

A mobile single sign on system

Mats Byfuglien

2006/1/6

Abstract

Today users have to manage a set of username, passwords for every service they are using. As the number of passwords to manage grow, people start writing them down, use easily guessable passwords or use the same password on different accounts. This severely reduces the security passwords provide and a better password managements system is needed. Single sign-on (SSO) allows users to store all their passwords securely in one place. Many of the SSO solutions available on the market today are either to expensive for the common user or they lack mobility.

This goal of this thesis is to propose a new mobile single sign-on system that automates logons for the user. The passwords will be stored on a mobile phone and transferred securely to a USB unit plugged into the PC. This unit will be configured as a keyboard and input characters as if the user was typing them on a conventional keyboard. The main contribution from this thesis will be to show whether or not it's possible to implement the proposed idea, and the results will be a prototype of the solution in accompanied by a security and usability analysis.

1 Introduction

1.1 Topic

The topic of this thesis is password management. I will implement a single sign-on solution that makes it possible for users to store their passwords on a mobile device and use this device to perform logins on every computer they use. The thesis will also include a security and usability analysis on the implementation.

1.2 Problem description

People today have a multitude of different accounts, such as web-shops, work accounts, home, ISP, e-mail, instant messaging, on line-banking etc. As a result people have to remember a lot of passwords. This is a problem because people start writing down their passwords or they use the same password on different accounts. Both these approaches severely reduces the security that passwords provide. For this reason there is a need to find a better way to manage passwords, that doesn't require the user to remember passwords or write them down.

1.3 Motivation

If users are to follow common security guidelines such as [12] and [9] he or she should have different passwords for each account. Each password should consist of upper and lower case letters and

preferably also numbers and special characters. Also the password should consist of more than 8 characters. The passwords should be memorized and changed often. In addition the passwords should not be easily guessable such as a word in a dictionary, your zip code, birthday of a family member etc. Most people are not able to follow these guidelines. By introducing single sign-on, some of these problems can be mitigated.

Since the solution proposed by this thesis stores passwords on mobile devices, it will benefit users who are not confined to a single computer. Users who only use a single computer, but has several passwords could also benefit from this solution. Essentially, almost everyone who uses a computer can benefit from this solution.

Finding a better way to manage passwords is the main motivation for this thesis.

1.4 Research questions

The following research questions are defined:

1. Does other alternative SSO solutions exist, if so - how are their security and usability?
2. What methods exist for transferring passwords securely between the mobile device and the computer, and how secure are they?
3. How secure is the prototype developed in this thesis?
4. How user friendly is the prototype developed in this thesis?

2 Related work

2.1 Does other alternative SSO solutions exist, and how are their security and usability?

A taxonomy of single-sign on systems has been proposed [15]. This taxonomy identifies four main categories that SSO products fit into:

- local pseudo SSO
- proxy-based pseudo SSO
- local true SSO
- proxy-based true SSO

Pseudo SSO systems uses a component that automatically executes the authentication mechanisms in place at the different service providers, such as username/passwords, X.509 certificates etc. At the beginning of a session, the user authenticates onces to the pseudo SSO component. For pseudo SSO systems, a separate authentication occurs between the SSO component and the service each time the users requires it. Local pseudo SSO means that the SSO component is stored locally at the users machine. Proxy-based pseudo SSO means that the SSO component is located on an external proxy and authentication is performed between the proxy and the service provider. In the latter case, the local machine never has access to the authentication credentials.

In a true SSO system the user authenticates to an Authentication Service Provider (ASP). The ASP has a relationship with every service provider it provides SSO for. The user authenticates

himself for the ASP once. Service providers are notified of the status of the authentication via authentication assertions. Username and passwords will not be passed to the service provider as is the case with pseudo SSO. A local true SSO system uses a trusted component within the user system which acts as an ASP. In a proxy-based true SSO system, the ASP is located on an external server. This server acts as a broker between the external user and the service provider.

Pseudo based SSO solutions are transparent and doesn't require any changes in the application/service providers. On true SSO solutions, the underlying applications/service providers needs to support the SSO product in use, i.e it's not transparent.

2.1.1 Local pseudo SSO

An example of this kind of single sign-on solution is Password Director [6]. Password Director also provides support for storage of the password database on external devices such as smart cards, USB-sticks etc. This makes it possible to carry your passwords with you at all times, and use the passwords on every computer where this software is installed.

One of the better SSO solutions available for personal use is MobiPassword [1]. This is a program that stores all your passwords in a database and it has the possibility to provide automated logins. Unfortunately, this solution only works on web forms.

2.1.2 Proxy-based pseudo SSO

An example of proxy-based pseudo SSO is web based single sign-on. Products such as Online Password Manager [5] and MyPasswordManager [4] are examples of this. Web based SSO works in almost the same way as SSO products confined to a single computer, the main difference is that the encrypted database with the passwords are available online (on an external proxy server). This makes it possible to access your passwords from every computer connected to the Internet. When using web bases SSO products it's even more important with a secure master password. Since the database is available online, the only thing preventing an attacker from viewing your passwords is the master password. MyPasswordManager [4] also provides a function to automate the login procedure on web forms by using auto complete. The automated login procedure provided here, only works on web forms.

Another proxy-based pseudo SSO solution is proposed by [18]. This scheme is designed to be used in untrusted networks, such as a terminal at an Internet café. Since the user doesn't trust the network, it's not safe to provide passwords over it. Therefore a trusted proxy is introduced. The user authenticates himself to the proxy once. And the proxy, which stores copies of the users access credentials, makes the real authentication to the service provider.

2.1.3 Local true SSO

In [16] an example of a local true SSO solution using a Trusted Computing Platform Alliance (TCPA) compliant platform is described. The scheme requires a trusted component and a public key infrastructure (PKI) to be in place.

2.1.4 Proxy-based true SSO

A popular web based or proxy-based true SSO is Microsoft passport [3]. The users registers with a valid e-mail and a password. Once they have logged in, they have access to all the services on the

Passport network. This is achieved by storing a ticket in the form of an encrypted cookie in the users browser. The functionality is quite similar to Kerberos [2].

A mobile proxy-based true SSO solution is also proposed [17] using GSM/UMTS operators as the authentication service provider (ASP). This is not an implemented solution. It only proposes the SSO protocol to be used. This solution is suppose to authenticate subscribers to service providers without any user interaction in a way that is transparent to the user. The proposed protocol requires some minor changes to the existing GSM infrastructure.

Two very good SSO solutions for the business market are Protocom SecureLogin [7] and Utimaco - Safeguard [8]. Both these systems implements a centralized manager. The user has to authenticate himself to this manager once. After that, the manager automates all logins to services that the user is allowed access to according to his credentials. Companies that implements theses solutions gets an effective access control system in addition to a SSO system.

Another solution proposed is to implement a SSO solution using Kerberos V [14]. The user has to authenticate himself to the key distribution center (KDC) using the strong authentication protocol in Kerberos. When you have authenticated once to the KDC, you are issued a ticket. As long as the ticket is valid (usually about 8 hours), you have access to every service that you are allowed to use. Kerberos also provides access control in addition to SSO.

Finally [19] describes a different approach to SSO. Instead of relying on static tokens such as username and passwords, this scheme introduces dynamic tokens. These tokens can describe data such as payment history etc. Most SSO schemes today requires an extra process when payment is required, such as a pay per view service. The solution proposed in scheme, will incorporate SSO and payment in a single process.

2.2 Methods for transferring passwords securely between a mobile device and a computer

Bluetooth is the most commonly used technology when a mobile device is communicating with a computer. One of the reasons for this is because Bluetooth provides reasonably good security, such as link layer encryption. [13] [11] describes the Bluetooth security architecture and the Bluetooth security respectively. [10] also describes some important issues about Bluetooth security.

Another important part of secure communication between devices is protocol design. In order to prevent man-in-the middle attacks, replay attacks etc, the protocols have to be secure. [20] describes some good principles and protocol examples when it comes to designing SSO solutions.

3 Summary of claimed contributions

My thesis will consist of implementing and analyzing a new single sign-on solution based on a USB/Bluetooth unit and a mobile device, and the main contribution will be to show whether or not it's possible to implement this idea. The solution will make it possible to auto complete login forms on a computer from a mobile device. This will be done by configuring the USB/Bluetooth device as an external keyboard. When the username/password is transferred to the unit, the characters will be received on the computer just as if they where typed on a conventional keyboard by the user. Designing a secure protocol for the units to use during communication will be a critical part of this thesis. After the implementation is complete, a security and usability analysis performed on the prototype. The results from the analysis will provide information which determines whether this was a good idea or not.

References

- [1] Inc com consulting - mobipassword. <http://www.mobipassword.com>.
- [2] Kerberos. <http://web.mit.edu/kerberos/www/>.
- [3] Microsoft passport. <http://www.passport.com>.
- [4] MyPasswordManager. <http://mypasswordmanager.com/index.htm>.
- [5] Online password manager. <http://www.handyarchive.com/Utilities/Password-Recovery/12343-Online-Password-Manager.html>.
- [6] Password director, lastbit software. <http://www.passworddirector.com>.
- [7] Protocom securelogin. http://www.protocom.com/html/securelogin_single_sign_on.html.
- [8] Utimaco - Safeguard. <http://www.utimaco.com/index6main.html>.
- [9] Sonnenbarg Alan. SSO: Enabling an effective password policy, Imprivata inc. 2003.
- [10] Peter Drabwell. Bluetooth security - fact or fiction? 2002.
- [11] Christian Gehrman. Bluetooth security whiteaper, Bluetooth SIG Security Expert Group. 2002.
- [12] Mark O. Kaletka. Easy things users can do to improve security: Recommendations of "best practices" for securing individual user's accounts. <http://security.fnal.gov/UserGuide/password.htm>, 1998.
- [13] Thomas Muller. Bluetooth security architecture version 1.0. 1999.
- [14] Antti T Niemi. Single sign-on with Kerberos V, Helsinki University of technology. <http://www.tkk.fi/cc/docs/kerberos/sso.html>, 2002.
- [15] Mitchell Chris J. Pashalidis Andreas. A taxonomy of single sign-on systems, information security group, Royal Holloway, University of London.
- [16] Mitchell Chris J. Pashalidis Andreas. Single sign-on using trusted platforms. technical report rhul-ma-2003-3, Mathematics Department, Royal Holloway, University of London. 2003.
- [17] Mitchell Chris J. Pashalidis Andreas. Using GSM/UMTS for single sign-on, information security group, Royal Holloway, University of London. 2003.
- [18] Mitchell Chris J. Pashalidis Andreas. A single sign-on system for use from untrusted devices, information security group, Royal Holloway, University of London. 2004.
- [19] Itoh Takayuki Satoh Fumiko. Single sign-on architecture with dynamic tokens, IBM Research, Tokyo Research laboratory, Japan. 2004.
- [20] Chen Kefei Zhao Gang, Zheng Dong. Design of single sign-on. 2004.