Defending a Web Browser Against Spying with Browser Helper Objects

Beomsoo Park, Sungjin Hong, Jaewook Oh, and Heejo Lee

Department of Computer Science and Engineering,
Korea University,
Seoul 136-701, Korea

Microsoft's Internet Explorer (IE) is the most widely used web browser, and the IE's global usage is reported as 93.9% share in May 2004 according to OneStat.com. The dominant web browser supports an extensible framework with a Browser Helper Object (BHO), which is a small program that runs automatically everytime starting IE. However, malicious BHOs abuse this feature to manipulate the browser events and gather private information, which are also known as *adwares* or *spywares*

Malicious BHOs have been used mainly for adwares which change the start page of IE or insert ads to web pages. Furthermore, it is possible to gather private information by spying on a web browser with a BHO for logging all inputs typed on the browser. This means that a malicious BHO can capture the passwords behind the asterisks and the credit card numbers copied by cut-and-paste mouse operations; thus, spying with BHOs is more powerful than conventional "keystroke" loggers. Nonetheless, proper countermeasures have not been studied extensively. One trivial defense is to disable BHOs on the browser, but disabling BHOs implies that users cannot make use of normal BHOs such as Google Toolbar.

While there are many detection and protection mechanisms for defending against keyloggers, malicious BHOs are another stream of threats insufficiently handled by anti-keylogging techniques. Therefore, we need to find another way to defend against malicious BHOs, while keeping normal BHOs working for good jobs.

In order to defend against malicious BHOs, we propose a secure automatic sign-in (SAS) architecture. The design goals of SAS are described as follows.

Securing sign-in information on web pages: If some information is entered on a web page through a browser, the information still remains until the page is submitted. And when submitting the page to the web server, subsequent events are incurred by the browser IE. At that time, a BHO can detect the events and obtain the contents of the web page. In order to protect from stealing sign-in information, the one and only way is making sign-in information inaccessible to the web page.

Preventing from keystroke logging: Protection mechanisms for keystroke logging have been proposed in many ways. One obvious way is the use of an alternative input method instead of a keyboard. Virtual on-screen keyboard comes under this category. In order to apply transparently to the protection of malicious BHOs, there are two conditions: 1) no use of keyboard and 2) no modification of web pages. We propose a defensive mechanism using a virtual keyboard in order to prevent from tracking keystrokes.

P. Kantor et al. (Eds.): ISI 2005, LNCS 3495, pp. 638–639, 2005.

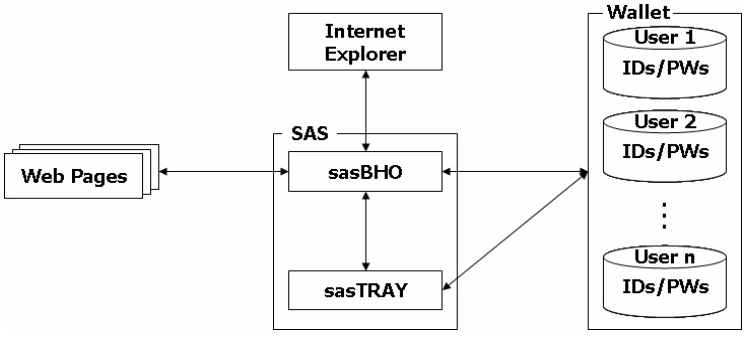

Fig. 1. Secure automatic sign-in (SAS) architecture

The avoidance technique for securing sign-in information works in two phases. First, fill the sign-in form of the current site with a fake information, and submit the page. Next, before the page is sent to the web server, intercept the HTTP request message and replace the fake information with the valid sign-in information stored at Wallet.

The SAS architecture, shown in Fig. 1, consists of three components as follows.

- **sasBHO** : A guardian BHO, called sasBHO, detects a sign-in form if exists in a web page. As well, this BHO program is responsible for invoking a virtual keyboard to register IDs and passwords, and for executing the automatic sign-in procedure using the registered IDs and passwords in Wallet.
- **sasTRAY** : An application program sasTRAY runs separately from IE or BHOs and resides in the system tray. And, sasTRAY is responsible for sustaining the information of master authentication even after terminating IE and sasBHO.
- **Wallet** : The access-controllable storage for maintaining registered IDs and passwords is called Wallet, which stores per-user IDs and passwords for registered sites. The IDs and passwords are stored after being encrypted by using a symmetric-key cryptography.

The proposed SAS architecture is to protect sign-in information from known and unknown malicious BHOs. We have implemented the SAS architecture on Windows systems. The reference implementation shows that the current implementation works properly for 83% of web sites among the most popular 100 web sites. Furthermore, we can increase the effective range by adapting the detection algorithm to other exceptional sites.

Full paper is available online at http://ccs.korea.ac.kr under the “publication” area. This work was supported in part by the ITRC program of the Korea Ministry of Information & Communications. For further correspondence, please contact with Prof. Heejo Lee by email `heejo@korea.ac.kr`.