IFrandbox - Client Side Protection from Malicious Injected Iframes

Tanusha S. Nadkarni, Radhesh Mohandas, and Alwyn R. Pais

Information Security Research Lab,
Department of Computer Science and Engineering,
National Institute of Technology Karnataka, Surathkal, India
{tanushanadkarni,radhesh,alwyn.pais}@gmail.com

Abstract. Drive-by downloads are currently one of the most popular methods of malware distribution. Widely visited legitimate websites are infused with invisible or barely visible Iframes pointing to malicious URLs, causing silent download malware on users system. In this paper, we present a client side solution for protection from such malevolent hidden Iframes. We have implemented our solution as an extension to Mozilla Firefox browser. The extension will check every Iframe loaded in the browser for properties emblematic of malicious Iframes such as hidden visibility styles and 0-pixel dimensions. These Iframes are then blocked by using browser content policy mechanism, hence alleviating the possibility of the malicious download taking place.

Keywords: Iframes, Malicious JavaScript, Obfuscation, nsIContentPolicy, Drive-by Downloads, Malware, Iframe Injection Attack, Mozilla Firefox Browser, Extension.

1 Introduction

Malicious code is either hosted directly on rogue web sites or injected using Iframes into legitimate web sites. The victim sites are compromised either partially or completely by cyber criminals to serve the malicious content. Hackers bring into play various social engineering techniques to lure users to visit rogue sites. But this requires attackers to trick users to open doors for malware to descend on their systems. To broaden their techniques of distributing malware hackers continuously work on inventing ways of downloading malware without users consent. Most Drive-by downloads happen when a user visits a website that contains a malicious Iframe. The Iframe may load the exploit script, or redirect to another malevolent site[1]. This script targets vulnerabilities in the browser or one of its plugins, successful exploitation of which results in the automatic execution of the spiteful code, triggering a drive-by download. The downloaded executable is then automatically installed and started on the infected system. Attackers use a number of obfuscation techniques to evade detection and complicate forensic analysis. The beginning point of a drive-by download attack is an Iframe, hence the main focus of our work is to utilize the power provided

S. Jajodia and C. Mazumdar (Eds.): ICISS 2011, LNCS 7093, pp. 383–386, 2011.

to browser extensions to identify potentially harmful Iframes and prevent them from loading in the browser.

2 Related Work

NoScript [8] an extension for Mozilla's Firefox browser, selectively manages Iframes. Iframes are blocked based on same origin policy and list of trusted sites. In our extension, we look for different properties of Iframes such as dimensions and visibility properties before blocking them.

3 Iframe Injection Attack

The key problem addressed in this paper is protection from malicious Iframes injected in legitimate websites, which are invisible to the visitors of the website. The <Iframe> tag is an HTML element that contains another document. A hidden Iframe is an Iframe which is present in the HTML of a webpage, but is not visible in the browser. Iframe Injection Attack threat injects an invisible Iframe into legitimate websites [2]. The Iframe source will be a malicious URL hosting an exploit or containing code which redirects to a malicious page. Malicious Iframes are made invisible by setting their attributes appropriately, and are usually injected in the footer or header. Invisible Iframes allow silently loading of malware. To evade detection, obfuscated JavaScript is infused into legitimate sites, which after decoding embeds malevolent Iframe. Many fall victims to such attacks as it is easy to inject such malicious Iframe into a legitimate webpages if the hosting server or FTP accounts are compromised or using Cross Site Scripting and SQL Injection attacks.

3.1 Techniques for Hiding Iframe

Iframe are rectangular elements and they occupy some space on web pages. Hackers use several techniques [3], such as dimension tricks, and visibility styles, to make them invisible for the compromised website visitor. Initially, either or both of the dimensions of the Iframes were set to 0. Since scanners searching for zeros, hackers started to use Iframes with very small dimensions, which makes an Iframe appear like a dot. To by pass scanners looking for dimensions, techniques such as setting visibility to 'hidden', or display to 'none', made the Iframes completely unseen, irrespective of their dimensions. The trick was to place visible Iframes within any parent node with hidden styles e.g. invisible <DIV> element. This hid Iframes despite not containing any code which made them invisible. A real example: <Iframe src="hxxp://google-analyz .cn/ count.php?o=1" width=0 height=0 style="visibility: hidden"> </Iframe>. JavaScript onload trick, is another technique, in which, hackers inject Iframe that does not have a src, style and dimension parameters that can make Iframe invisible. A script is specified for the "onload" event of Iframe which assigns