IDKeeper: A Web Password Manager with Roaming Capability Based on USB Key

Xing Wang, Zhen Han, Dawei Zhang
School of Computer and Information Technology
Beijing JiaoTong University
Beijing, China
10112071@bjtu.edu.cn, zhan@bjtu.edu.cn, dwzhang@bjtu.edu.cn

Abstract — **This paper presents an architecture that affords Web users a trusted ID/Password manager with roaming capability named IDKeeper. The design objectives of this solution are (i) to protect web users from phishing attacks especially malware-based phishing and web spoofing; (ii) to provide an easy-to-use system with roaming capability so that web user can focus on their transaction rather that security and they can use their credential information everywhere. At first, design principles are given based on the analysis of the existing attacks today. Secondly, the solution overview and usage overview are presented. Thirdly, we give the software architecture and the particular implementation. At last, the advantages of IDKeeper are given in conclusion. With IDKeeper, IDs and passwords are stored on users' USB Key, coupling ID/Password to users, not to their machines or to the third-party servers.**

Keywords-component; USB Key, phishing, roaming capability, password protection

I. INTRODUCTION

Web-based interfaces are used in a wide variety of client-server applications because of their intuitive and easy-to-use. More and more people are surfing the Internet for day-to-day activities, from shopping, banking and paying bills to consuming media and entertainment. And most of these applications are security-sensitive[3]. The vulnerabilities of these applications could endanger the privacy of Web users and the integrity of their data. The most severe threat to Web users is phishing attack. During phishing, attackers can use various methods such as malware attack, web spoofing, to steal user's ID and passwords for web application[2,4,9]. We will discuss malware-based phishing and web spoofing which will severely impair the security of web applications.

A. Malware-based phishing

There are many forms of malware-based phishing. Among the most prominent attacks are keylogging and screenlogging. Keylogging is the action of logging the keys struck on a keyboard in a covert manner that the user is unaware that their passwords are being monitored. A screenlogger can monitor both the user's input and the display on screen. Moreover, if the user stores his/her confidential information on the computer, the attacker may directly steal the information with malicious code.

B. Web spoofing

In contrast to malware-based phishing, web spoofing attackers is no longer passively waiting for users to be taken. The attacker creates misleading context in order to trick the victim into making an inappropriate security-relevant decision [2,4]. Mimicking webpages is trivial and users seem to easily fall for it because the average web user is unable to distinguish a legitimate site from a fake one.

C. Unwary web users

Besides the above attacks, we must pay attention to another significant fact: the unwary web users. We can't suppose that the ordinary Web visitors are sophisticated enough, in fact, lots of the Internet users are unwary. Many researchers have demonstrated the feasibility of forging the indicators of trust presented by the browser to spoof the most sophisticated users [4] [10].

In addition, another important feature for a password management method is that user should be able to access his personal information from different computers with the method. Traditional web browser password manager stores users' ID/Password in the registry table (Microsoft IE) or local file (Mozilla Firefox). That is not secure because of malicious software. And user could not access his password which stored in the registry table or local file from another computer.

Several approaches have been proposed to address the mentioned attacks[1,2,5,6,7,8,10]. Ye et al [10] proposed "Synchronized Random Dynamic Boundaries" to defend against visual spoofing. Their basic idea is to distinguish browser-provided status from server-provided content by changing the boundary colors of the browser GUI randomly and unpredictable for remote attacker. But the Internet users must study how to distinguish the spoofed web site and they may be confused because of the protean color. Moreover, the attacker can create a bogus window to overlap the original window to break the defense.

Richard et al [8] proposed a novel architecture to enhance web browsing security. Their approach is achieved by enabling web applications to split their client-side pages across a pair of browsers: one untrusted browser running on a public PC and one trusted browser running on the user's personal mobile [8].

Xmarks which is formerly known as Foxmarks is a famous browser extension which provides synchronization

978-0-7695-4792-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ICICEE.2012.326

and backup services by securely storing the users' data in their Xmarks web account. The Users can access their bookmarks and web site passwords from any browser or mobile phone by logging in at Xmarks web site. Here the most significant problem is that the user must trust Xmarks. There are clearly issues of privacy in sending passwords to a third party.

The approaches above have their own advantages and disadvantages, and they all did not seem to handle the roaming problem, which we think must be considered carefully. Furthermore, we proposed the following recommendations to prevent phishing attack.

- The password manager should interact with the web site as an agent for the user.
- Visual spoofing is a severe problem. It is significant for secure browser to provide a trusted path between the browser and the users. Given that all the UI on PC may be forged, we think the mobile phone-based approach [8] is instructive.
- A usable design of the User Interface must take into account the ease of use and simplicity of the UI.
- The password manager should make the users' data "mobile" so that the users can access their data from any computer.

In our solution, we will adopt USB Key as a trusted password manager. A browser extension called IDKeeper is implemented based on USB Key in this paper.

The rest of this paper is organized as follows. At first, our design principle and overview of our solution is demonstrated in detail. Secondly, the usage scenarios are discussed. Thirdly, the implementation of our solution is given. At last, we conclude this paper by presenting the advantages of this solution.

II. OVERVIEW OF OUR SOLUTION

In this section we describe the security objectives and our design principles, and give an overview of IDKeeper architecture.

A. Solution objectives and design principles

The objectives of our solution are:

- To protect web users from phishing attacks especially malware-based phishing and web spoofing.
- To provide an easy-to-use system with roaming capability so that web user can focus on their transaction rather that security and they can use their credential information everywhere.

And we proposed three principles of web user password management system design:

1) PC is not trusted. The most important information (e.g. user's password) should be stored in the equipment more safely, rather than the untrusted terminal.

2) User's web password should be coupled to user, not to machines. (namely roaming capability, by roadming we refer to the ability of a user to the same ID/Password and credentials across different terminals).

3) Easy-to-use. The security software should be as much as possible to simplify the user's operation, so as to reduce the risk and be more easy-to-use.

B. Solution Overview

Fig. 1 shows an overview of our architecture. The web browser connects to the web server over the Internet, using HTTP to request web pages. The trusted USB Key connects to the PC directly and communicates with the browser using browser extension technology (e.g. Mozilla Firefox extension). We adopt USB Key as a trusted user ID/Password manager for web application.

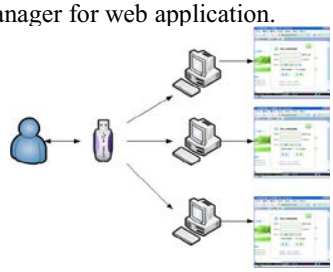

Figure 1. Solution Overview

USB Key is a kind of tamper-resistant device that stores the credential information of the user. And the hardware platform of USB Key consists of a security chip and a flash disk. The core software will run on the security chip. The flash disk will store programs (a Firefox Extension) running in PC OS so that users need not to install any additional software.

As shown in Fig. 1, with the USB Key, user's login information is coupled to the user himself. With this design, the user can carry the USB Key and access the web site easily and securely. The USB Key is a secure, mobile storage medium for managing user's personal information.

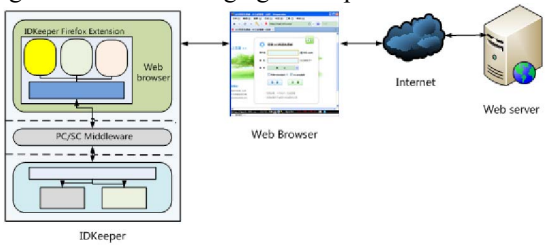

Figure 2. IDKeeper Architecture

The detailed architecture is given in Fig.2. As shown, IDKeeper is implemented as a standard Mozilla Firefox browser extension working together with a USB Key. The browser extension communicates with USB Key via PC/SC middleware. The browser extension can work normally if and only if the user inserts his USB Key and verifies his PIN to the USB Key successfully. This application is a classic two-factor authentication. Based on the user PIN, the extension and USB Key will generate a shared session key respectively. The session key is used to protect the data communicated between the browser extension and USB Key. Because the USB Key is secure and we build a trusted path

between the USB Key and browser extension, so we can say that we create a trusted path from user to the browser.

III. USEAGE OVERVIEW

Firstly, the user must insert USB Key on the machine's USB port and enter his user PIN via the IDKeeper browser extension to login to the USB Key. A Trusted Path from the extension to the USB Key will be established if the PIN authentication is successful.

When the user register a new account at a website, the user enters ID and Password on the website and click "Register", then IDKeeper browser extension will ask the user whether to save the credential information into the USB Key. If it was allowed IDKeeper will parse the web page and store the information of the web site contains the Site Name, the URL of the site and the corresponding User ID and Password to the USB Key via the Trusted Path.

To log into a previously registered web site, the user need not enter his ID/Password on the login page. The user just opens the login page and clicks the login button or presses a shortcut key. Then IDKeeper will check the website information and find the corresponding credential to fill in the login form.

Because all of the site information contains user ID/Password are stored in the USB Key, when the user wants to access web site from another computer, he just need to install IDKeeper browser extension and then insert his USB Key and login to it with his user PIN. Then the user could access the web site in security and freely with IDKeeper.

IV. SOFTWARE ARCHITECTURE AND IMPLEMENTATION

Figure 3 shows the architecture of IDKeeper software. IDKeeper consists three parts as follows:

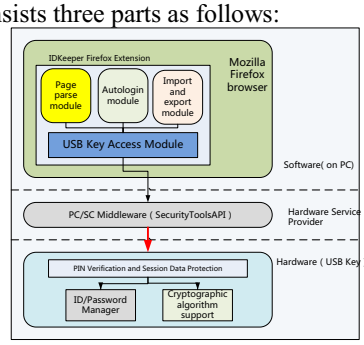

Figure 3. Software Architecture

1) IDKeeper browser extension (IDKeeper-EXT)

We implemented IDKeeper-EXT as a standard Mozilla Firefox browser extension.

As shown in Figure 2, IDKeeper-EXT consists of four modules:

- *PPM*: When the user open a web page contains login form, PPM will parse the page to get the site URL, site name and login form input elements name.
- *ALM*: When the user opens a registered web site, *ALM* will check the site's URL received from *PPM* and find it in the USB Key. Only when the URL is matched completely with one of the stored site information, *ALM* will fill the login form automatically. If the URL is "similar" with one of the stored site URL, *ALM* will give a warning to the user that maybe he is opening an imitative web site. We use Levenshtein distance to measuring the amount of difference between two web site URLs.
- *IEM*: With this module, user can export his/her web site login information encrypted by his/her PIN into a backup file. If the user loses his USB Key, the backup file can be imported to a new USB Key and recover all the data. The person who gets the lost USB Key could not get the inner data because he did not know the user PIN.
- *USB Key-ACM*: In Firefox, we use the Cross-Platform Component Object Model (XPCOM) to access local files and resources. Here AM module controls the access to the USB Key using XPCOM technology. It provides the ability of opening and connecting with the USB Key to the IDKeeper-EXT.

2) PC/SC middleware:

PC/SC is a specification for smart-card integration into computing environments. PC/SC middleware is used to communication between the IDKeeper-EXT and the USB Key. The PC/SC middleware functions as the hardware (USB Key) service provider to the IDKeeper-EXT via APDU messages, providing USB Key access control and ID/Password storage on the USB Key. In addition, the middleware is responsible for establishing the trusted path with the USB Key when the user enters his PIN to login the USB Key. We use a simple key agreement protocol based on the user PIN to create the trusted path.

3) ID/Password Manager on a USB Key:

In contrast to storing user's personal information (e.g. ID and Password) in the PC machine (like Microsoft Internet Explorer), we adopt USB Key as a trusted Password Manager to maintain the user's ID/Password. As mentioned in the first section, PC is not trusted for the malware reason. So we choose the USB Key as a hardware security token, which will be used to store users' personal important information and execute crypto operations. There are three function modules on the USB Key. PIN verification and session data protection module controls the access to the USB Key, this module connects with PC/SC middleware via APDU message to verify user PIN and encrypt the session data. ID/Password manager module stores all the significant data securely in the USB Key. And cryptographic algorithm support module provides the required cryptographic functions.

We implement our ID/Password Manager on a USB Key which supports Java Card technology. Using a Java Card supported USB Key, it is easy for us to develop applications using familiar Java language and programming paradigms.

V. CONCLUSION

The design and implementation of the prototype IDKeeper is given in this paper. We demonstrate the benefits of managing user ID and password with browser extension and USB Key. In contrast to existing solutions, (i) IDKeeper

can protect user's credential information from malware-based phishing attack especially keylogger or screenlogger because IDKeeper will communicate with the web server and users no longer need to enter their password via the keyboard or soft keyboard; and (ii) with IDKeeper, users can carry their personal information on the USB Key so as to login the web site everywhere. ID and password are coupled to users, not to machines. (iii) IDKeeper provides an easy-to-use and uniform web authentication user interface to the users. A prototype system to test and validate the concept was implemented as a Mozilla Firefox extension, based on USB Key and Java Card platform.

REFERENCES

[1] Amir Herzberg and Ahmad Jbara, "Security and Identification Indicators for Browsers against Spoofing and Phishing Attacks," ACM Transactions on Internet Technology(TOIT), New York, NY, USA, vol. 8, pp. 1–36, September 2008.

[2] Andre Adelsbach, Sebastian Gajek and Jörg Schwenk, "Visual Spoofing of SSL Protected Web Sites and Effective Countermeasures," Information Security Practice and Experience, LNCS, 2005, pp. 204-216.

[3] Dawei Zhang and Peng Hu, "Trusted e-commerce user agent based on USB Key," Proceedings of The International MultiConference of Engineers and Computer Scientists, 2008, pp. 800-806.

[4] Edward W. Felten, Dirk Balfanz, Drew Dean and Dan S. Wallach, "Web Spoofing: An Internet Con Game," Technical Report 540-96. Department of Computer Science, Princeton University, 1997.

[5] Haidong Xia and José Carlos Brustoloni, "Hardening Web Browsers Against Man-in-the-Middle And Eavesdropping Attacks," Proceedings of the 14th international conference on World Wide Web, 2005, pp. 489-498.

[6] Li Huang Ng and Daniel TH Tan, "A Novel JavaCard-Based Authentication System for Secured Transactions on the Internet," Proceedings of the 8th IEEE International Conference on Networks, 2000, pp. 262-266.

[7] Min Wu, Robert C. Miller and Greg Little, "Web Wallet: Preventing Phishing Attacks by Revealing User Intentions," Proceedings of the second symposium on Usable privacy and security, 2006, pp. 102-113.

[8] Richard Sharp, Anil Madhavapeddy, Roy Want and Trevor Pering, "Enhancing Web Browser Security on Public Terminals Using Mobile Composition," 2008, pp. 94-105.

[9] Sebastian Gajek, Hans Löhr and Ahmad-Reza Sadeghi, "TruWallet: Trustworthy and Migratable Wallet-Based Web Authentication," Proceedings of the 2009 ACM workshop on Scalable trusted computing, 2009, pp. 19-28.

[10] Zishuang Ye, Sean Smith and Denise Anthony, "Trusted paths for browsers," ACM Transactions on Information and System Security, 2005.3, pp.153-186.