ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

Implementation of an Extension for Browser to detect vulnerable elements on Web Pages and avoid Clickjacking

K. Sivaraman*, V. Khanna
Department of CSE, Bharath University, Chennai, India.
***Corresponding author:E-Mail: sivaraman2006@gmail.com**

ABSTRACT

This paper address to the problem of web security and provide its solution to enjoy secure web browsing facility. We have developed the browser extension "ClickProtect" which provide a generic solution to multiple Clickjacking attacks. ClickProtect will run on the browser while user is surfing on the internet. It is envisioned to detect clickjacking attacks by warning the user before proceeding with the unsafe actions. Here along with pop-up warning message the unsafe, hidden web components are made visible. This will help to reduce the chances of user information getting stolen, and thereby lead to secure browsing.

KEY WORDS: i-frame; fake cursor; google chrome; extension

1. INTRODUCTION

Since past few years, Internet has become latest trend. Using Internet one can do anything right from entertainment, business, information exchange, shopping, banking and so on. Together with the growth of Internet usage, security concerns are also increased. Cyber-attacks are increasing day by day. Thus web security is always a challenging issue for researchers. Web security is two-step process:

To find data piercing on web page

Provide solution.

In this paper, we have considered web threats under clickjacking category. In simple words clickjacking means pocketing user's mouse clicks. This attack was presented in 2008 by Robert Hansen and Jeremiah Grossman. After that its many variants were developed and solutions are suggested. Most common defense is to restrict browser by not allowing i-frame feature. This is known as frame busting. But this will not work if web page has any third party widget. Balduzzi (2010), have developed "ClickIDS" browser plug-in that intercepts the mouse click events and checks the interactions with the elements of a web page. Also introduced "NoScript" as a Firefox add-on that provides protection cross-site scripting. Huang (2012), have discussed many clickjacing techniques and their solutions in literature review part. They have also implemented three new attacks viz. cursor spoofing, whack-a-mole attack and double click attack and proposed InContext, a web browser or OS mechanism to ensure visual integrity and temporal integrity. Along with this vast research has been done to provide solution in the form of plug-in, add-on, framework and so on. But still there is a threat of being clickjacked. Thus, we tried to attempt this challenge and proposed an Extension for browser which will detect vulnerable element on the web page and notify user regarding that. This extension called "ClickProtect" is designed and tested for Google Chrome browser. The paper is organized as follows: In section 2, we briefly discuss the various attacks under consideration. Section 3 presents proposed system's implementation overview. Section 4 consists of result. Finally, conclusion is drawn in section 5.

Attacks under consideration: We have considered two most common types of attack:

Embedding i-frames

Using fake cursor

These attacks fall under "clickjacking attack" where user is fooled to click unintended page.

Embedding i-frames: It is most common type of adding malicious element. In this attack i-frame is embedded on the web page and made completely invisible. When the user clicks on element, he actually clicks on something else which he didn't intend to click on. There are various versions of this attack. The most common is "Likejacking". On social networking site like facebook, there is a "Like" button. It has been observed that many people often click on Like button to acknowledge the posts of their friends. In Likejacking i-frame is embedded skillfully below Like button. So whenever like is made should result in unintended action. Another common example of embedding i-frame is double click attack. Here after first click, embedded i-frame become top and on second click the invisible button got hit. The only solution to this attack is making hidden element visible. Fig. 1 shows original web page. When user click on "Next slide" button, the expected page to be opened is given in fig, 2. Fig. 3 represent the use of i-frame (red colour) to overlay original button (green colour). When this invisible button is overlaid on original page in fig. 1 and clicked instead of page in fig. 2 page in fig. 4 is opened which is a malicious one and user is knowingly force to visit.

Figure.1. Original Page

Figure.2. Expected Next Page

Figure.3. Overlaid Invisible button

Using fake cursor: Here, a fake cursor along with the real one is made available on the web page or original cursor is replaced by fake one or it is customized. This technique is used to deceive the user. In fig. 5 the hand cursor is a fake cursor. That fake cursor seems to be pointing on "YES" button but this is not a real pointer position. Actual pointer position is shown using arrow.

Figure.4. Unintended Page to which user is directed by embedding i-frame

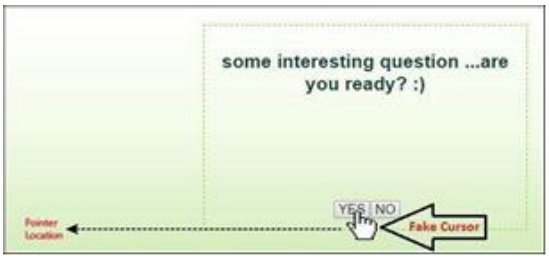


Figure.5. Attack Using Fake Cursor

Implementation overview: Along with studying and implementing variations of clickjacking attack, we present a generic solution to it. The proposed system is based on two major components.

The revealing unit: will get the source code of a webpage and scan it for vulnerable elements such as buttons and anchor tags (hyperlinks). After that it will check its visibility properties and cursor.

The extenuation unit: will notify user about the detected attack and will take appropriate actions to prevent user from performing unsafe actions by providing warning message pop-up to users who can then choose to continue or not. Fig. 6 shows the flowchart demonstrating working of the proposed system.

Attack implementation:

To embed iframe following pseudo code is used. //i-frame style with zero opacity

```
iframe {
width:615px;height:80px;margin: auto; position:absolute; top:180px; left: 370px; filter:alpha(opacity=50);
opacity:0.0; padding-top: 245px; z-index: 1000;35}
//Embedding iframe with target page address
<iframe src="ifA.php"></iframe>
//vulnerable target button
<a href="origA1.php" style="position:relative;left:20px;" id="btn">Next Slide &gt&gt</a>
```

Here, i-frame is embedded in the UI and it is made completely invisible by keeping the opacity zero. So when user will click on the next slide button he will actually click on the i-frame which may lead him to malicious page.

Solution Implementation: Following steps are followed for solution implementation:

1. Initialize nodeIdCount and flag to zero
2. Initialize nodeIdPrefix to '_no-clickjacking-'
3. Set timeout to function detect with 1000ms time interval
4. Extract elements from the webpage and Store clickable elements in an array (iframes)
5. visibility check for each clickable element
   - 5.1 For each element in array iframes get css information
   - 5.2 If opacity is less than 0.1 then store the id of that element to array nodeID
   - 5.3 If id is not assigned then create and assign node id with help of nodeIdPrefix and nodeIdCount
   - 5.4 Increment nodeIdCount
   - 5.5 Add these node ids to array transparentNodeIds
6. Change style for each detected element
   - 6.1 if transparentNodeIds is not empty set flag to 1
   - 6.2 for each element in array transparentNodeIds assign css to make element visible and store in array of arrays named css
   - 6.3 css += '#' + nodeId + '{opacity:1 !important;overflow: visible !important; }';
7. Embedding modified style to web page

7.1 if array css is not empty create element ‘style’
7.2 set innerText to css[i]
7.3 append style tag in head tag
7.4 end detection

8. Notifying user
   8.1 If flag equals 1 then notify user with proper message
9. Detection of fake cursor
   9.1 Get cursor properties
   9.2 if cursor style is none notify user with proper message
   9.3 create element ‘style’
   9.4 set innerText to 'body {cursor:default !important;}'

Thus ClickProtect will first extract the clickable elements and store it in an array. For extracted elements it will check for the visibility whether its opacity is less than 1.If found, it will make those elements visible and will notify the user. Simultaneously ClickProtect will check for cursor properties. If cursor style is set to none, ClickProtect will notify the user that cursor is customized and will the make original cursor visible

Extension: Steps for loading extension

1. Open Google chrome.
2. Click on the ‘customize and control Google chrome’ icon.
3. Go to more tools.
4. Click on extension.
5. Select “Developer mode” (Fig. 7).
6. Click “Load unpacked extension”(Fig. 7)
7. Select folder containing the source code of an extension. Enable the extension (Fig. 8).

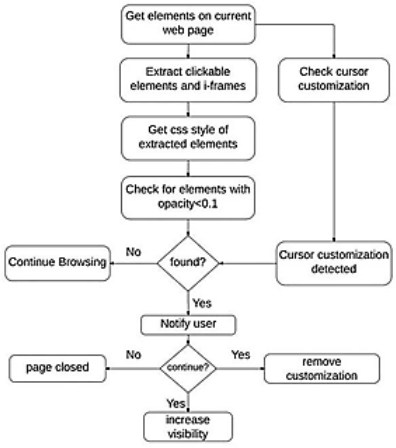


Figure.6. Flowchart demonstrating working of ClickProtect

Figure.7. Extension Loading

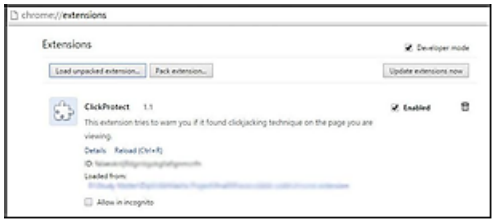


Figure.8. Enabling Extension

2. RESULT

First step of clickProtect is to find out all clickable elements in web page.

Figure.9. List of Clickable Elements

Fig. 9 represents log of all clickable entities. One clickable elements are listed out the next step is their visibility check. For that properties of clickable elements are inspected to separate out invisible elements. Fig 10 shows log of invisible element.

Figure.10. Invisible Element Found

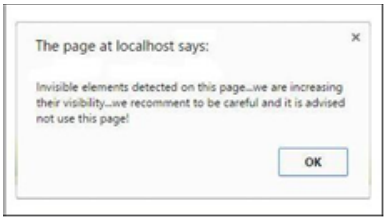


Figure.11. Invisible Element found Warning Message

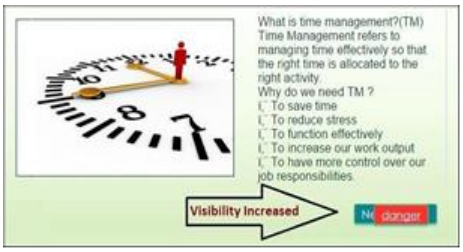
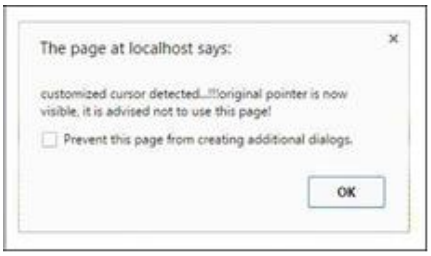


Figure.12. Solution on i-frame by increasing Visibility



Figurer.13. Notification for fake Cursor

After loading and enabling the ClickProtect extension; if user came across any web page embedded with invisible i-frame then he will be notified with pop-up warning message shown in fig. 11. Also the invisible elements are made visible and shown in fig. 12. Fig. 13 and 14 demonstrate solution for fake cursor attack. Here also warning message is provided and original cursor with target link is made available.

Table.1. Related work

Solution	Description

Frame Busting	The code snippet below check if the webpage is the topmost window or not. If it is not, it will makes itself as the topmost frame. This is attained with the help of top property which defines the topmost ancestor window.
	if(top.location != self.location) {
	top.location = self.location;
	}

X-Frame-Options	• Introduced by Internet Explorer 8, • doesn't rely on javascript to be executed • Site owner has privilege to allow/disallow frames • In HTTP header if X-FRAME-OPTIONS: SAMEORIGIN ---- can only be framed from same domain X-FRAME-OPTIONS: DENY---- framing not allowed
Content Security Policy	• Introduce by Mozilla • more general than X-FRAME-OPTIONS, • allow the website owner to enforce every type of content interaction
NoScript	• Mozilla Firefox add-on • Protection against invisible IFRAME
ClearClick	• Updated version of NoScript • Address to cross –origin window interaction

3. CONCLUSION

This paper presents JavaScript based extension called “ClickProtect” for google chrome browser to provide generic solution to clickjacking attack. The implementation steps are follows:

- Extracts web elements
- Filters clickable elements
- Find out vulnerable invisible elements

Following is the set of actions performed by ClickProtect after detection of attack:

For embedded i-frame: increase the visibility of the detected invisible i-frame or other clickable elements so that user will now know where exactly he/she clicking on.

For usage of fake cursor: after attack detection original pointer is made visible to make user aware.

REFERENCES

Achudhan M, Prem Jayakumar M, Mathematical modeling and control of an electrically-heated catalyst, International Journal of Applied Engineering Research, 9 (23), 2014, 23013.

Balduzzi M, Egele M, Kirda E, Balzarotti D, and Kruegel C, A solution for the automated detection of clickjacking attacks, In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, 2010, 135-144.

Rydstedt G, Bursztein E, Boneh D and Jackson C, Busting frame busting: a study of clickjacking vulnerabilities, at popular sites. In Proceedings of the Web 2.0 Security and Privacy, 2010.

Gopalakrishnan K, Sundeep Aanand J, Udayakumar R, Electrical properties of doped azopolyester, Middle - East Journal of Scientific Research, 20 (11), 2014, 1402-1412.

Gopinath S, Sundararaj M, Elangovan S, Rathakrishnan E, Mixing characteristics of elliptical and rectangular subsonic jets with swirling co-flow, International Journal of Turbo and Jet Engines, 32 (1), 2015, 73-83.

Huang L.S, Moshchuk A, Wang H.J, Schechter S & Jackson C, Clickjacking, Attacks and Defenses, Usenix Security Symposium, 2012.

Ilayaraja K, Ambica A, Spatial distribution of groundwater quality between injambakkam-thiruvanmyiur areas, south east coast of India, Nature Environment and Pollution Technology, 14 (4), 2015, 771-776.

Kerana Hanirex D, Kaliyamurthie KP, Kumaravel A, Analysis of improved tdtr algorithm for mining frequent itemsets using dengue virus type 1 dataset: A combined approach, International Journal of Pharma and Bio Sciences, 6 (2), 2015, 288-295.

Lingeswaran K, Prasad Karamcheti SS, Gopikrishnan M, Ramu G, Preparation and characterization of chemical bath deposited cds thin film for solar cell, Middle - East Journal of Scientific Research, 20 (7), 2014, 812-814.

Premkumar S, Ramu G, Gunasekaran S, Baskar D, Solar industrial process heating associated with thermal energy storage for feed water heating, Middle - East Journal of Scientific Research, 20 (11), 2014, 1686-1688.

Sundar Raj M, Saravanan T, Srinivasan V, Design of silicon-carbide based cascaded multilevel inverter, Middle - East Journal of Scientific Research, 20 (12), 2014, 1785-1791.

Thooyamani KP, Khanaa V, Udayakumar R, Application of pattern recognition for farsi license plate recognition, Middle - East Journal of Scientific Research, 18 (12), 2013, 1768-1774.

Thooyamani KP, Khanaa V, Udayakumar R, Efficiently measuring denial of service attacks using appropriate metrics, Middle - East Journal of Scientific Research, 20 (12), 2014, 2464-2470.

Thooyamani KP, Khanaa V, Udayakumar R, Partial encryption and partial inference control based disclosure in effective cost cloud, Middle - East Journal of Scientific Research, 20 (12), 2014, 2456-2459.

Thooyamani KP, Khanaa V, Udayakumar R, Using integrated circuits with low power multi bit flip-flops in different approch, Middle - East Journal of Scientific Research, 20 (12), 2014, 2586-2593.

Thooyamani KP, Khanaa V, Udayakumar R, Virtual instrumentation based process of agriculture by automation, Middle - East Journal of Scientific Research, 20 (12), 2014, 2604-2612.

Thooyamani KP, Khanaa V, Udayakumar R, Wide area wireless networks-IETF, Middle - East Journal of Scientific Research, 20 (12), 2014, 2042-2046.

Udayakumar R, Kaliyamurthie KP, Khanaa, Thooyamani KP, Data mining a boon: Predictive system for university topper women in academia, World Applied Sciences Journal, 29 (14), 2014, 86-90.