This includes various things like scanning the source code and detecting threats or scanning the binary code to detect threats. It also includes checking various parameters like CPU usage, memory occupied. If it suddenly exceeds then there is a threat. Threats can also be detected comparing source code from original sites with those from third-party codes. [11]

IV. CONCLUSION

Clickjacking is one of the attacks which don't get a lot of importance normally but it is dangerous. Many blogs, news items have been written on this topic however the extent upto which it can exploit private and important content is unclear. Thus, it's on us to be aware of such attacks. We need to be heedful towards the preventive measures of Clickjacking. As per the experts, Clickjacking entails targeting an innocent user onto a malicious website. Because of this, the user's browser could totally come under the control of the hacker who can perform dangerous tasks on the behalf of the user or may steal his confidential information. Clickjacking attacks are mostly website and browser dependent. There's no easy fix for this design bug. It's very likely that we'll continue to experience variations of clickjacking attacks in the near future. The best way to protect ourselves is to follow rational security practices and use only the latest, most up-to-date, and fully patched versions of our software products. As we started shifting from computers to smartphones it lead to the advent of tapjacking which is similar to clickjacking but is done on smartphones.Both these attacks can be highly dangerous and may lead to revealing of confidential information. So, we must take up the solutions cleverly to prevent Clickjacking and Tapjacking attacks on our browsers and smartphones respectively.

REFERENCES

1. D. Pawade, A. Lahigude, D. Reja, "Review Report On Security Breaches Using Keylogger And Clickjacking," International Journal of Advance Foundation and Research in Computer (IJAFRC), Volume 2, Special Issue (NCRTIT 2015), pp. 55-59, January 2015.
2. http://www.computerworld.com/article/2523151/web-apps/chrome-apes-ie8--adds-clickjacking--xss-defenses
3. http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx
4. https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options
5. http://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=1632&context=etd
6. https://blog.lookout.com/blog/2010/12/09/android-touch-event-hijacking/
7. http://blog.trendmicro.com/trendlabs-security-intelligence/tapjacking-an-untapped-threat-in-android/
8. 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.
9. L.S. Huang, A. Moshchuk, H. J. Wang, S. Schechter, and C. Jackson, "Clickjacking: attacks and defenses," in Proc. of USENIX Security Symposium, pp. 22–22,2012.
10. 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, pp. 135-144 , 2010.
11. http://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=1632&context=etd