Request and Response Analysis Framework for Mitigating Clickjacking Attacks

Hossain Shahriar, Department of Information Technology, Kennesaw State University, Marietta, GA, USA

Hisham Haddad, Department of Computer Science, Kennesaw State University, Marietta, GA, USA

Vamshee Krishna Devendran, Department of Information Technology, Kennesaw State University, Marietta, Georgia, USA

ABSTRACT

This paper addresses the detection of clickjacking attacks, which is an emerging web application security issue. The authors propose a web application request and response page analysis framework to detect clickjacking attacks. Their framework considers not only inspects visual features related to frame, JavaScript code pattern in details to match with known attack signatures. The proposed approach is able to detect advanced clickjacking attacks such as cursorjacking, double click, and history object-based attacks. The authors evaluate the proposed approach with a set of legitimate and malicious websites. The results indicate that their approach has low false positive and false negative rates. The overhead imposed by the proposed approach is negligible.

Keywords: Clickjacking, Client-side Framework, Double Clickjacking, Framebusting, iframe, UI Redressing, X-Frame-Options

INTRODUCTION

Clickjacking attacks allure users to click on objects transparently placed in malicious web pages. The resultant actions of the click may cause unwanted operations in the legitimate websites without the knowledge of the users (Hansen, 2008; Hansen & Grossman, 2008; OWASP, 2015; Aun, 2015). Many recent news reports suggest that victims are tricked to click on social media websites (Facebook, Twitter) (Balduzzi et al., 2010; Huang et al., 2012), shopping websites (Amazon, a victim ends up buying a book), and online banking web sites (Balduzzi et al., 2010). The consequence of attacks can affect victim’s security and privacy. For example, clickjacking attack has been used to enable the webcam and microphone of a victim’s computer (Aboukhadijeh 2011;
Aharonovsky 2008). Other reported incidents include liking a profile in Facebook victim not familiar with, and posting messages on Twitter, etc. (Balduzzi et al., 2010; Huang et al., 2012). Given that clickjacking needs to be addressed to stop much of these unwanted consequences.

A number of defense techniques have emerged in the literature against clickjacking attacks (Balduzzi et al., 2010; Huang et al., 2012; Bordi, 2015; Rydstedt et al., 2010). The widely deployed defense techniques is the inclusion of frame busting code (a small piece of JavaScript code) (Hansen & Grossman, 2008; Rydstedt et al., 2010) to disallow the rendering of a legitimate web page in an iframe. Unfortunately, the approach relies on the successful execution of JavaScript code and there are well known advanced attack techniques that can circumvent the execution of JavaScript. Another technique relies on sending a special HTTP header (e.g., X-Frame-Options supported by most browsers including Firefox (RFC7034, 2013)) to disallow rendering pages in iframes at the browsers. However, proxies located between the client and server may strip these special headers. The older versions of web browsers do not support special HTTP headers. Similarly, Noscript (2015) is a browser side add-ons that let users decide if JavaScript code should be executing or not. Such approach although helps to combat, poses the decision making capability on the user end where a user may not be familiar with the technical details of script code execution. Thus, there is a necessity to develop novel mitigation techniques and address the limitations of existing techniques.

In this paper, we propose a clickjacking attack detection framework (at the proxy level located at the client-side browser) by analyzing requests and responses from websites. The framework is a conceptual browser located at the client-side that can intercept incoming requests and analyze response pages. We analyze the parameter values of request pages and HTML JavaScript code of response pages and perform systematic checking to decide if attack symptoms are present or not. Our approach can remove malicious content related to clickjacking attacks. Moreover, the framework can be extended to detect new clickjacking attacks based on bypassing framebusting solution and advanced attack types including cursorjacking, double clickjacking, and history object-based clickjacking. We implement a prototype of the proposed framework. The prototype is tested with a set of legitimate and malicious websites. The evaluation results indicate that our approach detects well known clickjacking attacks and shows low false positive rate. It also incurs negligible performance overhead and does not break up the source code of legacy web applications. Further, it does not depend on specific HTTP header or enabling/disabling of JavaScript.

This paper is organized as follows: The next section presents illustrative example of clickjacking attack; we then highlight defense techniques found in the literature; the next section outlines advanced clickjacking attacks; then we present the proposed framework followed by evaluation results; and finally we conclude the paper.

ILLUSTRATIVE EXAMPLE OF CLICKJACKING ATTACK

We illustrate an example code snippet mimicking a clickjacking attack based on Facebook “Like” action. Here, the GUI element for “Like” is being hidden in an iframe of a web page controlled by an attacker. The attacker hides the GUI element by making the iframe invisible based on Cascading Style Sheet (CSS) features.

In Figure 1, a div tag named icontainer has a lower CSS z-index with an opacity level of zero. The div includes iframe fbframe that loads the clickable object “Like” from Facebook for an individual’s profile (xyz). The URL represents a clickable “Like” action element which can be embedded in any web page and if the element is clicked then the profile of xyz is being liked by the person who clicked it. Since the visibility of the div is zero, the loaded iframe will not be
Extracting Ontology Properties from the Web-Tables
Song-il Cha and Z. M. Ma (2012). *International Journal of Systems and Service-Oriented Engineering* (pp. 64-77).
www.igi-global.com/article/extracting-ontology-properties-from-the-web-tables/79239?camid=4v1a

Evaluating the Consistency between Navigation and Data Models for Web Applications
www.igi-global.com/article/evaluating-consistency-between-navigation-data/45926?camid=4v1a