Poster: Hide and seek: On the disparity of browser security settings

Alexios Mylonas, Nikolaos Tsalis, Dimitris Gritzalis
Information Security & Critical Infrastructure Protection Laboratory, Athens University of Economics & Business
76 Patission Ave., Athens GR-10434, Greece
{amylonas, ntsalis, dgrit} @aueb.gr

1. INTRODUCTION

When users connect to the Internet, apart from the obvious gains and opportunities (e.g. e-commerce, social medial, etc.), they come across to various threats. The threats range from common client-side attacks (e.g. Cross-Site-Scripting (XSS)) up to more sophisticated ones that target vulnerabilities in the browser or in its third-party software (e.g. plugins).

The majority of Internet users are protected from these threats only by the security controls that are offered by web browsers. This is the case even in critical infrastructures [8-9]. Some users [1] configure them as they see fit, to protect their security and privacy. Hence, providing an easy way to configure them is important. The literature on web security has not adequately covered this aspect of web browser security. Therefore, this work contributes by presenting our preliminary work on the usability of web browsers.

2. APPROACH

Assumptions: The user (Alice) is not security and technically savvy [7]. We also assume that Alice has not changed the default interface of the browser, i.e. the menu bar is not apparent in Firefox, Internet Explorer, Opera, and Safari.[1] It is also assumed that Alice has not added any extension in her browser that enhances its security, such as *NoScript*, *Ghostery*, etc.

Methodology: The analysis' scope includes the latest versions (as of May 2013) of the popular web browsers [6], namely: Chrome (v. 27), Mozilla Firefox (v. 21), Internet Explorer 10, Opera (v. 12.15), and Safari (v. 5.1.7) for Windows 7.[2]

Initially, we enumerated all windows gadgets (widgets) that are included in each browser's graphical interface. We marked the path of every widget that affects settings of the browser's security controls. A path to a control is the sequence of widgets that must be selected to reach it, starting from the 'interface root'. The interface root is the first interface that the browser presents when Alice accesses its settings. We note that more than one path may exist for a control. For example, in Safari Alice may navigate either to *Menu→Block Pop-up Windows* or *Menu→Preferences→Security →Block pop-up windows*, in order to enable/disable pop-ups. Therefore, our analysis includes the minimum paths of security controls, i.e. the ones with the least widget 'hops'. Then, we marked the security controls that reside together in each browser's interface. This offered insights on whether its structure needs to be reorganized, e.g. when the user has to scroll among several security controls. Furthermore, we marked whether security controls appear along with widgets for browser settings that are not security oriented.

The analysis' scope omits keyboard shortcuts (e.g. CTRL+Shift+a in Firefox presents the configuration interface of add-ons and plug-ins), as well as 'hidden menus' (e.g. *about:config*) that can be used to navigate to a security control. These are excluded, as it is hard for an average user to be adequately aware of them.

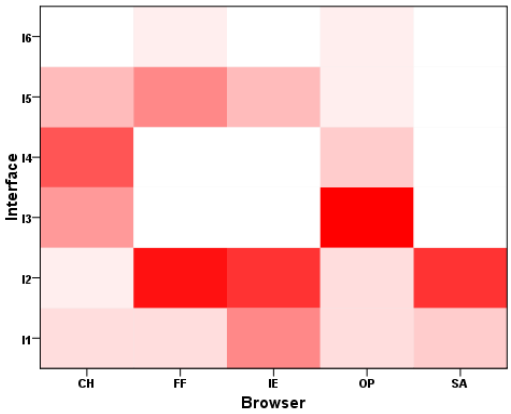

Figure 1: Disparity of security controls in browsers

3. PRELIMINARY RESULTS

After the enumeration of browsers' interfaces, 33 security controls were collected. Due to space limitations, they are given in [4], along with their paths. The majority of the controls' labels are self-explanatory (c.f. [4]), e.g. block cookies, block JavaScript, etc.

The rest of them are briefly described here, namely: (a) c17 refers to the existence of a link towards a web service that analyzes the browser's plugins for vulnerabilities, such as [3] and [5], (b) c19 enables Alice to make a local blacklist/whitelist of web pages and enforce controls on them (e.g. per-site blocking of cookies, JavaScript, etc.), (c) under c23 the browser requests the entry of a master password every time it restarts, before accessing any stored passwords, and (d) c33 enables Alice to manually initiate analysis (for malware/phishing) on the web site she visits.

Our results revealed that in all browsers, the security controls are mixed with browser settings that are not security oriented (such as correction of spelling errors, auto-scrolling, etc.). Moreover, the settings for security controls can be easily altered, without any alert about the significance of this action (e.g. a confirmation prompt, similar to the one of browsing history deletion).

Table 1 summarizes the paths towards interfaces where security controls reside in each browser. Fig. 1 presents a heat map of the number of security controls that reside in these interfaces. Their paths follow a hierarchical structure in most browsers (cf. Table 1), starting from a widget that presents the settings menu. In Mozilla and Opera this widget resides in the browser's upper left corner, whereas in the rest browsers it resides in the upper right corner. However, since users tend to look first into the upper left corner of an interface [2], this will result in usability issues in the early stages of use of the latter browsers.

[1] These browsers may be configured to show a menu bar.
[2] Windows 7 was selected as it appears to be the most popular operating system for the desktop platform [6].

Table 1: Interfaces of security controls (CH: Chrome, FF: Firefox, IE: Internet Explorer, OP: Opera, SA: Safari)

I_i	Path in CH	Path in FF	Path in IE	Path in OP	Path in SA
I1	Menu button	Menu button	Menu button	Menu button	Menu button
I2	l1 →Settings	l1 →Options	l1 →Internet Options	l1 →Settings-Preferences	l1 →Preferences
I3	l2 →Show advanced settings	-	-	l2 →Advanced	-
I4	l3 →Content Settings	-	-	l1 →Page-Developer Tools	-
I5	l1 →Tools	l1 →Add-ons	l1 →Manage add-ons	l1 →Extensions →Manage Extensions	-
I6	-	Search bar →Manage Search Engines	-	Search bar →Manage Search Engines	-

Furthermore, security settings are concentrated differently among interfaces (c.f. Fig.1), as well as reside in different interfaces (c.f. [4]). This may frustrate Alice if she migrates to a different browser, or when she temporarily uses a different browser.

As depicted in Fig. 1, Chrome's security settings are distributed in 5 interfaces. Among them, I3 and I4 contain the majority of them. They appear sequentially in all interfaces, thus, requiring scrolling, which may cause user fatigue and/or frustration. However, Chrome provides a search textbox that locates any widget, thus, allowing Alice to quickly find the widget for a security control.

The majority of the security settings in the rest browsers - in contrast to Chrome - reside in a tabbed interface. This provides enhanced usability, since Alice can access subsets of controls more quickly. Firefox, Internet Explorer and Safari provide dedicated tabbed panes for security and privacy and, thus, the majority of controls reside in I2 (c.f. Fig. 1). In Internet Explorer, however, these two tabs include many security controls in a long list of radio button widgets, which are expected to frustrate Alice. In Opera, on the other hand, most security settings are contained in the *Advanced* tab (I3) (c.f. Fig.1). Thus, more effort is required in order to navigate to them. Security controls in all the above browsers, reside together with other non-security oriented ones, implying that their taxonomy in the tabbed panes is not efficient.

Firefox and Internet Explorer group together the security controls for third-party software (i.e. add-ons and plug-ins) in I5. In contrast, in Safari these controls reside in I2, whereas both Chrome and Opera group plugins in I4 and add-ons in I5 (c.f. [4]). Moreover, Firefox and Opera place c30 in the search bar (I6). On the contrary, Chrome and Internet Explorer place it in I2 and I5 respectively, whereas Safari does not support c30. Therefore, especially in the case of Internet Explorer, where the configuration widget is in an unexpected path (i.e. the same interface with third party software), Alice will find it more difficult to locate c30.

Finally, Chrome and Internet Explorer are the only browsers that do not provide help links in interfaces with browser settings. The rest browsers provide links to appropriate support pages, attempting to aid Alice in configuring the security controls by providing a brief documentation.

4. CONCLUSIONS AND FUTURE WORK

This work provides preliminary results of our on-going work focusing on the usability of security controls that are available in web browsers. Our effort focuses on five popular web browsers, i.e. Chrome, Firefox, Internet Explorer, Opera and Safari.

Our early results indicate that a number of existing usability issues should be addressed:

- The interfaces of the security controls should be reorganized because they mix security settings with those of a different focus.
- Placing together security controls that are conceptually similar will aid, perhaps considerably, users who search for them.
- Users should be alerted each time they alter a browser's security settings (e.g. with a security prompt).

In our next steps we plan to examine the manageability of the security controls. We are also working on an in-depth technical analysis of them.

REFERENCES

[1] Mylonas, A., Kastania, A., Gritzalis, D. 2013. Delegate the smartphone user? Security awareness in smartphone platforms. Computers & Security, 34(0) (May 2013), 47-66.

[2] Eyetrack III - Homepage viewing patterns http://www.poynter.org/extra/eyetrack2004/viewing.htm

[3] Mozilla plugin check and updates http://www.mozilla.org/en-US/plugincheck/

[4] Paths to security controls of web browsers http://www.cis.aueb.gr/public/research/paths.pdf

[5] Qualys BrowserCheck, https://browsercheck.qualys.com/

[6] StatCounter Global Stats, http://gs.statcounter.com

[7] Gritzalis, D., Theoharidou, M., Kalimeri, E. 2005. Towards an interdisciplinary information security education model. In: Proc. of the 4th World Conference on Information Security Education (May 2005), 22-35, Moscow.

[8] Mitrou L., Gritzalis D., Katsikas S. 2002. Revisiting legal and regulatory requirements for secure e-voting. In: Proc. of the 17th IFIP International Information Security Conference (May 2002), 469-480, Kluwer Academics, Egypt.

[9] Iliadis, J., Gritzalis, D., Spinellis, D., Preneel, B., Katsikas, S. 2000. Evaluating certificate status information mechanisms. In: Proc. of the 7th ACM Computer and Communications Security Conference (October 2000), 1-9, Athens.