Securing Web Applications with Browser Add-ons: an Experience Report

Philippe De Ryck, Lieven Desmet, Frank Piessens, Wouter Joosen
iMinds-DistriNet, KU Leuven, Belgium

With the vast expansion of the web, and the increasing involvement in our everyday lives, web security is as important as ever, illustrated by the frequent reports of security breaches and vulnerabilities, even in mainstream media. Most of the exploited vulnerabilities have been known for some time, have been subject to active research and have been solved on a practical or academic level. Unfortunately, widespread adoption of such solutions is difficult to achieve, either due to an imperfect web developer, or due to the complexity of dealing with legacy code, leaving the user or the user's data vulnerable in the end. Offering a client-side security measure allows the user to protect himself against these vulnerabilities, since these problems are not (immediately) targeted at the server-side.

In our research, we have developed several browser add-ons that increase the user's security at the client-side, without depending on cooperation from the targeted web sites:

- **CsFire** [1, 2] autonomously protects the user against malicious cross-origin requests, such as cross-site request forgery (CSRF). CsFire is publicly available for Firefox and Chrome, and with just under 51,000 downloads, CsFire clearly addresses a need with security-conscious people.
- **Serene** [4] is a Firefox add-on that protects users against session hijacking and session fixation attacks. Serene is available as a research prototype, being the first to illustrate how session fixation attacks can be mitigated at the client-side.
- **TabShots** [3] is a Chrome add-on that protects users against tabnabbing, a particularly sneaky variant of phishing attacks, for which no effective countermeasure existed until now.

From our experience with developing autonomous client-side security measures, we learned several important lessons, useful for anyone looking to develop a client-side security measure, either as an actual product, like CsFire, or as a proof-of-concept for a certain idea or approach, like Serene and TabShots. The contributions we cover in our presentation can be summarized as follows:

1. *Advantages* Browser add-ons have certain advantages over other client-side techniques, such as a proxy. A browser add-on has access to all context information within the browser, including user interaction and special browsing modes (e.g. incognito mode). Since a browser add-on can access requests and responses at the endpoint, it is not limited by secure HTTPS connections. Finally, the APIs offered by the browser enable plenty of functionality, such as inspecting or manipulating pages or browser state, or even taking screenshots of web pages.
2. *Development* Browser add-ons are typically developed in JavaScript, although native code is often supported. Since most browsers run on multiple platforms, JavaScript is the best choice, and suffices in most cases. With careful development, a browser add-on can easily be shared between Firefox and Chrome, as we illustrate with CsFire.
3. *Evaluation* Browser add-ons typically offer additional security for existing sites, requiring them to be compatible with the majority of these sites, to avoid breaking the user experience. We have conducted several automated, large-scale experiments using a browser with the security add-on installed, indicating both the add-ons effectiveness and compatibility.

An automated crawling experiment does have some limitations (e.g. getting past authentication forms), which can be addressed by using test audiences. For example, CsFire second client policy has been extensively tested by a group of 50 people before adding it to the public release.

References

[1] Philippe De Ryck, Lieven Desmet, Thomas Heyman, Frank Piessens, and Wouter Joosen. Csfire: Transparent client-side mitigation of malicious cross-domain requests. *Engineering Secure Software and Systems*, pages 18–34, 2010.

[2] Philippe De Ryck, Lieven Desmet, Wouter Joosen, and Frank Piessens. Automatic and precise client-side protection against csrf attacks. *Computer Security–ESORICS 2011*, pages 100–116, 2011.

[3] Philippe De Ryck, Nick Nikiforakis, Lieven Desmet, and Wouter Joosen. Tabshots: Client-side detection of tabnabbing attacks. *To appear on AsiaCCS 2013*, 2013.

[4] Philippe De Ryck, Nick Nikiforakis, Lieven Desmet, Frank Piessens, and Wouter Joosen. Serene: self-reliant client-side protection against session fixation. In *Distributed Applications and Interoperable Systems*, pages 59–72. Springer, 2012.