Securing Legacy Firefox Extensions with SENTINEL

Kaan Onarlioglu¹, Mustafa Battal², William Robertson¹, and Engin Kirda¹

¹ Northeastern University, Boston
{onarliog,wkr,ek}@ccs.neu.edu
² Bilkent University, Ankara
mustafa.battal@cs.bilkent.edu.tr

Abstract. A poorly designed web browser extension with a security vulnerability may expose the whole system to an attacker. Therefore, attacks directed at “benign-but-buggy” extensions, as well as extensions that have been written with malicious intents pose significant security threats to a system running such components. Recent studies have indeed shown that many Firefox extensions are over-privileged, making them attractive attack targets. Unfortunately, users currently do not have many options when it comes to protecting themselves from extensions that may potentially be malicious. Once installed and executed, the extension needs to be trusted. This paper introduces SENTINEL, a policy enforcer for the Firefox browser that gives fine-grained control to the user over the actions of existing JavaScript Firefox extensions. The user is able to define policies (or use predefined ones) and block common attacks such as data exfiltration, remote code execution, saved password theft, and preference modification. Our evaluation of SENTINEL shows that our prototype implementation can effectively prevent concrete, real-world Firefox extension attacks without a detrimental impact on users’ browsing experience.

Keywords: Web browser security, browser extensions.

1 Introduction

A browser extension (sometimes also called an add-on) is a useful software component that extends the functionality of a web browser in some way. Popular browsers such as Internet Explorer, Firefox, and Chrome have thousands of extensions that are available to their users. Such extensions typically enhance the browsing experience, and often provide extra functionality that is not available in the browser (e.g., video extractors, thumbnail generators, advanced automated form fillers, etc.). Clearly, availability of convenient browser extensions may even influence how popular a browser is. However, unfortunately, extensions may also be misused by attackers to launch privacy and security attacks against users.

A poorly designed extension with a security vulnerability may expose the whole system to an attacker. Therefore, attacks directed at “benign-but-buggy”
extensions, as well as extensions that have been written with malicious intents pose significant security threats to a system running such a component. In fact, recent studies have shown that many Firefox extensions are over-privileged \[4\], and that they demonstrate insecure programming practices that may make them vulnerable to exploits \[2\]. While many solutions have been proposed for common web security problems (e.g., SQL injection, cross-site scripting, cross-site request forgery, logic flaws, client-side vulnerabilities, etc.), in comparison, solutions that specifically aim to mitigate browser extension-related attacks have received less attention.

Specifically, in the case of Firefox, the Mozilla Platform provides browser extensions with a rich API through XPCOM (Cross Platform Component Object Model) \[20\]. XPCOM is a framework that allows for platform-independent development of components, each defining a set of interfaces that offer various services to applications. Firefox extensions, mostly written in JavaScript, can interoperate with XPCOM via a technology called XPConnect. This grants them powerful capabilities such as access to the filesystem, network and stored passwords. Extensions access the XPCOM interfaces with the full privileges of the browser; in addition, the browser does not impose any restrictions on the set of XPCOM interfaces that an extension can use. As a result, extensions can potentially access and misuse sensitive system resources.

In order to address these problems, Mozilla has been developing an alternate Firefox extension development framework, called the Add-on SDK under the Jetpack Project \[21\]. Extensions developed using this new SDK benefit from improved security mechanisms such as fine-controlled access to XPCOM components, and isolation between different framework modules. Although this approach is effective at correcting some of the core problems associated with the security model of Firefox extensions, the Add-on SDK is not easily applicable to existing extensions (i.e., it requires extension developers to port their software to the new SDK), and it has not been widely adopted yet. In fact, we analyzed the top 1000 Firefox extensions and discovered that only 3.4% of them utilize the Jetpack approach, while the remaining 96.6% remains affected by the aforementioned security threats.

Hence, unfortunately, a user currently does not have many options when it comes to protecting herself from legacy extensions that may contain malicious functionality, or that have vulnerabilities that can be exploited by an attacker.

In this paper, we present Sentinel, a policy enforcer for the Firefox browser that gives fine-grained control to the user over the actions of legacy JavaScript extensions. In other words, the user is able to define detailed policies (or use predefined ones) to block malicious actions, and can prevent common extension attacks such as data exfiltration, remote code execution, saved password theft, and preference modification.

In summary, this paper makes the following contributions:

- We present a novel runtime policy enforcement approach based on user-defined policies to ensure that legacy JavaScript Firefox extensions do not engage in undesired, malicious activity.