User-centered design of secure software

Ursula Holmström
Telecommunication Software and Multimedia Laboratory
Helsinki University of Technology
PO Box 9700, FIN-020150 HUT
phone +358 9 451 5125
fax +358 9 451 5351
email ursula@tcm.hut.fi

Abstract

With the advance of electronic commerce and other networked services there is a growing need for easy to use secure software. The problem with the security features and applications in much of today's software is that it is very technology-oriented. In order to correctly use it a basic knowledge of the underlying technology is often necessary. Our objective is to develop a security concept that supports a user in making educated decisions and managing security issues in everyday networked service access situation.

In this paper we present a user centered approach to the design of security software. We apply user centered design to the development of a security manager concept for a portable computer and communication device. The security manager supports the users in building a security policy and following this policy to form secure connections over an open network.

The development of a user centered security concept for a personal communication device is described. The main focus is on the development of the relevant security concepts for a non-technical user of networked services. An example of how to implement such a concept using public-key infrastructures and digitally signed certificates is also presented, as well as discussion on how this concept can be applied to a more general case of secure access to networked services.

Introduction

With the growth of the Internet, more and more people of different backgrounds and skills are using networked services. Services such as electronic commerce require a variety of security measures when used on an open and insecure network. The result is a growing need for security solutions that are easy to use and understandable to the average web-surfer.

Looking at security from a technical point of view the fundamental objectives of computer security are confidentiality, integrity and availability [1]. Confidentiality requires that data in a computer system or transmitted between systems is revealed only to authorised individuals. In a strict sense it includes not only protection from disclosure of actual data, but also the fact that the data exists. Integrity means protection from unauthorised modification, deletion or creation of data in a computer system or transmitted between systems. Availability, or protection from denial of service attacks, means that resources are available to authorised users when they are needed. In addition to these three fundamental objectives other secondary objectives such as authorised use (or access control), authentication and non-repudiation are often included [2]. To ensure confidentiality and integrity we need some means to tell who is an authorised user. This is often implemented as some sort of access control - only authorised users have access to the data. Authentication means making sure that, for example, a user is who he claims to be while non-repudiation ensures that actions, such as sending a message, can not later be denied.

It is often said that one of the biggest security risks in any reasonably well implemented computer system are the people using it. Technically security issues can be solved with methods such as strong cryptography, digital signatures and secure communication protocol, but this is of no help if the user of the system fails to encrypt a confidential messages or switches to an insecure application because the security software is too confusing. We have found very little published research on the subject of usability and security. In the Adage project [3] [4] usability was an important design objective when designing an interface for professional system administrators. Whitten and Tygar [5] have studied the usability of PGP and also defined some characteristics of the usability and security problem:

- If a secret is revealed, even for a short period of time, there is no way of making sure that it has not been read by an attacker. To avoid possibly high-cost mistakes users must understand their security well enough.
- A network is only as secure as its weakest link. This means that the user must be guided to attend all aspects of their security.
- Security is only a secondary goal. Users can not be expected to put much effort on reading manuals or look for security controls in obscure places.
- Security management usually involves security policies and other abstract constructs that may be alien to many users.
- To prevent errors it is crucial that the user gets sufficient feedback on the security configuration.

The work presented in this paper was done as a part of a telecommunication software security architecture (TeSSA) project[1]. The objectives of the usability study were on the one hand to gather experience on applying user centered design methods to the development of security software and on the other hand to develop a user interface concept for our security architecture. The detailed design and implementation of the actual application have not been

[1] http://www.tcm.hut.fi/Tutkimus/TeSSA/

started as of writing this. For the user interface design we chose a small portable computers as the physical network access device. This could be any device that has the possibility to run some software similar to a web browser, the capability to add new software and a means to access the Internet. In practice this could be the combination of a palmtop computer and a mobile phone or a future generation mobile phone.

Methods and material

In this work our guideline for user centered design has been the ABC of user centered design [6]. Since our main goal was to come up with a good design concept, we concentrated on the A, B and C of the process: "analysis", "begin with objectives" and "conceptual design". For the analysis part we used focus groups and interviews. The objectives of the focus group sessions were to define a user profile and to identify the main use scenarios for a security manager application. The focus group consisted of computer security and usability experts familiar with applications in networked and multimedia communication. After a short introduction presenting the physical device and the overall objectives of the session, the group members were asked to come up with ideas on what this device could be used for. The discussion was not restricted to realistic ideas, but the focus was kept on services and applications and not the physical design of the device. Once the group members got started with the applications and services they were also asked about the likely users for these services.

The ideas produced by the focus group were organised primarily according to what kind of communication would be needed but also with likely security needs in mind. The result was five different communication situations:

- Information retrieval & web browsing. This category includes information services similar to those found on the WWW: timetables, traffic and weather information, news and web-magazines, dictionary, maps, … The services can be freely accessible or they may require registering. The user submits some sort of request for information and gets some amount of data as a response.
- Submitting information. Possible applications include submitting feedback, voting (people, decisions, polls), lottery and electronic visiting cards. Also in this category are services like ticket reservations and other forms of electronic ordering or subscription of products that will paid and delivered by other means.
- Buying and selling. These include electronic money and shopping on the net as well as banking services. A slightly different group of services would be electronic payment - or tokens - for some local service such as vending machines, parking meters or public transport.
- Communication. This could be audio, video, text and multimedia communication between two parties or a group of people.
- Remote access. The remote access category has two main applications: distance working and controlling a "smart home" when away. Distance working could be e.g. accessing a database or checking your mail while visiting a customer. Applications for a smart house include remote control of burglary alarms, lights and (sauna) heating.

Although the original idea was to develop a user interface for a non-technical user and mainly for personal communication it turned out that, at least in the near future, a more likely user group are business users. The need for secure communication is also more prominent in business use where sensitive company information is handled. The likely user is someone

whose work includes travelling, for example visiting customers, and who needs access to e.g. the company intranet while on the move. The user is familiar with using a computer as well as mobile phones to communicate while away from the office. On security issues the user is aware that there are security risks, but not necessarily how these risks affect him(her). He is also not familiar with the security technology.

The main objective of the user interviews was to get an idea of what the users think of information security. More specifically we were interested in what risks the users perceive and what they feel they need to protect as well as how much they want to be involved in making these security decisions. Other objectives were to find out who the user will trust and on what basis and also what kind of concepts and terms they use when talking about network security.

These objectives would best be met by doing a proper user study. Due to lack of resources this had to be postponed to future research projects. Instead freeform interviews were conducted with some students and graduates at the university campus.

After presenting the concept to the interview subjects they were asked what they would use such a device for or what they thought others (e.g. business users) would use them for. In the cases where this seemed like a too difficult question the subjects were asked about their current Internet use. Next they were asked about any possible security risks in the services they (would) use. When needed the interviewer presented some possible threats and asked whether they were relevant in any of the services discussed. During the interview we also asked what the users wanted to specify themselves and what the program should take care of as well as what kind of feedback they wanted to get. The results of the interviews show that:

- The interviewed persons were all aware that there are some security risks in communicating on the Internet, but most felt that there was no threat to them personally as long as they did not make "obvious stupidities" revealing passwords to the whole world.
- Information received through the Internet was not fully trusted. Trust was based more on information from friends and people you trust. Banks and commercial organisations were trusted based on their reputation.
- The user wants to be aware of what the computer does. If secure connections are made automatically the user wants to know that this was done. Many suggested that they want to specify their settings once and then just get feedback on events automatically taken care of.
- No obvious metaphors for a security manager were found. Familiar examples such as viruses or unwanted e-mail (spam) could be used to explain more general security features as access rights for downloaded software or restricting what information is given out about the user and the computer.

The (high level) usability objectives we formulated turned out very similar to those suggested in [5]: to reliably make the user aware of all the security tasks that need to be performed, to enable the user to figure out how to perform those tasks, to prevent the user form making dangerous mistakes and making the user feel comfortable enough with the interface to continue using it.

Results

Based on the use situations identified in the focus group sessions the security needs and tasks were identified. Although the overall need for secure communication varies, the main tasks are more or less the same in all the use situations: defining or modifying the security settings and starting secure communication with a new or previously known party.

If there are only a few parties who require secure communication they can each be handled separately. As the number grows it becomes more practical to handle them as groups, each group having its own security settings. For each group or single party the user has to define: what information is given to members of this group (privacy and confidentiality issues), what access rights do they have to the computer (authorisation) and the need for e.g. authentication, encryption and digital signing. Setting up the groups also includes adding, moving and removing members and changing properties of a group. Other tasks at set-up include entering personal data and contact information as well as making some more general policy decisions such as choosing a default group and specifying a validity time for the settings (when should they be verified again). Much of the security settings are in the end about trust - who do I trust and to what extent. Expressing this trust in an electronic form so that it includes all the "buts" and "and ifs" is more than an average user is willing to do. Making automatic trust decisions on behalf of the user is one potential source of critical errors.

Although some applications, like ordinary web browsing, can be run "insecurely" they still need some basic access control as well as confidentiality and privacy. When a previously unknown site is requesting secure communication the user is faced with a number of security tasks and decisions. The user must consider the information provided about the site and make a decision on what to trust this site with and should it be trusted just once or also in future. In security terms this means defining appropriate levels for authorisation, confidentiality and other group settings and possibly authentication of the requesting party (via a third party). When communicating with a known site that supports secure communication the site needs to be authenticated and the communication encrypted as specified in the settings for that site. In this case the user only needs to confirm that the settings are still valid. (In some cases it may be the user who requests secure communication. The tasks are the same as above, but now the user has to set the requirements instead of just approving or discarding them.)

For a user unfamiliar with security issues certificates, security policies or access control lists are all too unfamiliar. Still these are the terms used in most current applications. Trying to come up with an alternative solution gave us a good idea why this is the case - presenting security in a user friendly but still accurate way is a real challenge!

One metaphor that seems to cover most areas is a "secure electronic business card". The information you put on the card determines what others will know about you (confidentiality) but it can also be seen from an access control point of view: the contact information given (phone number, e-mail, bank account number…) determines what authorities you give to the receiver. Receiving a card does not necessary mean that the information on it has to be trusted in any way. I may want to see some sort of identification to confirm that the this person really is who (s)he claims to be. If I decide to trust the person and card, I put the card in the folder for trusted cards and maybe choose to reveal more about my self by giving this person my card for trusted persons. A possible problem with the card metaphor is that paper business cards are given to the receiver and thus you can not take away some right that you have once given. In a security setting revocation of rights has to be possible. If on the other hand we think of the information on the card as information that is necessary but not as such enough for access then granting access is in the end up to the current security settings. The fact that I give my phone number to somebody does not mean that I will answer the phone every time - I may later decide that I do not want to talk to this person and refuse to answer the phone.

On an abstract level the card metaphor covers most of the user tasks. There seems to be no major conflicts between this metaphor and the security objectives. The question is whether this still holds when these objectives are implemented with available security methods.

The TeSSA architecture, that we are using for our implementation, builds on the use of Simple Public Key Infrastructure (SPKI) [7] certificates to express authorisation and thus trust relations [8]. In SPKI terms a certificate is a signed five tuple (I, S, D, A, V) where I is the issuer, S is the subject (or receiver), D specifies whether delegation is allowed, A is the authorisation field describing the content of the certificate e.g. the permissions granted to the subject, and V expresses the validity of the certificate (could be time or other conditions for use). The issuer and subject are usually represented by their public keys. The delegation capability allows forming certificate chains, from say a provider of a service via a distributor to the user of the service. To access the service the user presents his certificate to the provider thus forming a closed certificate loop that can be verified by the provider who then grants access to the requested service.

One of the advantages of SPKI certificates is that they do not rely on a hierarchy of certification authorities (CA) as do for example the X.509 based identity certificates. In a hierarchical structure trusting a CA automatically implies trusting all the instances trusted by that CA. In other words trust is seen as transitive. In SPKI the delegation capability lets the issuer specify if the subject is trusted to make decision on behalf of the issuer.

The secure business card can be implemented as just a SPKI certificate or it can be some other data structure (such as the vCard electronic business card [9]) where only the security information is expressed as a certificate. This should not have any significant effect on either the card concept as seen by the use or the use of certificates - it is more a question of compatibility with other applications. Next we present two examples of how the secure business card metaphor could be implemented with certificates: granting a colleague the right to read your (business) mail during your holiday and downloading and running a Java application.

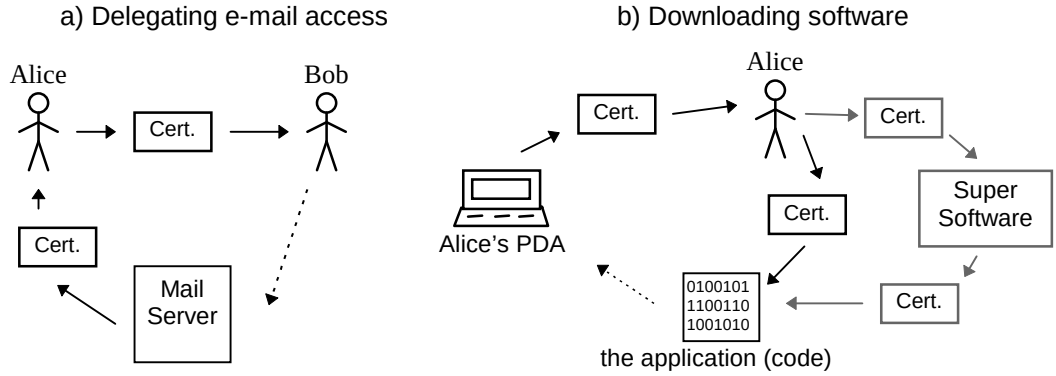

Figure 1. SPKI certificate loops delegating e-mail access and for downloading software.

The certificate loop needed for granting access to read mail is shown in figure 1a. The Mail Server has issued a certificate to Alice giving her the authorities needed to read and send mail. Alice is also given the right to delegate these authorities for a short period of time to avoid interruptions in customer services. Now Alice can issue a certificate to Bob, possibly with reduced authorities, valid for the duration of her holiday. Bob presents this certificate to the Mail Server and is granted access to Alice's mail. What Alice would do at her computer is to move Bob to the "colleagues allowed to read my mail" group if she has one or, if not, modifying the card given to Bob to include mail access. Next she notifies Bob of the changes by sending him the card or asking him to pick up the card from the company's card server. As long as the certificate is valid Bob now has access to Alice's mail.

Downloading and running a Java application involves several parties. The ones relevant from Alice's point of view are shown in figure 1b. For the application to run on Alice's computer it has to be given certain access rights. To make the certificate loop complete the computer must issue a certificate to Alice giving her the right to delegate access to the computers resources. In practice this would be something taken care of by an administrator. Alice now needs to issue a certificate delegating the authority to access relevant resources to the application [10]. Since the applications does not have a public key it is identified by a secure hash of the code. The application is (after some certificate chain reduction) probably signed by the person or company who distributes it. If Alice decides to trust the distributor enough to run any of their applications (for example in the case of company internal distribution) she could instead issue the certificate to the distributor, allowing it to delegate the certificate further to other applications. What Alice does at her computer once she has decided this is an application she trusts, is to choose which secure business card she wants to give the application. She chooses the predefined card for "local applications" that allows reading and writing certain files but not access to network resources, and thus moves the application to the group "local applications". If Alice instead decides to trust the distributor she saves the card she got from the distributor in the "local applications" group. Next time she wants to download software by this distributor she will only need to click OK once to confirm that she still trusts this vendor.

Besides the trust and authorisation issues handled by SPKI certificates, we need to express and negotiate aspects like encryption of communication and supported key exchange protocols. To the user these are visible as choices when specifying a card. In the TeSSA architecture these are implemented through the Internet Security Association and Key Management Protocol (ISAKMP) [11]. To store certificates in a place that can be accessed by whoever wants to verify a certificate loop the TeSSA implementation uses DNS servers [12]. Overall the TeSSA architecture and the secure business card concept are in good agreement. As of writing this the user interface has yet to be implemented. Implementing and testing will in the end show how well it all works on a detailed level.

Discussion

Based on the experience gathered in this work we claim that applying user centered design methods in developing secure software is worthwhile. To develop a user interface that is intuitive and clear for the average user it is essential to find out what the user's awareness and knowledge on security issues is as well as what the needs for security are. The challenge is similar to that of developing completely new product concepts: how to find out what the user wants when the user herself does not yet know that? We chose to ask the user about the services and applications they would like to have on their mobile multimedia computer, and relied on security experts to define the security needs for those services. This does still not give any answer on *how* the users would like to deal with the security issues. In our work we tried to extract this information from general discussions on computer security. The objective was to find out what expressions and terms are used in this context and also what the users feel they want to control and what should not be their problem. The results were not very convincing, but we believe the reasons are more in the inexperience of the interviewers and a too small material.

The development of this user centered security concept was done in a 'meet halfway' type of situation. Throughout the design process we had a specific implementation method in mind (SPKI certificates), but since we did not have any specific application in mind it was natural to start with an empty table and work from the user side towards the security application and

the implementation. The situation is somewhat different when designing a user interface for an existing solution [5]. The secure business card concept is not bound to any specific type of hardware. The same metaphor can be applied to any personal computer, be it a desktop computer, laptop or pocket computer. The concept is, on the other hand, designed for personal use and is not likely to scale well to, e.g. the tasks of a professional network administrators.

Our preliminary tests show that users feel comfortable with the secure business card concept. More testing still needs to be done to confirm that it covers all the security tasks the user needs to perform as well as to identify any sources of dangerous misunderstandings. In the future we intend to design a working pilot application to further test and to demonstrate a user centered security concept. Other more immediate future work includes further research on user security awareness and need. One interesting application area to look into is security and usability issues of the electronic identity (EID) and EID cards that are being taken into use in Finland and many other countries [13].

References

[1] White G. B., Fisch E. A., Pooch U. W.: Computer Systems and Network Security, CRC Press 1996, p 1-3.

[2] Haller N., Atkinson R.: On Internet Authentication, RFC 1704, 1994

[3] The Open Group Research Institute, Adage System Overview, published on the web at http://www.camb.opengroup.org/www/adage/relatedwork.htm, July 1998.

[4] Zurko M. E., Simon R. T., User-Centered Security, New Security Paradigms Workshop, 1996.

[5] Whitten A., Tygar J. D.: Usability of Security: A Case Study, Carnegie Mellon School of Computer Science Technical Report, December 1988.

[6] JRA - The Usability Connection, ABC of User-Centered Design, published on the web at http://www.jra4usability.com/abc.htm, April 1997.

[7] Ellison, C. M., Franz B., Lampson B., Rivest R., Thomas B.M., Ylönen T.: Simple Public Key Certificate, Internet-Draft, work in progress, March 1998.

[8] Lehti I., Nikander P.: Certifying Trust, Proceedings of the Practice and Theory in Public Key Cryptography, 1998.

[9] The Internet Mail Consortium (IMC): vCard- The Electronic Business Card version 2.1, September 1996 (http://www.imc.org/pdi/vcard-21.doc).

[10] Nikander P., Partanen J.: Distributed Policy Management for Java 1.2, Proceedings of Network and Distributed System Security Symposium, 1999.

[11] Maughan D., Schertler M., Schneider M.,Turner J.: Internet Security Association and Key Management Protocol (ISAKMP), Internet-Draft, work in progress, November 1998.

[12] Nikander P., Viljanen L.: Storing and Retrieving Internet Certificates, Proceedings of the Third Nordic Workshop on Secure IT Systems, 1998.

[13] Vestrekisterikeskus (Population Register Centre): FINEID Technical Specifications, published on the web at http://www.vaestorekisterikeskus.fi/hst.htm, February 1999.