A Case Study in Using Design Principles for Secure Operating System Interfaces

Yiming Sun[1], Adity Mutsuddi[1], Baizil K. Jacob[2], Kay H. Connelly[1], and Minaxi Gupta[1]

[1]Computer Science Department, Indiana University
150 S. Woodlawn Avenue, Bloomington, IN 47401
{yimsun, amutsudd, connelly, minaxi}
@cs.indiana.edu

[2]School of Informatics, Indiana University
901 E. 10th Street, Bloomington, IN 47408
bkjacob@indiana.edu

ABSTRACT

We present a three-phase case study in the use of design principles for designing usable and secure operating system (OS) interfaces. For the first phase, we performed a cognitive walkthrough of existing interfaces of a popular OS for creating accounts, logging into the computer, and configuring the set of services which run on the computer. After identifying potential problems, in phase 2, we designed interfaces that were meant to mitigate these problems. In the final phase, we performed a user study that examined the use of both interfaces. Participants using our interfaces had more secure behavior than those who used Windows XP interfaces.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: User-centered design, D.4.6 Security and Protection [**Operating Systems**].

General Terms

Design, Security, Human Factors.

Keywords

Security, interfaces, case study, usability, design principles, configuration, computer accounts, login.

1. INTRODUCTION

Any device connected to the Internet today is vulnerable because countless viruses, worms, and other types of malicious software (aka *malware*) are on a constant lookout for attack targets. Home machines make particularly attractive targets because most are managed by end users who do not possess adequate knowledge for securing them.

According to generally accepted guidelines for securing machines on the Internet, the end users should: 1) practice the *principle of least privilege* (the principle of least privilege advocates for users and processes to use the minimal privileges necessary to perform tasks), 2) run only the services they require, 3) choose strong passwords, 4) keep their anti-virus and anti-spyware checker software up-to-date, 5) apply patches released by companies to fix security bugs in existing software, and 6) use firewalls to block unwanted connections. Of these guidelines, the first two relate to operating system (OS) interfaces users encounter while using their machines on a daily basis, the third relates to passwords, and the last three are security-related applications or tasks. In this paper, we focus on the OS interfaces that users encounter on a daily basis without realizing that they have important security implications.

Broadly speaking, two approaches can be taken to encourage users to behave securely in their daily interactions with their machines. The first is to educate them about secure behavior. Since it is not clear how this goal can be achieved effectively, we take the second approach, which is to ensure that the interfaces themselves encourage secure behavior.

There is limited work demonstrating techniques to help designers with security-related interfaces. In this paper, we present a three-phase case study in the use of design principles for designing usable and secure OS interfaces. For the first phase, we performed a cognitive walkthrough of existing interfaces of a popular OS for creating accounts, logging into the computer, and configuring the set of services which run on the computer. After identifying potential problems, we designed interfaces that were meant to mitigate these problems during phase two. In the final phase, we performed a user study that examined the use of both.

Overall, we found that participants using our interfaces had more secure behavior. The errors that remained suggest that interfaces need to be more aggressive in automating secure decisions, perhaps removing less secure options from view. In addition, our study showed that participants with more computer and security knowledge did not perform better than those with less, suggesting that user education would not solve security problems. Finally, we had contradictory results in the use of warnings with participants ignoring some warnings that encouraged more secure behavior and heeding others that prevented them from correctly configuring their machine.

2. RELATED WORK

In this section, we first review the existing research that has investigated the usability of security interfaces where security is not the primary task. We then describe the efforts in usable security design guidelines, followed by a discussion of the methods for performing user studies in this domain.

2.1 Secure Interfaces

There are many studies on the usability of security interfaces [3, 4, 6, 7, 9, 10, 13, 14, 22, 23, 24, 25] but not as many on applications where security is not the primary task yet where there are serious security implications. Ensuring such applications have secure interfaces has been investigated in the areas of wireless networks [2], peer-to-peer file sharing [11] and encrypted email [8]. These studies have identified many usability issues with existing interfaces and have provided improvements to make them more usable and secure.

Warnings that explicitly describe the consequences of a risky decision have been found to be useful [11, 12]. In addition to text, visual cues and icons are valuable for alerting users about security and privacy issues [8].

Good and Krekelberg observed that interfaces most often assume that users are knowledgeable, which is not the case [11]. Users think about security in terms of their application goals and not in terms of security specific detail [1, 21]. Therefore, the technology must implement *implicit* security, a method in which security is automatically enabled when required by a particular application. Interfaces that hinder users from accomplishing their task will result in them having either everything turned on or off [5].

While designing our interfaces, we assume little user knowledge and take this application approach. We also use warning and visual cues to alert the user about the results of their choices.

2.2 Design Guidelines

There exist many design guidelines for security or usability, but relatively few for usable security. In 1975, Saltzer and Schroeder presented the first security design principles [18]. These 8 principles specifically target protection mechanisms. The last specifically addresses usability: *psychological acceptability*. The practical interpretation of the principle of psychological acceptability has been to make security as invisible as possible, leaving the user out of the loop. While it is true that users will make mistakes, the complex nature of computing makes it impossible to anticipate all possible situations and thus automate all security decisions. It is important to understand why users make errors in order to prevent and recover from such errors.

Norman provided design principles of usable systems based on common human errors, and suggests 4 principles (i.e. feedback, similarity of response sequences, reversible actions and consistency of the system) [16]. Nielsen developed "Discount Usability Engineering" where he provides ten cost-effective heuristics for usability engineering: 1. simple and natural dialog, 2. speak the user's language, 3. minimize the user's memory load, 4. consistency of language and interface, 5. feedback at appropriate times, 6. clearly marked exits, 7. shortcuts for expert users, 8. good error messages which suggest appropriate user actions, 9. prevent errors whenever possible and 10. help and documentation relevant to user's tasks [15].

Whitten and Tygar introduce the first explicit investigation of usability and security in their seminal paper [22]. They performed a traditional usability study on the use of security software and identified five properties that make usability for security software different than usability for other software.

Yee used an Actor-Ability model to identify ten design principles aimed at mitigating computer viruses, which are the first to focus on both usability and security [26]. For example, *safe path of least resistance* states that the easiest way to accomplish a task should also be the safest way.

In this paper, we identify five design principles from the literature that are most relevant to OS interfaces for end users and use those to evaluate existing interfaces and design new ones.

2.3 Methods

When performing a user study of security interfaces, it is important to replicate the conditions that a user would experience in the real world. A primary difficulty is that instructions given to participants and tasks to be performed can prime the user to attend to security issues more than they normally would. In addition, participants are highly motivated to complete tasks during studies because they believe that is what is expected of them [17]. So making claims about how participants would perform on their own computers is difficult.

To minimize some of these effects, user studies often employ the use of role-play with the security-related tasks embedded in a larger scenario [4, 20, 24].

An important part of our study is to understand why a participant makes errors or has difficulty with a particular task. A common technique to elicit this information is to ask the participant to *think aloud*. However we have found that participants have trouble with this technique, often forgetting to explain their actions.

Instead, we utilized the constructive interaction scenario of *joint-exploration* [19]. Joint-exploration utilizes deception to have the participant complete tasks with another "participant", who is actually a researcher. This method is particularly useful when the tasks are highly complex, as the joint explorer pretends to have a similar background as the participant and thus can effectively engage the participant in a dialog without the participant feeling inadequate for not being able to complete the task. In addition, when the participant becomes stuck, the joint explorer can move the task along by "discovering" or "suggesting" the next step.

3. CASE STUDY: PHASES 1 and 2

Our goal was to examine OS interfaces that have important security implications, but where security is not the primary focus of the user. We focus on account creation, user login and configuration of background services. We first identified five principles to guide us in this study:

P1: use consistent and meaningful vocabulary

P2: minimize user's memory load

P3: have safe defaults

P4: make security choices visible

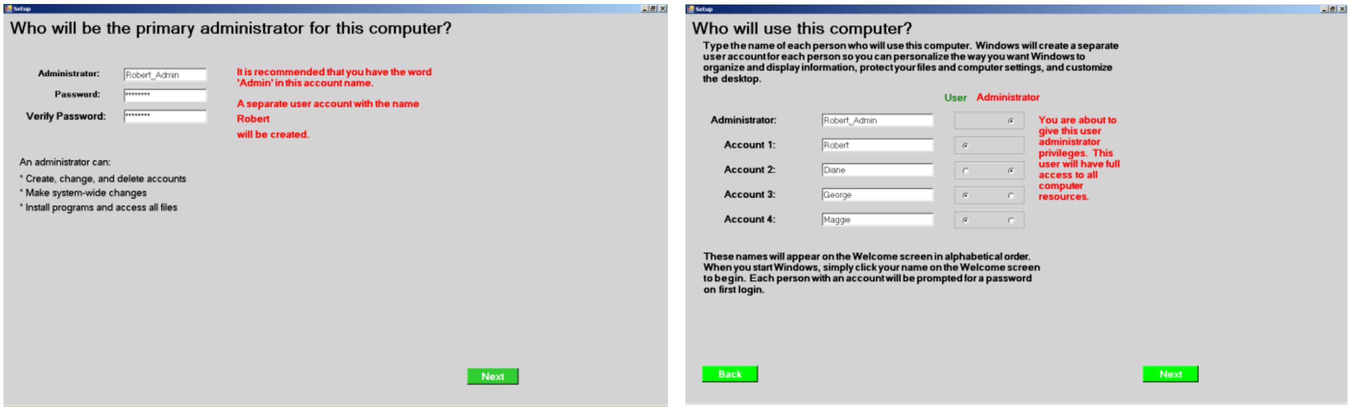
P5: make secure choices the most obvious ones.

We then performed a cognitive walkthrough of the interfaces used by a popular OS: Windows XP. Once we identified potential problems in the existing interfaces, we designed new interfaces, attempting to adhere to the five design guidelines. Finally, we performed a user study to evaluate both sets of interfaces. This section summarizes the results from the first two phases, namely, cognitive walkthrough and interface design, for each of the three target tasks.

3.1 Account Creation

Windows XP has two types of accounts: *Computer Administrator* and *Limited*. An administrator can install software, modify OS settings, create new accounts and change privileges for existing accounts. A limited account can change settings on their own profile, such as their password, but may not be able to install certain software or modify operating system settings.

During OS installation, Windows XP allows up to five accounts to be created, but automatically gives them all administrator privileges without clearly articulating this default, violating P3 and P4. It is up to the user to login and navigate a complex series of screens to change those accounts to Limited, violating P2 and P5. Further, the descriptive text during installation and when changing account permissions is often inadequate and inconsistent, violating P1.



Setup

Who will be the primary administrator for this computer?

Administrator: Robert_Admin

Password: ********

Verify Password: ********

It is recommended that you have the word 'Admin' in this account name.

A separate user account with the name
Robert
will be created.

An administrator can:

* Create, change, and delete accounts
* Make system-wide changes
* Install programs and access all files

(a)

Setup

Who will use this computer?

Type the name of each person who will use this computer. Windows will create a separate user account for each person so you can personalize the way you want Windows to organize and display information, protect your files and computer settings, and customize the desktop.

		User	Administrator
Administrator:	Robert_Admin		●
Account 1:	Robert	●	
Account 2:	Diane	○	●
Account 3:	George	●	○
Account 4:	Maggie	●	○

You are about to give this user administrator privileges. This user will have full access to all computer resources.

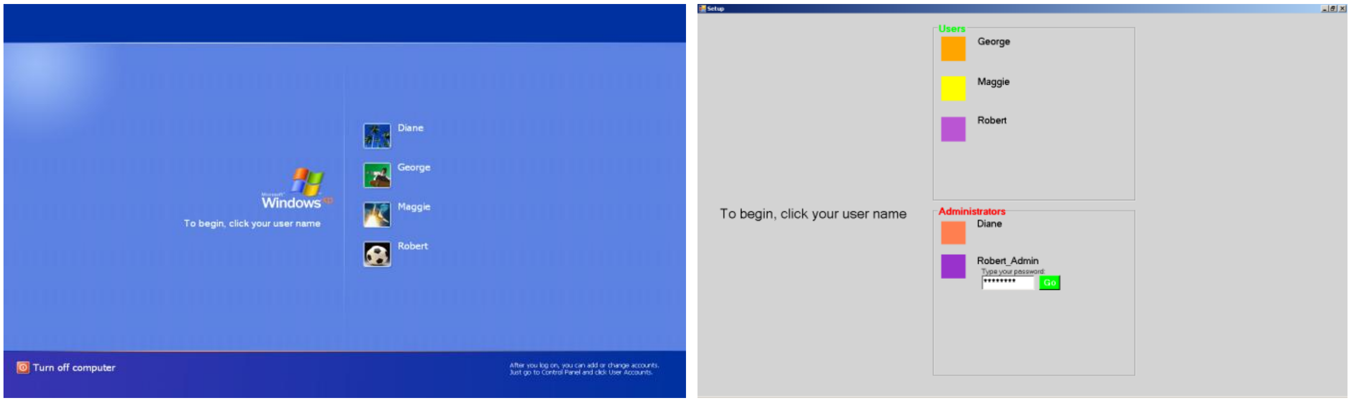
These names will appear on the Welcome screen in alphabetical order. When you start Windows, simply click your name on the Welcome screen to begin. Each person with an account will be prompted for a password on first login.

(b)

Figure 1: Account creation screens for our design.

Figure 1 shows the account creation interface we designed for the installation process. Figure 1(a) shows the first screen where the user chooses an administrator account name and password. This screen articulates the privileges of an administrator in simple terms. Once the user types in an account name, the interface automatically appends an "admin" to the name and gives descriptive text as to why.

Figure 1(b) shows the next screen where the user can choose to create additional accounts. Our interface automatically takes the administrator account name from the previous screen and creates two accounts: 1) <name>_admin, which has administrator privileges and 2) <name>, which is a regular user account. In this way, the default is to create both an administrator and user account for the primary person in charge of maintaining the computer. Additional accounts created at this time are explicitly assigned to be users (default) or administrators. If a second administrator account is created, the interface displays a textual warning explaining the significance of the decision.



To begin, click your user name

After you log on, you can add or change accounts.
Just go to Control Panel and click User Accounts.

(a)

Setup

To begin, click your user name

Type your password: *********

(b)

Figure 2: The login screen for (a) Windows XP and (b) our design .

3.2 User Login

Figure 2 shows the login screen for a) Windows XP and b) our proposed interface. The Windows interface does not indicate the account type, violating P4. Our design separates accounts by type, and presents user accounts on the top in order to encourage people to choose a user account (P5).

3.3 System Configuration

Upon default installation, Windows XP has over 60 processes running at system startup. Many of these processes are not required for a typical user, and thus unnecessarily expose their machine to attacks on those services. Windows XP has a suite of system tools to assist users in managing the processes and services on their machine. Next, we briefly describe these tools. A more complete description including screen shots can be found in Appendix D.

Services is a tool for managing all Windows services. It provides information about the name of the service, how it starts, and whether it is currently running or not. Users can start and stop services and decide if they should be enabled upon system boot. The tool provides a description of each service, but in highly technical terms which can be confusing, even to people trained in Computer Science. The tool can be used to view the dependencies between services, but the navigation is complex and difficult to use as services are not grouped in any logical order.

System Configuration (i.e., msconfig) is a tool to manage all currently running services. Users can disable services and indicate if a service should run at startup. The tool uses technical language and does not provide such useful information as recommended settings.

Task manager is a tool for viewing and managing running applications and processes. Through the task manager, users can access detailed information such as a process's memory usage, CPU usage and network activity. While many processes are related to services, the task manager does not provide information to link a particular process to a particular service.

These three tools violate our design principles in numerous ways. The most significant problems for each principle are:

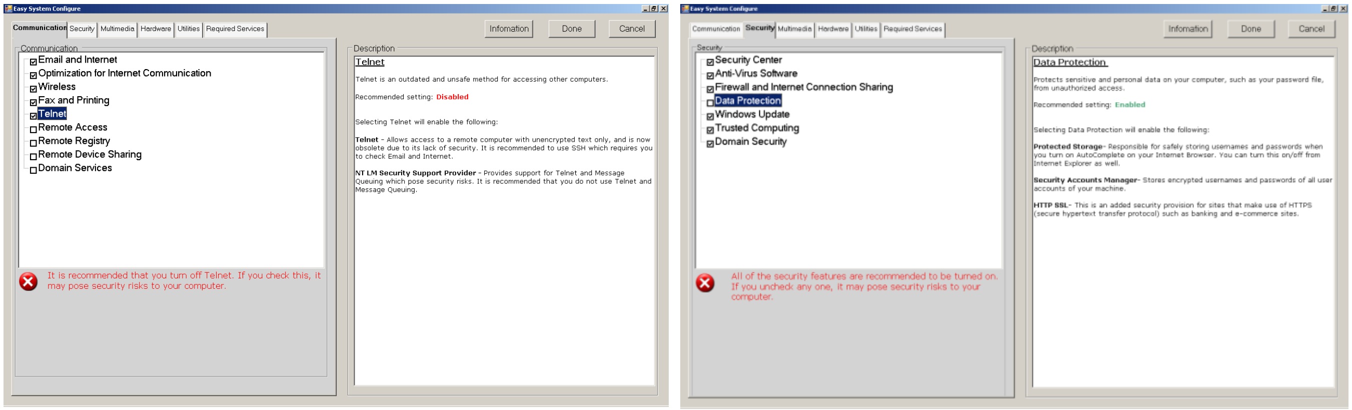
P1: Most descriptive text is at an inappropriate level for users, incorporating highly technical terms. There is a mismatch between the language people use to describe their interactions with computers and the vocabulary employed by the tools. For example, people talk about performing well-defined *tasks*, like writing email with specific *applications*, like Outlook. The Windows tools utilize system-centric language such as "process" and "service" which is not grounded with people's experiences.

P2: In order to understand the state of the system, the user must use multiple tools. Linking information between the tools requires the user to match up various fields, such as process ID and executable name. Further, the tools are often difficult to find and located in different places instead of one central location.

P3: The default configuration is to have a plethora of services turned on which the user may or may not need. Indeed, the tools often advise the user to run a service or process "unless [it is] suspected to be causing problems".

P4: Since security is not the focus, these tools do not make security implications of running a service visible to the user.

P5: The tools do not make which services are required for a particular application obvious to the user. If the user wants to turn off unnecessary services, they must compile information from other sources (e.g. the web), or use trial and error to turn off suspect services and see if the application still functions.



☑ Email and Internet
☑ Optimization for Internet Communication
☑ Wireless
☑ Fax and Printing
☑ Telnet
□ Remote Access
□ Remote Registry
□ Remote Device Sharing
□ Domain Services

Telnet

Telnet is an outdated and unsafe method for accessing other computers.

Recommended setting: **Disabled**

Selecting Telnet will enable the following:

Telnet - Allows access to a remote computer with unencrypted text only, and is now obsolete due to its lack of security. It is recommended to use SSH which requires you to check Email and Internet.

NT LM Security Support Provider - Provides support for Telnet and Message Queuing which pose security risks. It is recommended that you do not use Telnet and Message Queuing.

It is recommended that you turn off Telnet. If you check this, it may pose security risks to your computer.

(a)

☑ Security Center
☑ Anti-Virus Software
☑ Firewall and Internet Connection Sharing
□ Data Protection
☑ Windows Update
☑ Trusted Computing
☑ Domain Security

Data Protection

Protects sensitive and personal data on your computer, such as your password file, from unauthorized access.

Recommended setting: **Enabled**

Selecting Data Protection will enable the following:

Protected Storage- Responsible for safely storing usernames and passwords when you turn on AutoComplete on your Internet Browser. You can turn this on/off from Internet Explorer as well.

Security Accounts Manager- Stores encrypted usernames and passwords of all user accounts of your machine.

HTTP SSL- This is an added security provision for sites that make use of HTTPS (secure hypertext transfer protocol) such as banking and e-commerce sites.

All of the security features are recommended to be turned on. If you uncheck any one, it may pose security risks to your computer.

(b)

Figure 3: Easy System Config Interface

Because of the numerous problems with the existing set of tools, it was unrealistic to fix the problems by making simple adjustments to the existing interfaces, as was our approach for account creation and login. In particular, we needed to:

1. ensure that the management tools operate at the appropriate semantic level (P1)
2. provide all of the necessary information through one interface (P2)

3. guide users to disable any unnecessary services (P3)
4. inform users when they enable a service which is not recommended (P4, P5)

Figure 3 shows two of the screens for the tool we designed, called *Easy System Config*. The interface allows users to select which applications they want to use by ticking a checkbox. The tool is responsible for linking the type of application to the underlying operating system services it requires. The tool organizes applications on different tabs according to type. For example, the *Communications* tab in Figure 3(a) includes “Email and Internet”, “Wireless”, “Telnet” and “Remote Access”; while the *Security* tab in Figure 3(b) includes “Firewall and Internet Connection Sharing” and “Antivirus Software”. A user simply has to select the tasks they intend to perform with the computer.

When a user selects a particular task, a description of the task, the types of services it requires and the recommended setting appears on the right hand side of the screen. Descriptions were carefully constructed to minimize technical jargon and to ensure consistency of vocabulary, as well as to give enough information so the user could determine if they did, indeed, desire that type of application.

Figure 3 also shows that if a user goes against the recommended setting, a warning is provided in red notifying the user of a potential risk.

4. CASE STUDY: PHASE 3

For the final phase of our case study, we performed a user study to evaluate both the Windows XP interfaces and the ones we designed. The user study split participants into two groups. The control group completed tasks with the Windows XP interfaces and the test group completed tasks with our interfaces.

Our study utilized a family-based scenario where participants were asked to role-play one of the family members. We informed participants that we are evaluating user experiences with computers and avoided providing detailed instructions. Further, the primary tasks were not security related, which removed the focus from security making it difficult for participants to guess the purpose of the study and adapt their behavior accordingly.

Nonetheless, it is likely that participants put in extra effort to complete the tasks in order to please the study facilitators. However, our primary goal was to examine if participants are able to use the interfaces appropriately. Once we have designed interfaces that are usable by motivated users, an actual deployment where we can study if users actually use the interfaces correctly in real life will be the next step.

Participants completed a series of tasks related to account creation, login and system configuration. We wanted to determine if the potential problems we had identified in the Windows interfaces during the cognitive walkthrough actually were problems, and if our designs better guided users towards secure behavior. We videotaped all study sessions for later analysis. Study sessions lasted one to two hours, and participants were paid $15 at the end of the session.

4.1 Participants

To participate in our study, participants had to 1) be between 25 and 65 years old, 2) use a computer regularly, and 3) not hold a degree in a computing related field. Participants completed a detailed questionnaire at the beginning of the session. The questionnaire collected information about demographics, computer experience, knowledge and behaviors. Questions related to security were distributed amongst the numerous non-security related questions. We included one sub-task in the main part of the study to examine participants’ understanding of a common security-related issue (i.e. a certificate warning). We also had an interview at the end of the session where we asked them questions about their answers on the pre-questionnaire. In this section, we describe participants based on the questionnaire, sub-task and interview results. In general, the participants were fairly representative of typical home users as they were not technically savvy, held many misconceptions about how computers work and at most, employed common heuristics to secure their computers.

4.1.1 Demographics

There were twenty-two participants in total (n=22), eleven in both the test and control groups. Participants were randomly placed in a group. There were an equal number of men and women in the study, with seven females in the control group and seven males in the test group. Five of the participants in the control group, and seven in the test group were between 25-29 years. The control group had two, three and one participant in their thirties, forties and sixties, respectively. The test group had one, one and two participants in their thirties, forties and fifties, respectively.

In terms of highest education level, one participant had completed high school, five had some college (three of whom were currently in college), eleven had Bachelors degrees and five had Masters Degrees. Current occupations included eleven students, three administrative assistants/secretaries, two electricians, and one each of teacher, fire fighter, business owner, academic advisor, publications coordinator and working for a not-for-profit.

4.1.2 Computer Experience

Twenty participants had a computer at home, and two participants in the control group only used a computer at work. There were only two Mac users and the rest were Windows users. We asked participants to self-rate their computer expertise on a scale of one to five, with five being the highest. From each group, three participants rated themselves as a 2, five as a 3 and three as a 4.

Two participants in each group sought professional help to set up their home computer. The other participants either set it up themselves or asked someone they knew to help. All participants used computers everyday primarily for email, word processing, web, music, videos and editing/viewing photos. Many participants also used the computer for accounting or financial purposes. Among the external devices, printers and scanners were widely used. Participants also used memory sticks, flash drives and external hard-drives. Among the multimedia devices, digital cameras, sound related devices and iPods were most common. More than half of the participants used multiple browsers. The two most popular browsers were Internet Explorer and Firefox. Safari, Opera, Mozilla and Netscape were also used. With the exception of two participants, everyone had changed their browser settings. Basic settings such as default homepage, policies on cookies and privacy settings were changed most often. A few participants had also changed settings that limited specific scripts and web sites.

4.1.3 Computer Knowledge and Behavior

Although everyone used the computer for email and web browsing, participants did not have an accurate notion of computer networks. Only six participants in each group thought their computer was on a network. One participant said the computer at work was part of a network but the one at home was not. It seems that many people think a computer is on a network when they see other computers physically connected to theirs.

We asked participants about their download and web browsing behavior. We found that participants download software, music and video from the Internet knowing that these may be harmful. One participant said he was very concerned about security, yet downloaded everything he could.

Many participants said they did not perform software updates. Participants also had a tendency to readily trust web sites. While 19 participants shopped online, only 12 reported checking to see if the web site they were visiting was secure. To trust a web site, participants relied on measures that could be manipulated easily, such as a claim on the web site, the *VeriSign* icon, a warning prompt, *https* or a lock symbol at the corner of the browser. Many used intuition and common sense to trust a web site. One participant said that well-known web sites (e.g. Amazon.com and web sites of financial institutions) should be safe and that the web address could be a clue to the safety of a web site.

4.2 Study Design and Procedure

Participants were recruited through flyers posted in public buildings such as grocery stores. Upon arriving at the test location, participants were given the informed consent statement and told that they would be completing the study with another volunteer. After signing their consent, participants were given the pre-questionnaire. While completing the questionnaire, the joint explorer arrived and was given the same instructions, consent form and questionnaire as the participant. Once both the participant and the joint explorer had finished the questionnaire, the study facilitator gave them the following scenario to read:

The Jones family has just purchased a brand new computer at home. The computer came with Windows XP. Family members will use the computer for various activities as follows:

• Robert (the father) has been using a computer for quite awhile at work and is familiar with the basic maintenance of the system. He plans to use the computer for email, keeping track of expenses, playing games and accessing his office computer from home. He is planning to record his work files onto CD and is interested in purchasing a digital camera in the future so he hopes to learn how to edit pictures on the computer. He is also concerned about security and wants to run anti-virus software and a firewall.

• Diane (the mother) is also familiar with general computer usage. She plans to use the computer mainly for email, listening to music, watching movies and downloading music file from the Internet. She has an iPod and wants to transfer music files to her iPod. She is interested in experiencing the new Windows XP look and feel through 'Desktop themes'.

• George (the 15 year old son) is highly computer literate and has used computers both at school and at friends' houses. His parents want him to use the computer mainly for homework (i.e. using a word processor). George needs to print out assignments for school. He is really looking forward to playing games.

• Maggie (the six year old daughter) has not had much experience with computers yet. Her parents would like to buy her some educational games to improve her math and reading ability, but they feel she is too young for email and the Internet.

Imagine that you and your partner are members of this family. You will have to install the OS and create accounts for all four members of your family. In order to setup your computer to be usable for daily requirements, you and your partner have to complete the given task together. Make sure you work together to solve each task and come to agreement on the best configuration of the machine.

Participants were then given a series of tasks (Full task text can be found in Appendix A). Each task was provided on a piece of paper and the next task was not provided until the participant and joint explorer indicated they had finished.

Task 1 was concerned with account creation. We noted how many and what types of accounts participants created. We considered correct behavior to be the creation of a user account for every administrator account so that a person in charge of administrative tasks on the computer could also log in as a user for day-to-day tasks.

Tasks 2, 3 and 4 examined user login. Task 2 instructed participants to log onto the computer and visit a web site. The purpose was to see if participants would log in with a user account to perform day-to-day tasks which do not require administrative privileges. Task 3 directed participants to log onto the computer in order to install a software package. The purpose was to see if participants thought they needed to log in as an administrator for software installation. If the participant had created at least one user account during Task 1 and if they had logged in as an administrator for Task 3, participants were given Task 4. Task 4 explained that it was possible to install software while logged into a user account, and instructed participants to log into a user account to install another software package. Participants in the control group would have to use the "*Run As*" feature provided by Windows XP, while participants in the test group had a special installation interface designed by us that automatically prompted the participant to select an administrator account for the installation process. For these three tasks, we noted which account participants used, and for Task 4, if they understood that installation was occurring as an administrator.

Task 5 was the most time consuming task, and instructed participants to configure the computer so that it would run the applications the family wanted as described in the scenario. The control group was told the names of the three applications described in Section 3.3, while the test group was given the name of the tool we designed.

For the control group, we noted how long it took for participants to either find the tools or give up. If they gave up, the study facilitator showed them where to find the tools. For both groups, we noted the time participants took using the tool(s), the configuration they arrived at and how they checked the configuration of the computer, if at all. The joint explorer actively engaged the participants in a dialog about the decisions they were making. During later analysis of the video, we also categorized the strategies that participants employed in order to configure their machines.

After Task 5, the study facilitator explained that she needed to interview both participants and asked the joint explorer to wait in another room while she interviewed the first participant. The interview included questions about performing the tasks with their partner, their perceptions of the interfaces and their understanding of general and security-related computer terms. After the interview, the facilitator debriefed the participant and explained the deception employed by use of the joint explorer. Five participants suspected that the joint explorer was a researcher.

5. FINDINGS

Table 1 summarizes participant knowledge about computers and security issues and their success at account creation, login, and configuring services. For each category, we ranked the participants on a scale of 0 to 2, with 0 implying failure/lack of knowledge and 2 implying success/good knowledge. For ease of visual depiction, Table 1 shows a filled circle for a score of 2, a semi-filled circle for a 1, and a hollow circle for a 0. Overall, participant knowledge about computers varied across both groups and the test group had better success at completing the tasks securely. This section details these findings.

Table 1: Summary of results

Particpant ID	Knowledge	Account Creation	User Login	Configuration
TEST				
10	●	◐	○	●
2	●	◐	○	◐
9	●	◐	◐	○
20	◐	●	●	●
4	◐	●	●	◐
12	◐	●	●	◐
5	◐	◐	○	●
11	◐	◐	◐	◐
7	○	●	◐	◐
6	○	◐	○	●
8	○	◐	○	◐
CONTROL				
19	●	◐	○	◐
21	○	○	○	◐
3	◐	○	○	○
15	◐	○	○	○
16	◐	○	○	○
18	◐	◐	○	—
22	◐	○	○	—
23	◐	○	○	—
13	○	◐	○	—
17	○	○	○	○
14	○	○	○	—

	●	◐	○	—
LEGEND	Good	Average	Poor	Skipped

5.1 Participant Knowledge

Using the answers to 24 questions asked during the exit interview, we ranked participants' knowledge about computers, networks, and security issues on a scale of 0 to 2. The result of this ranking is shown in the second column of Table 1. Most participants did not seem to have a good grasp on concepts like what it means for a computer to be on a network, what makes a web site secure or how harmful web downloads can be.

As an example, we discuss in depth one particular aspect of user knowledge about security: certificate warnings displayed by web browsers. A valid certificate authenticates that the web site being visited is indeed genuine, and a web browser displays a dialog about a certificate only when something is wrong with it.

As part of task 2, we asked participants to visit an unfamiliar web site. When they typed in the web address we provided, the browser displayed a certificate warning and gave the participant the choice to accept the certificate always, once, or not at all. The participants could also examine the certificate. The joint explorer was careful to engage the participants in a discussion.

A few participants exited in fear of going to an untrustworthy web site and a few exited in confusion. Some participants, although concerned about the web site's authenticity, ended up visiting the web site. Eight participants ignored the certificate and visited the web site with a carefree attitude. These participants appeared confused about what to do and did not understand what a certificate was. Seven participants examined the certificate without prompting from the joint explorer, while five examined with prompting from the joint-explorer. When participants examined the certificate, they looked for trusted names, such as university names and well-known organizations. Most participants did not understand what they were examining and the examination did not help in making a decision about the certificate warning.

Their conversations with the joint-explorer suggest that most participants did not understand the purpose of a certificate. Only one correctly stated what a certificate was and what it meant when the browser displayed the dialog regarding the certificate. Two were able to vaguely explain what a certificate meant. Another one could not explain what a certificate was, yet did point out that when the browser showed a dialog about the certificate, the web site was not the web site the user wanted. Five participants had no idea what a certificate was.

The remaining thirteen participants had incorrect notions about certificates thinking that the browser was displaying the certificate as some sort of proof of security. Six thought a certificate meant the web site was free from virus or scams. One thought a certificate was to authenticate the identity of the client (i.e. the participant). Two thought a certificate was an indication of how secure the web site was. One participant thought it was a part of the web site and another said he could not access the site unless he accepted the certificate. The remaining two used intuition when examining the certificate, with one determining trustworthiness based on keywords such as well-known organization names, and the other accepting the certificate if it was for a "known" site. None of these participants realized that the browser displayed a dialog about a certificate because the certificate was questionable. Instead, they came to the opposite and incorrect conclusion that the site was safe because it displayed a "certificate."

5.2 Account Creation

Task 1 of the study compared the number and types of accounts created in Windows XP with those created with our interface. We considered correct behavior to be the creation of a user account for every administrator account so that a person in charge of administrative tasks on the computer could also log in as a user for day-to-day tasks. As the third column of Table 1 shows, none of the control group participants were fully successful in completing this task. The participants in the test group fared much better. Overall, our interface succeeded when only one administrator account was created but failed to enforce the best behavior otherwise. We discuss the details next.

5.2.1 Control group

Most participants in the control group failed to create user accounts, and those who did only created them for the children. The subjects fell in two categories:

1. **No user accounts:** Seven of the eleven participants did not make any user accounts because the default setting of the windows interface only creates administrator accounts. None of these participants created passwords for the accounts.

2. **No user accounts corresponding to administrator accounts:** Four participants created some user accounts, but the administrator account(s) did not have corresponding user accounts. Each of these participants made a user account for Maggie (daughter) because of her age. Two of them also made a user account for George (son). These participants appeared to be concerned about security, but did not understand the significance of having two different types of account for one person.

Eight participants in the control group did not create passwords for any of the accounts. Recall that there is no password enforcement in Windows XP. One participant created a password for the administrator accounts and another created a password for the user account, but was prompted by the joint explorer to do so.

5.2.1 Test Group

Participants in this group created both administrator and user accounts. However, those who created more than one administrator account failed to create a corresponding user account. Participants fell into three categories:

1. **One administrator account with a corresponding user account:** Five of the eleven participants created one administrator account and left the accounts for other family members as user accounts. Four of them made Robert (father)

as the primary administrator and created user accounts for the remaining family members. The interface responded by creating a user account for Robert. The fifth user specified "*admin*" as administrator, which our interface handled differently as described later.

2. **No user account for the additional corresponding administrator account**: Six participants created two administrator accounts, one for both Robert and Diane (mother) Although our interface automatically created a user account for the primary administrator, it did not enforce creation of a user account for an administrator account created on the second screen (Figure 1b). Even though our interface displayed a warning when they created a second administrator account, participants were not able to grasp the idea of multiple roles for one person. None of the participants seemed to notice there were two accounts for the primary administrator. When the joint-explorer pointed it out to two of the participants, they ignored the comment and moved on.

3. **Primary administrator account did not have corresponding user account**: Two of the eleven participants created "*admin*" as the primary administrator, as opposed to using a person's name. In this case, our interface did not create a corresponding user account. One of them considered "*admin*" to be George (son) and did not create a user account for George. The other participant considered the "*admin*" account as the system administrator, and, in addition, gave Robert and Diane administrative privileges, without creating corresponding user accounts.

All of the participants in the test group created passwords for the primary administrator account because our interface enforced this behavior using principle P5. Most of them assigned a weak password for that administrator account. Only one participant was concerned about creating a secure password and spent some time on it. Of the eleven participants, only three created passwords for both administrator accounts and user accounts. One of these three was prompted to do so by the joint-explorer.

5.3 User Login

Tasks 2, 3, and 4 of the study examined user login behavior for tasks ranging from visiting a web site to installing software. We considered correct behavior to be logging in with a user account for both the web browsing and installation tasks. As the fourth column of Table 1 shows, participants in the control group performed poorly, always logging in as an administrator. Participants in the test group performed better, but still had significant problems. In general, we found participants logged in as an administrator because they wanted the power of an administrator or because they played a role from the scenario for which they had only created an administrator account in task 1.

5.3.1 Control Group

None of the participants in the control group used a user account for tasks 2 and 3. Recall that during account set-up only four participants created user accounts, and only for the children. These participants pretended to be Diane or Robert, so used the administrator accounts they had created for the parents. The remaining seven participants only had administrator accounts from which to choose.

5.3.2 Test Group

Task 2: Logging in to visit a web site

Six participants in the test group used the administrator account for visiting a web site and five participants used a user account. One of the participants who logged in as a user was prompted by the joint-explorer who asked the participant if it was possible to go to a web site using a user account. The participant explained that all type of accounts had privileges to use a web browser and so they could login as a user. Without the prompt, however, the video analysis shows that the participant would have logged in as an administrator.

The remaining four participants who logged in as users had created user accounts for everyone and an administrator account for Robert in task 1. Since everyone in the family had a user account, the participants were more likely to choose a user account if they were role-playing or choosing an account randomly. So, the two participants who randomly chose an account and the two participants who chose accounts based on the role they were playing, picked user accounts. In these cases, our interface for task 1 was helpful in motivating participants to log in as users in task 2. One of the participants was using Robert's account which also had an administrator account. She chose the user account over the administrator account as she didn't think there was much difference between the two accounts.

The six participants who logged in as administrators had created multiple administrator accounts during task 1. They logged in as administrators for three reasons: 1. administrator accounts have more control over the computer, 2. out of habit and 3. because of role-play. It is interesting to note that four participants logged on with Robert's administrator account even though he had a user account.

Task 3: Logging in to install a new web browser

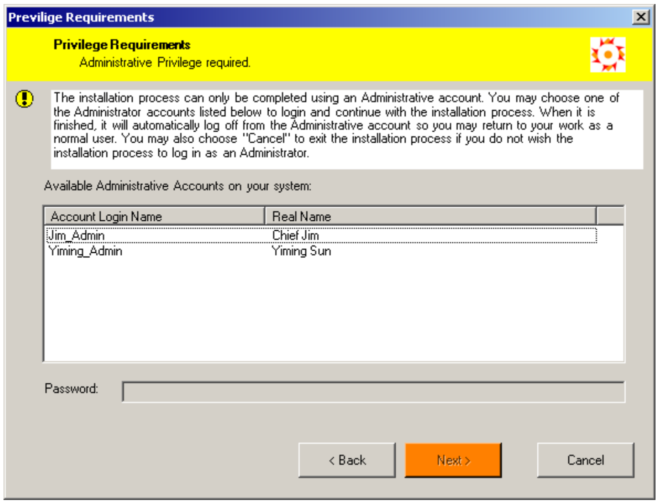
Seven participants in the test group used the administrator account to install the new browser. Two of these participants had used user accounts for the previous task. One of these logged in as an administrator because he thought he needed administrator privileges to install. The other participant thought that the certificate warning in task 2 showed up because she logged in as a user and decided to login as the administrator for this task. The other five participants logged in as administrators because administrator accounts had more privileges and out of habit.

The remaining four participants used user accounts. Two of these participants did not know the difference between a user and an administrator account and so used the user account they had chosen during the first task. One participant knew the differences but continued with the user account he had chosen in the previous task. The remaining participant who had used an administrator account for the previous task, changed his role for this task to a family member (Robert) who only had a user account. He said that since Robert would be using the web browser he should login as Robert to install the web browser. Again, this was a result of role-playing rather choosing a particular account type.

All participants who used user accounts were presented with our special Install Shield, which we designed to prompt participants for an administrator password if they logged in using a user account. We comment on the ease-of-use of our special Install Shield in Section 5.3.3.

5.3.3 *Task 4 for Test and Control Groups*

A typical Windows user logged into a user account has two options if she wishes to perform administrative tasks, such as install new software: 1. log out and log back in as an administrator, or 2. use the *Run As* command, which allows users to perform administrative tasks when they are logged in with limited privileges. Since accessing *Run As* requires special knowledge, we wanted to explore an interface that will make its functionality more accessible. Toward this goal, we designed *InstallShield* (Figure 4), which automatically prompts the user for an administrator password if they are logged in as a user.



Previlige Requirements

Privilege Requirements

Administrative Privilege required.

The installation process can only be completed using an Administrative account. You may choose one of the Administrator accounts listed below to login and continue with the installation process. When it is finished, it will automatically log off from the Administrative account so you may return to your work as a normal user. You may also choose "Cancel" to exit the installation process if you do not wish the installation process to log in as an Administrator.

Available Administrative Accounts on your system:

Account Login Name	Real Name
Jim_Admin	Chief Jim
Yiming_Admin	Yiming Sun

Password:

Figure 4: Interface of our special Install Shield

Participants who had created at least one user account and had logged in as an administrator in task 3 were instructed to login to a user account to install a music player in task 4. We tested the special Install Shield during tasks 3 and 4.

We found that except for two participants, everyone understood the purpose of our special Install Shield. The two participants who did not understand it were able to follow the directions and install the software correctly. The recent release of Windows Vista includes a similar feature. From our study, we can conclude that general users will be able to use this interface.

5.4 System Configuration

Having a minimal set of services reduces the degree of vulnerability to attacks by disabling exploitable services that are not needed by end-users. Our goal in task 5 was to examine participants' ability to turn off unnecessary services while enabling those required by the scenario. As the fifth column of Table 1 shows, participants performed much better with our tool than with the Windows tools. With our tool, they were able to configure the computers without any help from the joint explorer. In the control group, most participants were confused by the Windows tools and half of the participants who attempted to configure their machine needed the joint explorer's help.

Further, our interface failed more gracefully. The errors that the participants made were minor: they did not disrupt the function of the computer and many unnecessary services were disabled. On the other hand, the participants in the control group ended up with a system in which many required applications did not function and many unnecessary services were enabled. In addition, the average configuration time for the test group was considerably less than for those who used the Windows tools.

5.4.1 Control Group

Finding Windows Tools

An important step when configuring with the Windows tools is to find where the tools reside in the computer. *msconfig* can be found by going to the *Run* dialog and typing in *msconfig*. The *Services* tool is in the *Control Panel* inside the *administrative tool*s folder. The *Task Manager* can be found by pressing the keys *Ctrl-Alt-Del* or by right clicking on the taskbar and choosing *Task Manager* from the menu that shows up. During the search process, most participants looked in the *Control Panel* and searched using the search tool in *Windows Explorer*. Many participants looked for information in the *Windows Tour* to help find the tools. *Help* and *Support Center* assisted most participants to find the tools.

On average, participants spent about 6 minutes trying to find the tools, with a standard deviation of 5.69. Two participants spent 10-15 minutes without success. Only one was successful in finding all of the tools. If a participant gave up looking for a tool, the study facilitator showed it to them.

Six participants found *msconfig*. Among these, five found it through *Help* and *Support Center's* search tool. One participant used *Run* to open *msconfig*, but it was already in the cache. One participant took over 19 minutes to find it.

Three participants found the *Services* tool through the *Control Panel*. Two of these were prompted by the joint explorer. The participant who found the tool on her own had used the tool before and knew its location. Even though most participants searched for the tools in the *Control Panel*, they could not locate *Services* as it was under the folder *administrative tools*.

Five participants found the *Task Manager* tool by pressing *Ctrl-Alt-Del* keys and one used the *Help* and *Support Center*. The one who used the *Support Center* knew it could be accessed through *Ctrl-Alt-Del*, but did not know its name was *Task Manager*.

These observations suggest that configuration tools should be kept in the *Control Panel* but not inside a subfolder. Tools should be searchable by *Explorer* as well as *Help* and *Support Center*.

Using the Tools

We considered services to be configured correctly if 56 of the total 80 services were enabled. Of these 56, 48 are required for correct configuration for our scenario and 8 were optional. Additionally, 24 of the services needed to be turned off because they were unnecessary. Nobody in the control group configured their machines correctly. They either had too few or too many services enabled. For this task we report on data for ten participants; one participant was excluded due to computer failure during this task.

Even when equipped with Windows XP tools, five participants did not attempt to configure their computers. These participants have a "-" in column three of Table 1. They were perplexed by the tools and could not determine how to use them. They did not change any settings; their computers worked but they did not have a minimal set of services enabled. Of these, two participants insisted that Windows came loaded with everything and it was not necessary to change anything.

The remaining five participants made some changes to their computers. Four participants used *msconfig* and one used the *Services* tool for configuration. Three of the participants who used *msconfig* were prompted by the joint explorer on how to use the tool. The other participant had used *msconfig* before and knew how to use it.

msconfig: All of the participants were confused by *msconfig*. The participants explored the tool by clicking on the various tabs. One participant said "*This does not make any sense to me at all.*" Another said he had seen the tool but did not know what to do with it. One participant had used it to change startup programs but did not know what else he could do. Both of these participants were unwilling to make any changes even after the joint explorer showed them how the tool might be used. They insisted that Windows came loaded with everything and it was not necessary to change anything.

Of the four participants who made configuration changes using *msconfig*, three were confused at first. It was only after the joint explorer's prompting that they realized how to use the tool. They started by disabling all the services. Later, they enabled those that they thought were required by the family. Two of these four participants were incorrect in their judgment at least 40% of the time, according to the scenario. Even the participant who came closest to the correct configuration made a judgment error about 10 services. We will discuss later the methods participants used to determine the required services. Those who used *msconfig* spent on average about 20 minutes with this tool, with a standard deviation of 6.

Services: With the exception of two participants, everyone was intimidated by this tool. Most of them looked at it, scrolled down and exited. Three participants made the connection between the services list in this tool and in *msconfig* but they did not change any settings.

Only two participants said they were comfortable with this tool. But they did not use it to configure any services. Of these, one had used it at home with outside assistance. The participant who used *Services* for the task relied on the descriptions in the tool. She started some of the required services but did not disable any of the unnecessary services. This participant had refrained from using *msconfig* as the tool did not provide any textual descriptions. This participant spent 18 minutes trying to change the settings.

Task Manager: Six participants had used *Task Manager* to force quit unresponsive programs in the past and had some idea about it. These participants realized that *Task Manager* was not applicable for configuration because it shows the processes that are already running in the system and does not connect them to the services. Though useful for system diagnostics, it cannot be used for configuration. Nobody used this tool for configuration. Among the four participants who could not find this tool, two managed to determine that it showed programs or processes that were running in the computer.

5.4.2 Test Group

Participants were able to use our interface without any help from the joint-explorer. They explored the tabs and went down the list of applications in each tab to enable those that were required. When confused, they read the descriptions to make decisions. If a warning message was shown for a service they had enabled, some participants immediately disabled it even if it was required by the scenario. Only one participant needed prompting from the joint explorer. He wanted to enable everything as he was very confused. The joint explorer prompted him to read the descriptions and then decide. In the post interview, the participant reported that he would have been more cautious if it wasn't an experiment.

Using our configuration tool, four participants correctly configured everything according to the scenario. They turned on the services that the family needed and did not turn on anything that was not necessary. We considered it correct if they enabled services such as *Backup*, *System Restore* and other optional services if they considered it necessary.

Three participants paid more attention to the recommended settings suggested by the tool than the scenario. These participants disabled *Remote Access*, although required by the family, due to the warning displayed by the tool. Two of them disabled *Windows Time* for the same reason. One of the participants also disabled *Desktop Themes* as she thought it would make the computer slow. These participants were correct otherwise.

Three other participants made minor errors such as enabling *Remote Registry* and not enabling *Removable Storage*, which was required by the family to use their digital camera. The descriptive text for *Removable Storage* did not specifically mention digital cameras, which could have assisted these participants. Participants enabled *Remote Registry* thinking that it was related to *Remote Access,* which was needed by the family to allow technical support to access their computer remotely. However *Remote Registry* allows the Windows program registry to be accessed remotely. This error is a result of the way services are named creating confusion for these participants.

The only participant who performed poorly with our interface did not notice that our tool had multiple tabs, one for each category of services. He enabled everything in the communication tab and exited, later explaining that he thought he would not have to configure much as the computer was already configured.

The average time taken to configure the computer using our tool was 10 minutes, with a standard deviation of 4.33, excluding the incorrect participant who did not explore our tool.

5.4.3 Configuration Strategies

In general participants' knowledge about configuration was limited to software configuration. Most participants had configured their browser or other software. One participant had used *Services* with technical assistance and one had used *msconfig* to modify startup programs.

After doing task 5, in both the control and the test group, some participants were able to grasp the idea of computer configuration while some could not think beyond software. In the test group, there were some participants who were confused even after successfully configuring their system.

Most participants realized that configuration can minimize security risks after doing task 5. Two participants mentioned that configuration can actually make the computer more vulnerable if you don't know what you're doing. Overall, various configuration strategies emerged as participants in the two groups worked on task 5.

The minimalist: Two participants turned on only those things they recognized and thought were necessary, disabling all others. Of them, one was very confused and was not confident of his actions. Both of them were using keywords to identify required services. Some of these keywords were *print*, *help and support*, *themes*, *anti-virus related* (e.g. Symantec), *time*, *web client, audio, firewall, image acquisition, computer browser, CD burning, remote access, removable storage, installer, plug and play*.

These participants had a relatively secure computer, as they did not turn on many risky services. However, their computers were not functional, as they failed to enable many necessary services.

When in doubt, turn it on: Two participants enabled services whenever they were in doubt. One of them turned on everything that was related to the vendors Microsoft, Intel or Symantec. It was her belief that since the operating system was Microsoft, one had to trust them. In addition, she thought that the Microsoft services were all interlinked and she did not want to disable something necessary.

The other participant also had a similar strategy: whenever he was unsure about a service he enabled it. In his own words "*I never turn anything off, when in doubt turn it on*."

Both participants were able to turn on many of the required services in addition to enabling many unnecessary and unsafe services. Even with this "turn it on" strategy, one of them failed to enable some essential services required for proper functioning of the computer. Regardless of the strategies, the participants were unable to correctly configure their machines that were both functional and secure.

Follow the warning: While we made use of warnings in our interface to notify participants of possible security vulnerabilities, we found that some participants adhered to warnings even if they conflicted with the scenario. Even those participants who turned something on despite a warning were hesitant. In the post interview participants reported that they were worried when they had to enable something that was recommended to be disabled.

6. DISCUSSION

Our study gave us several insights in to the design of secure OS interfaces. First is that while the principles aided us in designing more secure interfaces, we did not always take them far enough. For example, while our interface helped in creating user accounts, it did not clearly communicate the advantages of having a user account for *every* administrator account. In particular, while people play different roles in real life, (for example, a woman may fill the roles of mother, wife, and employee) participants did not have the notion of assigning multiple roles to a computer user, as demonstrated by no participant who created a second administrator account with our interface also creating a corresponding user account. A better design would automatically create a corresponding user account for all administrator accounts.

Similarly, visually separating the user and administrator accounts was not sufficient for encouraging login to user accounts. The participants who had a choice between an administrator and user account for the role they were playing most often chose the administrator account. This was because of an inaccurate perception that they are unable to do everything when logged in as a user. While this may change over time as users become more familiar with the new security interfaces in Windows Vista, the login screen itself could go further by only showing user accounts. An interface which allows login to administrator accounts could be embedded into the secure key sequence (*ctrl-alt-del*) for more advanced users.

Finally, only a few people tried to secure their system with a password when passwords were not enforced. All the participants in the test group created a password for the initial administrative account because of the enforcement of password by the interface. However, only a few participants in the control group secured their accounts using a password. It seems that requiring end-users to enter a password when creating an account or at initial login would add additional security to their system.

Another theme that emerged during the user study was the use of trust heuristics. In the pre-questionnaire, certificate examination (during task 2) and configuration (task 5), several participants employed different techniques for determining trust. Most often, the participants looked for recognizable keywords, or names of organizations or vendors. Attackers can easily manipulate these heuristics by embedding names of known entities in certificates or application names. Our study shows that most users' limited understanding of the underlying security and operating system mechanisms make them vulnerable to such attacks.

Unfortunately, our study also suggests that general computer and security education is not adequate to solve this problem. Looking at statistical test 1 in Appendix E, there is no correlation between level of computer knowledge and performance on the tasks. Indeed, for the control group, all of the participants with average knowledge and two with poor knowledge performed as well as or better than the three participants with good knowledge. More work must be done to embed correct behavior in tools/applications more fully.

Finally, we had mixed results with warnings. Warnings did not prevent participants from making a second administrator account during task 1, yet did prevent participants from enabling potentially risky services in task 5, even when they were needed. One reason may be that participants were comfortable with the concept of account types, allowing them to ignore the warning. With system configuration, however, many participants were confused and thus more attentive to warnings.

7. CONCLUSIONS

A cognitive walkthrough of common operating system interfaces that have security implications showed several violations of security design principles. Based on those results, we modified interfaces for account creation and login and created a new interface for system configuration.

Our user study verified that participants think at the application level and are better able to use an interface that matches that conceptual model. Further, participants who used interfaces not built around the concept of applications were utilizing heuristics that employed this mental model by looking for keywords associated with particular applications.

While participants using our interfaces had some errors when configuring their machines, their errors were less critical than those made by participants using the existing interfaces. Indeed, not only did our interface result in a more secure configuration, but the existing interfaces often resulted in a serious lack of functionality. Still, there were problems with our proposed interfaces. Once we further iterate on the interfaces and have ones that people *can* use securely, the next step will be a real-life deployment to see if people *will* use them.

8. ACKNOWLEDGMENTS

The authors would like to thank Lynne Crohn for technical assistance with the project, and Anne Faber for working on the preliminary version of the login interface.

APPENDIX A

Table T1. Task instructions for both groups

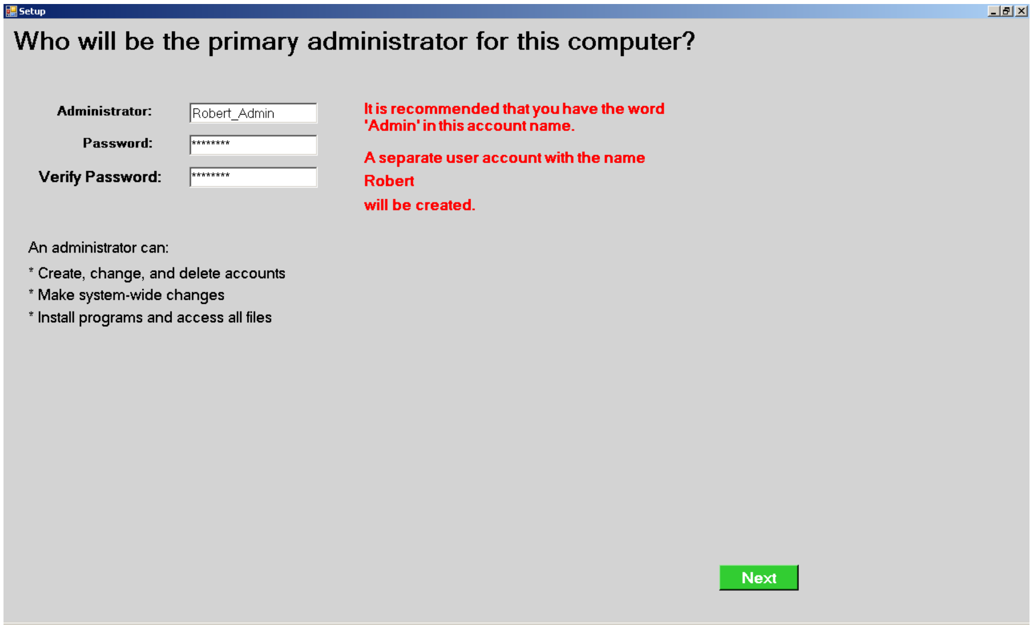
	Control group	Test group
Task 1	Create accounts for all family members according to the scenario.	Create accounts for all family members according to the scenario.
Task 2	1. Login to the computer using your account 2. Using Internet Explorer, go to the website (http://portal.leadproject.org) 3. When you are done, logoff	1. Login to the computer using your account 2. Using Internet Explorer, go to the website (http://portal.leadproject.org) 3. When you are done, logoff
Task 3	As advised by a technically savvy friend, you have decided not to use Internet Explorer as your web browser. Instead, you have found a more secure alternative called Mockingbird. In order to surf the Internet, you need to install Mockingbird on your computer. Here your task is to install Mockingbird.	As advised by a technically savvy friend, you have decided not to use Internet Explorer as your web browser. Instead, you have found a more secure alternative called Mockingbird. In order to surf the Internet, you need to install Mockingbird on your computer. Here your task is to install Mockingbird.

	1. Login to the computer using your account 2. Install Mockingbird 3. When you are done, logoff	1. Login to the computer using your account 2. Install Mockingbird 3. When you are done, logoff
Task 4	N/A	(Optional. This task will only be given to participants who: 1. Logged in to an administrator account during Task 3. 2. Created a user account in Task 1.) In the last task, you logged on to an account with administrator privileges. For this task, log into one of the accounts you created with user privileges. Login to the computer using a user account. 1. Install MyodPlayer 2. When you are done, logoff
Task 5	Members of the family want to use the computer for the following: • Browsing the web and sending/receiving emails • Writing letters/papers using a word processor • Printing out web pages, letters, emails etc. • Allowing technical support representative to take over your computer via remote assistance and perform advanced configurations on it. • Running anti-virus and firewall • Listening to songs, watching movies • Scanning and/or editing family photos, whether they being digital or not. • Experiencing the new Windows XP look and feel through it themes • Allowing your computer to automatically synchronize its clock with dedicated time servers on the Internet • Recording your work files onto CD-R/CD-RW media using Windows XP Now you need to configure the computer so that the family can do all of the above. 1. Login to the computer using your account 2. Configure your computer using tools provided by Windows XP such as “System Configuration”, “Utility Tool” and “Windows Service Tool” 3. When you are done logoff	Members of the family want to use the computer for the following: • Browsing the web and sending/receiving emails • Writing letters/papers using a word processor • Printing out web pages, letters, emails etc. • Allowing technical support representative to take over your computer via remote assistance and perform advanced configurations on it. • Running anti-virus and firewall • Listening to songs, watching movies • Scanning and/or editing family photos, whether they being digital or not. • Experiencing the new Windows XP look and feel through it themes • Allowing your computer to automatically synchronize its clock with dedicated time servers on the Internet • Recording your work files onto CD-R/CD-RW media using Windows XP Now you need to configure the computer so that the family can do all of the above. 1. Login to the computer using your account 2. Configure your computer using “Easy System Config” 3. When you are done logoff

The participants are randomly assigned into one of the two groups: the control group and the test group. Each group has a set of tasks to complete. While the overall goals of the two sets of tasks are the same, there are subtle differences between them that reflect the differences in the tools/software used for each group. Table [T1] lists the task instructions.

APPENDIX B

For the test group, we are using an interface that we designed that could assist a user to setup OS accounts more securely.



Who will be the primary administrator for this computer?

Administrator: Robert_Admin

Password: ********

Verify Password: ********

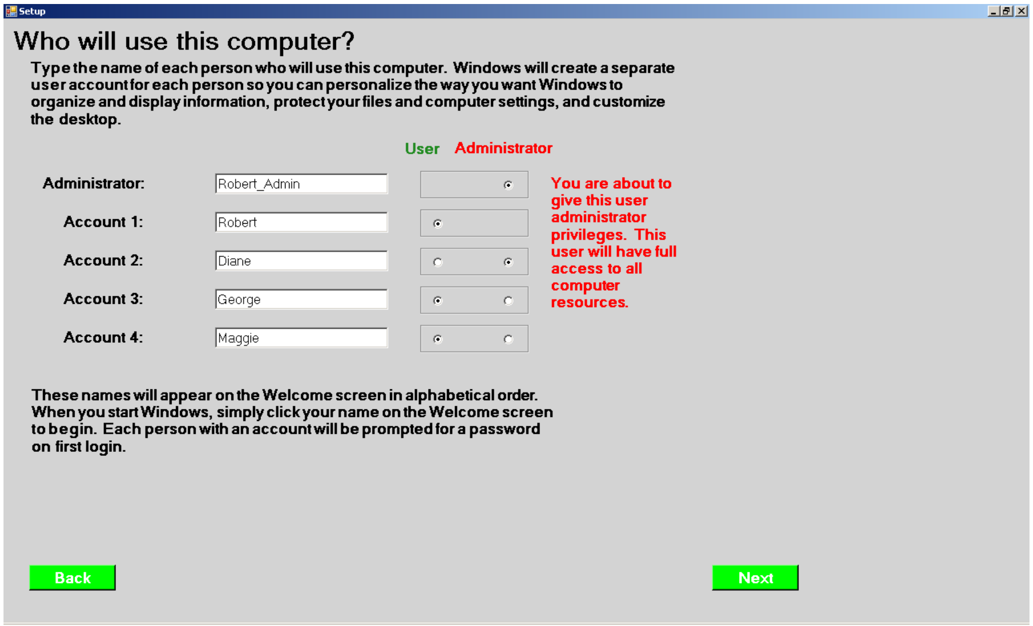
It is recommended that you have the word 'Admin' in this account name.

A separate user account with the name
Robert
will be created.

An administrator can:

- Create, change, and delete accounts
- Make system-wide changes
- Install programs and access all files

This is the first screen, where a user can setup the account for the primary administrator of the computer. The interface tries to add an "_Admin" suffix to the username chosen by the user, to distinguish it from a regular user account.



Who will use this computer?

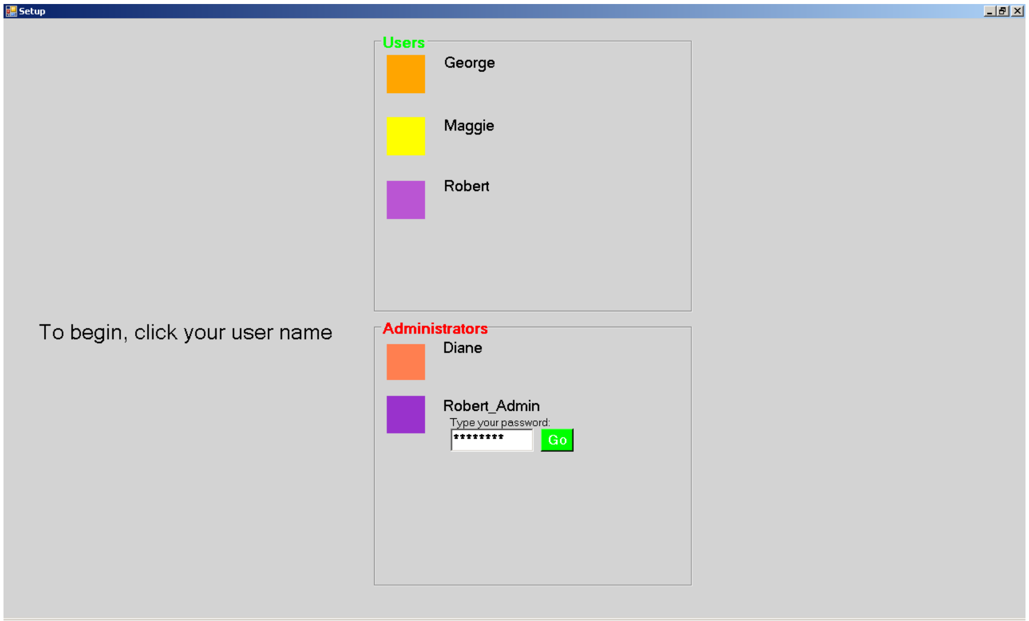
Type the name of each person who will use this computer. Windows will create a separate user account for each person so you can personalize the way you want Windows to organize and display information, protect your files and computer settings, and customize the desktop.

		User	Administrator
Administrator:	Robert_Admin		●
Account 1:	Robert	●	
Account 2:	Diane	○	●
Account 3:	George	●	○
Account 4:	Maggie	●	○

You are about to give this user administrator privileges. This user will have full access to all computer resources.

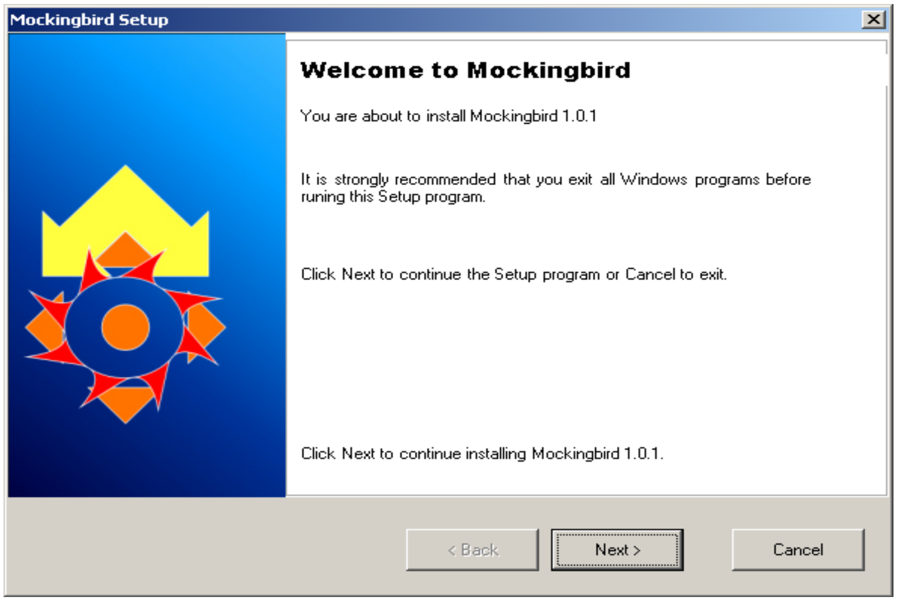
These names will appear on the Welcome screen in alphabetical order. When you start Windows, simply click your name on the Welcome screen to begin. Each person with an account will be prompted for a password on first login.

On the second screen, the user can setup more accounts. It automatically appends the "_Admin" suffix to the account name of the primary administrator, and also fills the first user account with the original username, and assigns it a user role.



To begin, click your user name

This is the login screen a user would see. It separates regular user accounts from the administrative accounts as a part of the effort to encourage users to login as a user role for day to day tasks.



Welcome to Mockingbird

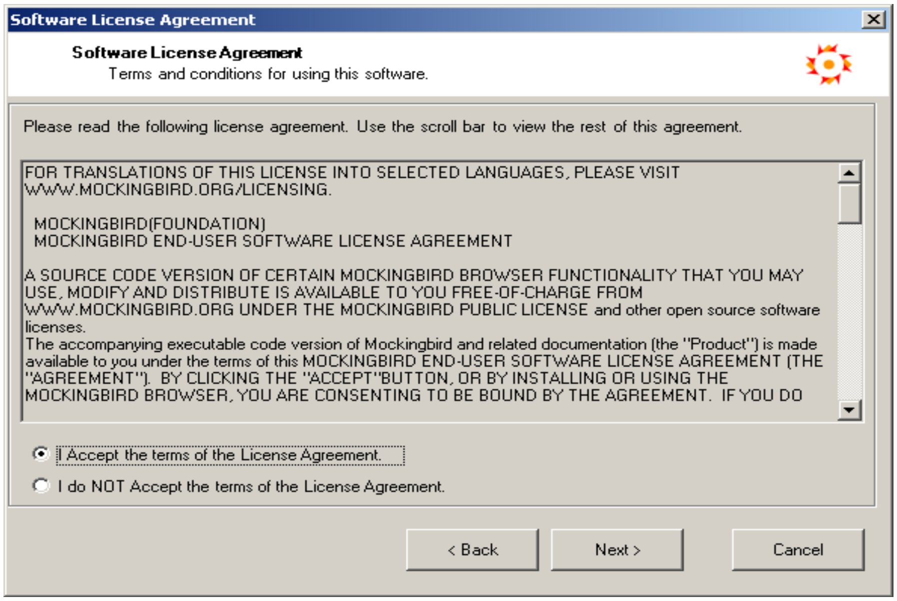
You are about to install Mockingbird 1.0.1

It is strongly recommended that you exit all Windows programs before runing this Setup program.

Click Next to continue the Setup program or Cancel to exit.

Click Next to continue installing Mockingbird 1.0.1.

Mockingbird is the software package a participants need to setup as one of the tasks. This is the initial screen of the install shield. This install shield mimics typical install shields we see for installing software, so that it gives users a familiar look and feel.



Software License Agreement

Software License Agreement

Terms and conditions for using this software.

Please read the following license agreement. Use the scroll bar to view the rest of this agreement.

FOR TRANSLATIONS OF THIS LICENSE INTO SELECTED LANGUAGES, PLEASE VISIT WWW.MOCKINGBIRD.ORG/LICENSING.

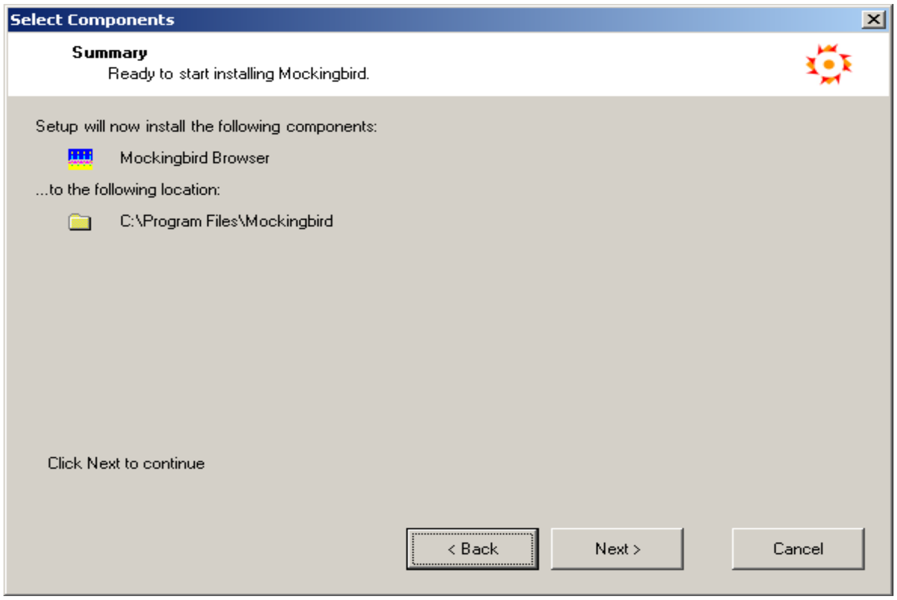
MOCKINGBIRD(FOUNDATION)
MOCKINGBIRD END-USER SOFTWARE LICENSE AGREEMENT

A SOURCE CODE VERSION OF CERTAIN MOCKINGBIRD BROWSER FUNCTIONALITY THAT YOU MAY USE, MODIFY AND DISTRIBUTE IS AVAILABLE TO YOU FREE-OF-CHARGE FROM WWW.MOCKINGBIRD.ORG UNDER THE MOCKINGBIRD PUBLIC LICENSE and other open source software licenses.
The accompanying executable code version of Mockingbird and related documentation (the "Product") is made available to you under the terms of this MOCKINGBIRD END-USER SOFTWARE LICENSE AGREEMENT (THE "AGREEMENT"). BY CLICKING THE "ACCEPT"BUTTON, OR BY INSTALLING OR USING THE MOCKINGBIRD BROWSER, YOU ARE CONSENTING TO BE BOUND BY THE AGREEMENT. IF YOU DO

● I Accept the terms of the License Agreement.

○ I do NOT Accept the terms of the License Agreement.

This screen displays the End User License Agreement (EULA), and the wording is quite typical for software nowadays.



Select Components

Summary

Ready to start installing Mockingbird.

Setup will now install the following components:

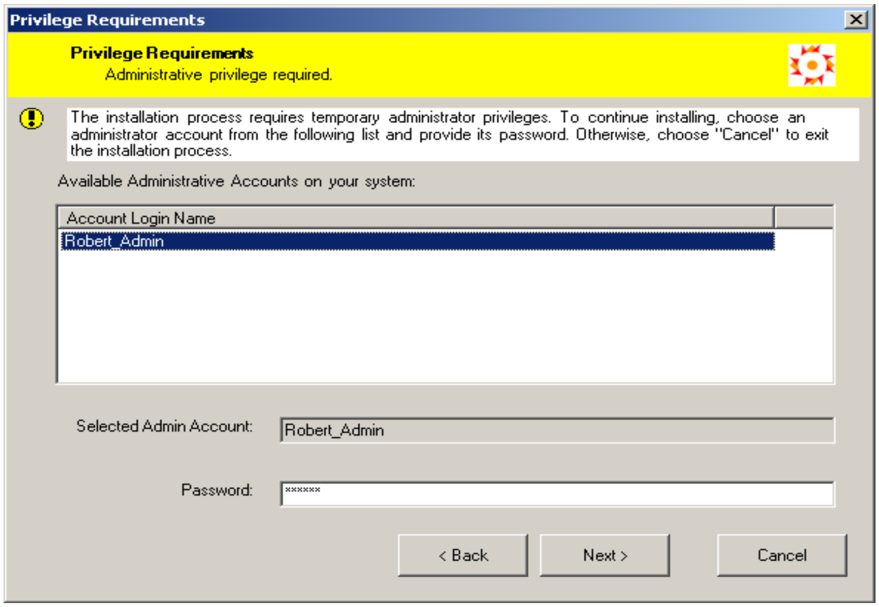
Mockingbird Browser

...to the following location:

C:\Program Files\Mockingbird

Click Next to continue

This is a summary screen before the installation. This can also be found in many install shields.



Privilege Requirements

Privilege Requirements

Administrative privilege required.

The installation process requires temporary administrator privileges. To continue installing, choose an administrator account from the following list and provide its password. Otherwise, choose "Cancel" to exit the installation process.

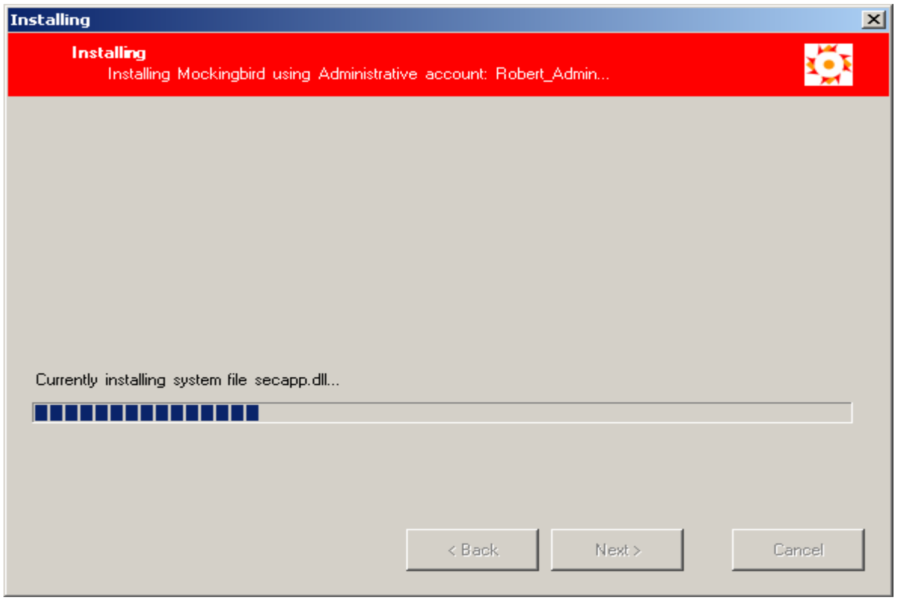
Available Administrative Accounts on your system:

Account Login Name
Robert_Admin

Selected Admin Account: Robert_Admin

Password: ******

This is a special screen built into our interface that is different from other install shields. When the user logs in as a regular user, this screen allows the user to run the installation as an administrator. It allows the user to choose an administrative account and runs under that account after the user has provided the correct password. This is also a part of our effort to encourage users to log in as regular users, even for installations.



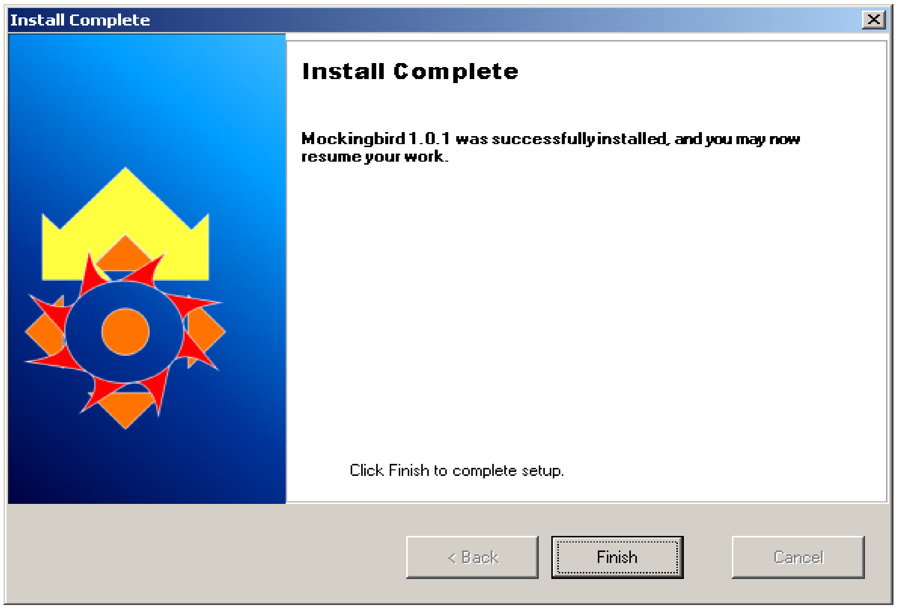
Installing

Installing

Installing Mockingbird using Administrative account: Robert_Admin...

Currently installing system file secapp.dll...

This screen shows the installation progress of the software. Its purpose is to add realism to our simulated install shield.

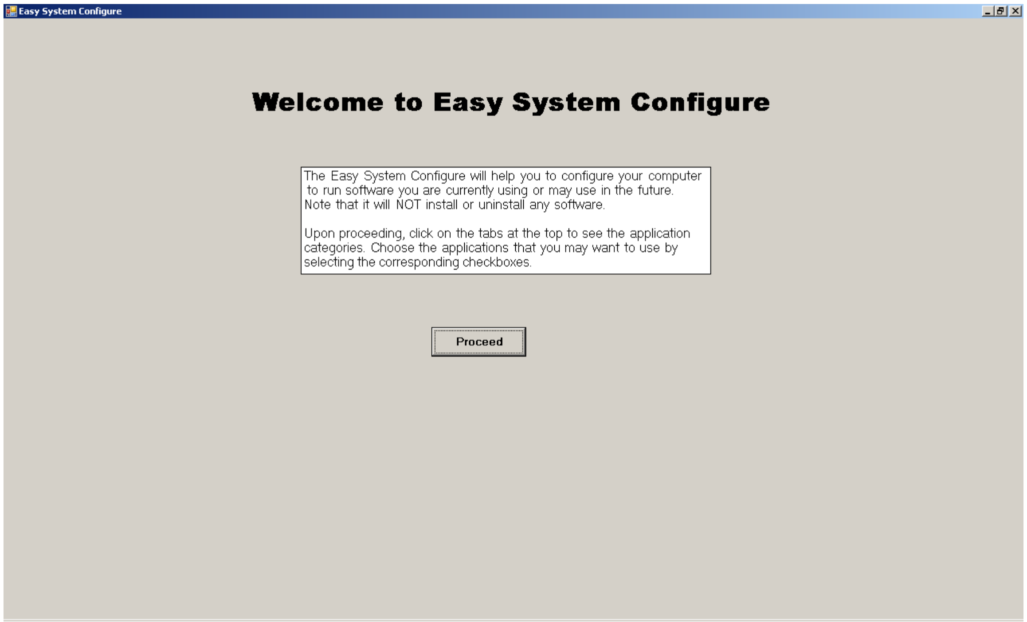


Install Complete

Mockingbird 1.0.1 was successfully installed, and you may now resume your work.

Click Finish to complete setup.

When the installation finishes, the user sees this screen, which is quite typical for install shields.

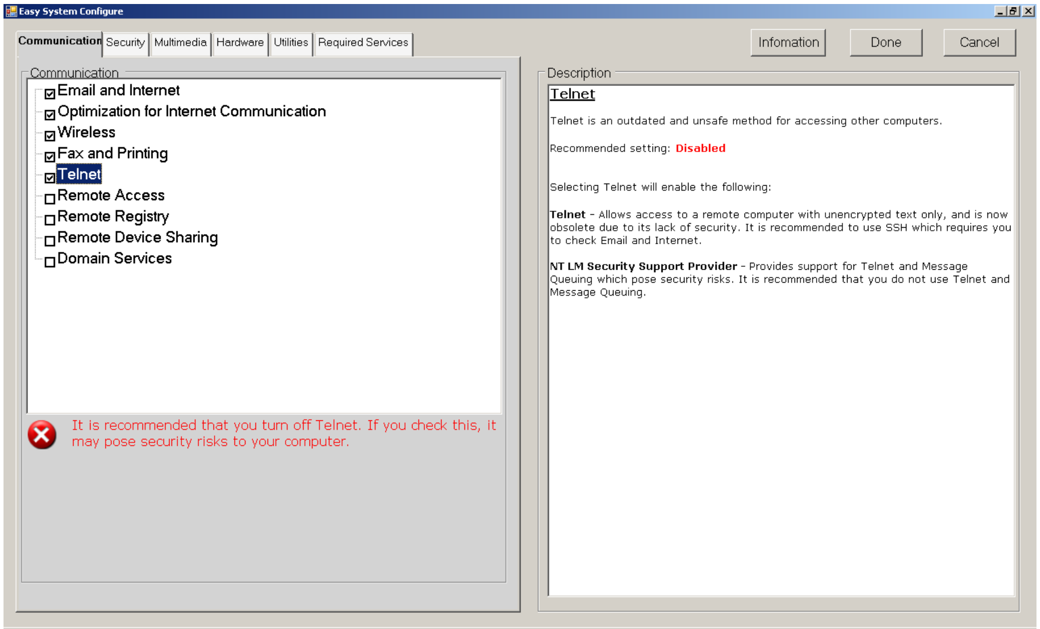


Welcome to Easy System Configure

The Easy System Configure will help you to configure your computer to run software you are currently using or may use in the future. Note that it will NOT install or uninstall any software.

Upon proceeding, click on the tabs at the top to see the application categories. Choose the applications that you may want to use by selecting the corresponding checkboxes.

Easy System Config is another interface we designed as an effort to ease user's configuration tasks. It groups low-level services into higher-level tasks to allow average computer users to pick the correct services to run while disabling those unneeded. We have also taken a lot of efforts to stay away from technical terms, and present those tasks using layman's term.



Communication

- ☑ Email and Internet
- ☑ Optimization for Internet Communication
- ☑ Wireless
- ☑ Fax and Printing
- ☑ Telnet
- □ Remote Access
- □ Remote Registry
- □ Remote Device Sharing
- □ Domain Services

It is recommended that you turn off Telnet. If you check this, it may pose security risks to your computer.

Description

Telnet

Telnet is an outdated and unsafe method for accessing other computers.

Recommended setting: **Disabled**

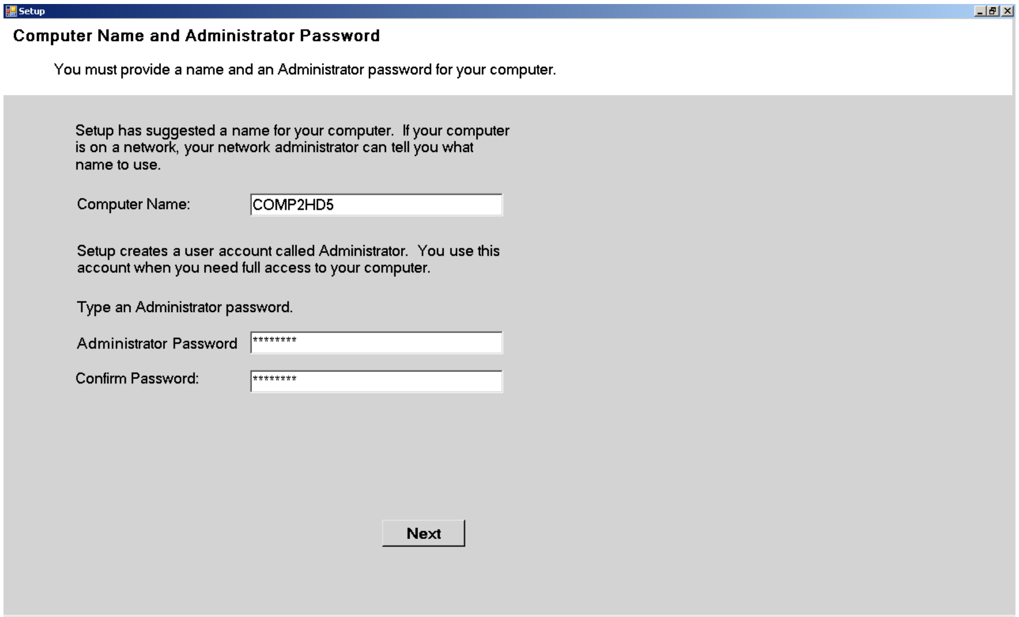
Selecting Telnet will enable the following:

Telnet - Allows access to a remote computer with unencrypted text only, and is now obsolete due to its lack of security. It is recommended to use SSH which requires you to check Email and Internet.

NT LM Security Support Provider - Provides support for Telnet and Message Queuing which pose security risks. It is recommended that you do not use Telnet and Message Queuing.

Each tab represents a major category of tasks a typical user would do on a computer. Under each category, there are checkboxes, each showing a high-level task. A user can simply enable/disable low-level services by pick and choose what tasks he/she would normally perform. When a task could present security risks to the system, our interface shows a warning message, telling the user to reconsider the action. The right half of the interface shows explanation of each task, as well as what services go under it. This can assist a user to make a decision when he/she encounters something unfamiliar.

APPENDIX D



Computer Name and Administrator Password

You must provide a name and an Administrator password for your computer.

Setup has suggested a name for your computer. If your computer is on a network, your network administrator can tell you what name to use.

Computer Name: COMP2HD5

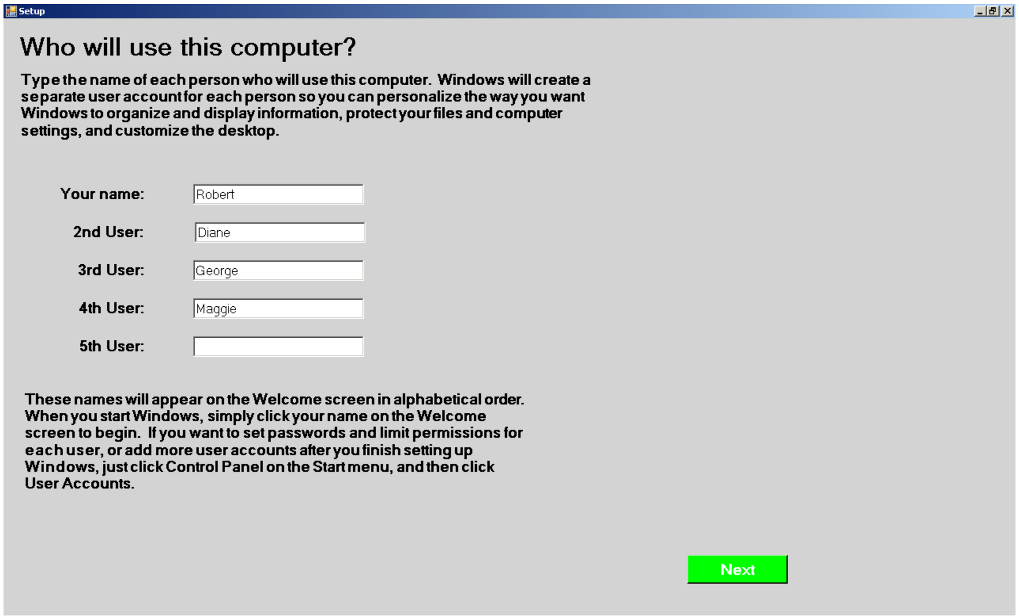
Setup creates a user account called Administrator. You use this account when you need full access to your computer.

Type an Administrator password.

Administrator Password ********

Confirm Password: ********

This screenshot shows a simulated screen displaying the installation process of Windows® XP. The user can provide a name for the computer, and also provide an optional password for the built-in administrative account: Administrator.



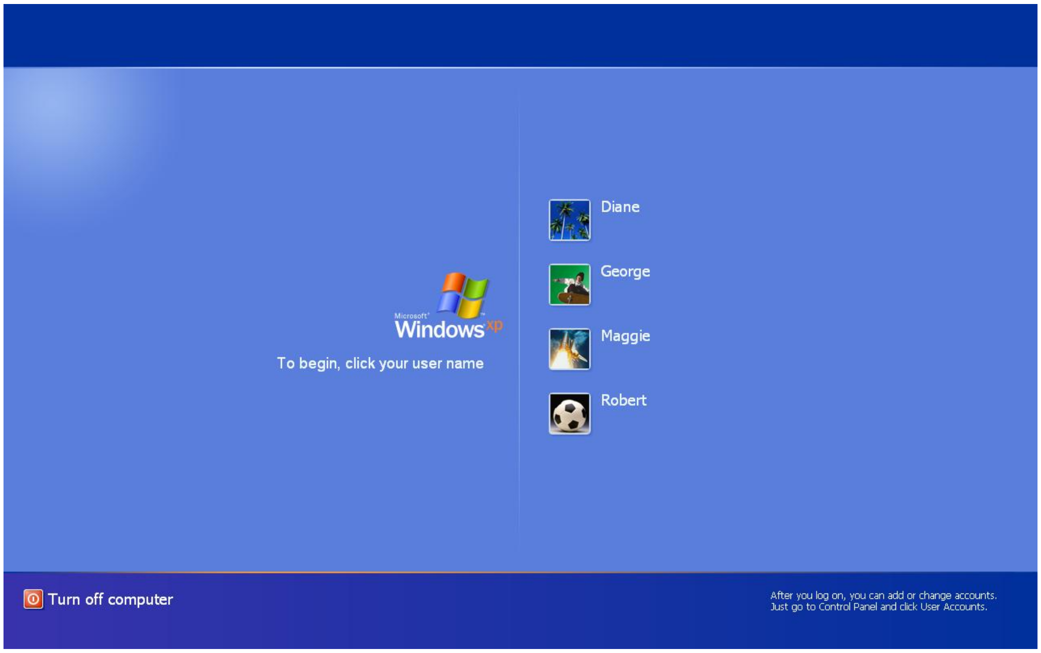
Who will use this computer?

Type the name of each person who will use this computer. Windows will create a separate user account for each person so you can personalize the way you want Windows to organize and display information, protect your files and computer settings, and customize the desktop.

Your name:	Robert
2nd User:	Diane
3rd User:	George
4th User:	Maggie
5th User:	

These names will appear on the Welcome screen in alphabetical order. When you start Windows, simply click your name on the Welcome screen to begin. If you want to set passwords and limit permissions for each user, or add more user accounts after you finish setting up Windows, just click Control Panel on the Start menu, and then click User Accounts.

This is a simulated screen displaying how user accounts are currently being setup in Windows® XP after the operating system has been installed. A user must provide at least 1 and up to 5 administrative accounts in addition to the built-in account. There is no way to setup an account with less privilege on this screen.



To begin, click your user name

Diane

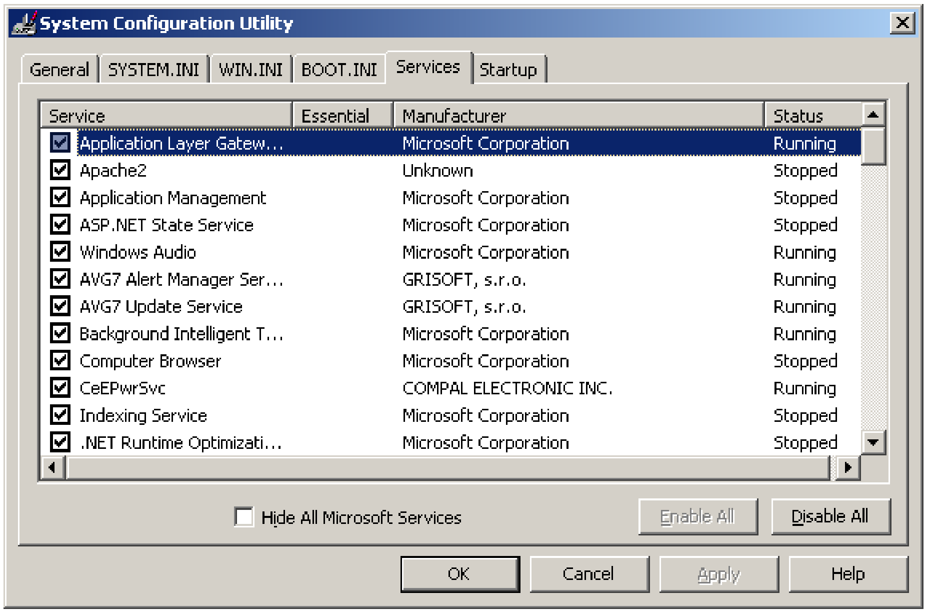
George

Maggie

Robert

After you log on, you can add or change accounts. Just go to Control Panel and click User Accounts.

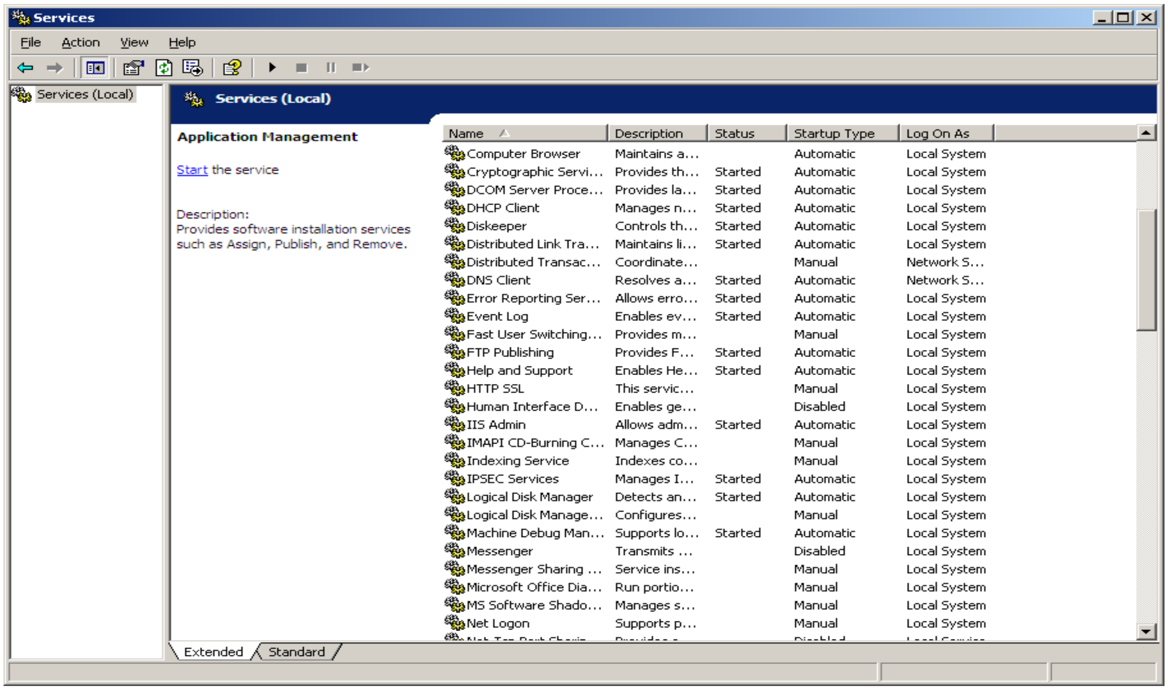
This is an actual login screen of Windows® XP. The different types of accounts are mixed and listed alphabetically. In addition, the built-in Administrator account cannot be accessed from this screen.



Service	Essential	Manufacturer	Status
☑ Application Layer Gatew…		Microsoft Corporation	Running
☑ Apache2		Unknown	Stopped
☑ Application Management		Microsoft Corporation	Stopped
☑ ASP.NET State Service		Microsoft Corporation	Stopped
☑ Windows Audio		Microsoft Corporation	Running
☑ AVG7 Alert Manager Ser…		GRISOFT, s.r.o.	Running
☑ AVG7 Update Service		GRISOFT, s.r.o.	Running
☑ Background Intelligent T…		Microsoft Corporation	Running
☑ Computer Browser		Microsoft Corporation	Stopped
☑ CeEPwrSvc		COMPAL ELECTRONIC INC.	Running
☑ Indexing Service		Microsoft Corporation	Stopped
☑ .NET Runtime Optimizati…		Microsoft Corporation	Stopped

☐ Hide All Microsoft Services

This is a screenshot of the actual "System Configuration Utility" tool. This tool does not have any accessible icons and can only be accessed by executing the command "msconfig" in the Run dialog. It combines several Windows startup-related things, and the services list is shown here. A user can enable or disable a service by simply using the checkboxes; however, it is rather difficult for an average home user to know exactly what each service is from here.



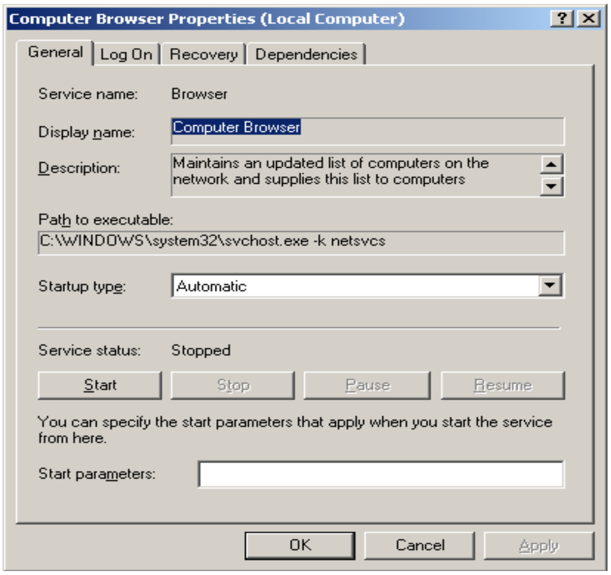
Application Management

Start the service

Description:
Provides software installation services such as Assign, Publish, and Remove.

Name	Description	Status	Startup Type	Log On As
Computer Browser	Maintains a…		Automatic	Local System
Cryptographic Servi…	Provides th…	Started	Automatic	Local System
DCOM Server Proce…	Provides la…	Started	Automatic	Local System
DHCP Client	Manages n…	Started	Automatic	Local System
Diskeeper	Controls th…	Started	Automatic	Local System
Distributed Link Tra…	Maintains li…	Started	Automatic	Local System
Distributed Transac…	Coordinate…		Manual	Network S…
DNS Client	Resolves a…	Started	Automatic	Network S…
Error Reporting Ser…	Allows erro…	Started	Automatic	Local System
Event Log	Enables ev…	Started	Automatic	Local System
Fast User Switching…	Provides m…		Manual	Local System
FTP Publishing	Provides F…	Started	Automatic	Local System
Help and Support	Enables He…	Started	Automatic	Local System
HTTP SSL	This servic…		Manual	Local System
Human Interface D…	Enables ge…		Disabled	Local System
IIS Admin	Allows adm…	Started	Automatic	Local System
IMAPI CD-Burning C…	Manages C…		Manual	Local System
Indexing Service	Indexes co…		Manual	Local System
IPSEC Services	Manages I…	Started	Automatic	Local System
Logical Disk Manager	Detects an…	Started	Automatic	Local System
Logical Disk Manage…	Configures…		Manual	Local System
Machine Debug Man…	Supports lo…	Started	Automatic	Local System
Messenger	Transmits …		Disabled	Local System
Messenger Sharing …	Service ins…		Manual	Local System
Microsoft Office Dia…	Run portio…		Manual	Local System
MS Software Shado…	Manages s…		Manual	Local System
Net Logon	Supports p…		Manual	Local System

This is a screenshot of the more elaborated "Services" tool. This screen is similar to the one in "System Configuration Utility" and lists all the services as well as their status.



Computer Browser Properties (Local Computer)

Service name: Browser

Display name: Computer Browser

Description: Maintains an updated list of computers on the network and supplies this list to computers

Path to executable:
C:\WINDOWS\system32\svchost.exe -k netsvcs

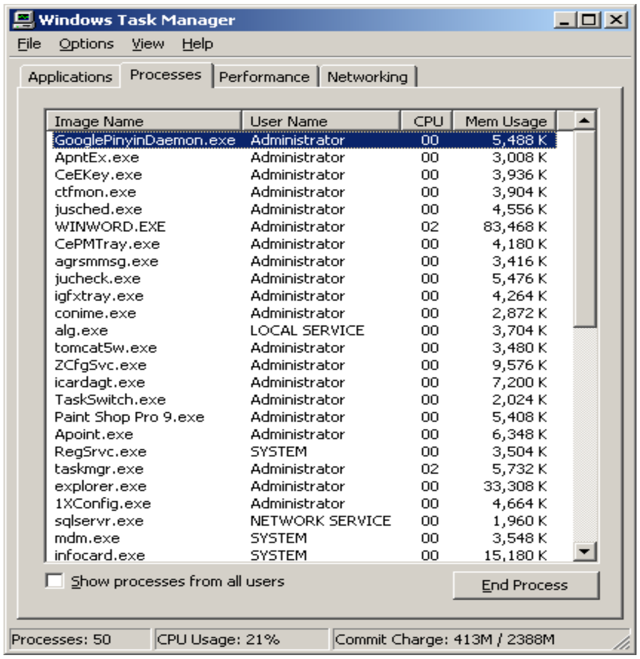
Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters:

This is the property sheet of each service in the "Services" tool. It has a lot of details about each service, including a description. However, to an average home user, it is still almost as good as nothing, because the descriptions are either vague or highly technical.



Windows Task Manager

Image Name	User Name	CPU	Mem Usage
GooglePinyinDaemon.exe	Administrator	00	5,488 K
ApntEx.exe	Administrator	00	3,008 K
CeEKey.exe	Administrator	00	3,936 K
ctfmon.exe	Administrator	00	3,904 K
jusched.exe	Administrator	00	4,556 K
WINWORD.EXE	Administrator	02	83,468 K
CePMTray.exe	Administrator	00	4,180 K
agrsmmsg.exe	Administrator	00	3,416 K
jucheck.exe	Administrator	00	5,476 K
igfxtray.exe	Administrator	00	4,264 K
conime.exe	Administrator	00	2,872 K
alg.exe	LOCAL SERVICE	00	3,704 K
tomcat5w.exe	Administrator	00	3,480 K
ZCfgSvc.exe	Administrator	00	9,576 K
icardagt.exe	Administrator	00	7,200 K
TaskSwitch.exe	Administrator	00	2,024 K
Paint Shop Pro 9.exe	Administrator	00	5,408 K
Apoint.exe	Administrator	00	6,348 K
RegSrvc.exe	SYSTEM	00	3,504 K
taskmgr.exe	Administrator	02	5,732 K
explorer.exe	Administrator	00	33,308 K
1XConfig.exe	Administrator	00	4,664 K
sqlservr.exe	NETWORK SERVICE	00	1,960 K
mdm.exe	SYSTEM	00	3,548 K
infocard.exe	SYSTEM	00	15,180 K

☐ Show processes from all users

Processes: 50 | CPU Usage: 21% | Commit Charge: 413M / 2388M

This is a screenshot of the "Windows Task Manager" tool. It shows the dynamic state of the machine rather than configuration. From our study, it seems that more users are familiar with this one than the other two, and they usually use this to kill an application that hangs.

APPENDIX E

We have used the statistical software R to perform some statistical analysis and tests on the data we collected.

Test 1. We have not found any significant correlation between a participant's knowledge related to computer use and his/her performance in our test.

```
> cor.test(myod$post_raw, myod$perf)

        Pearson's product-moment correlation

data:  myod$post_raw and myod$perf
t = 0.098, df = 20, p-value = 0.9229
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4034181  0.4394656
sample estimates:
       cor
0.02191827
```

Test 1(a). Test of correlation between participants' knowledge and their performance

```
> cor.test(myod$post_raw[as.character(myod$group)=="control"],
myod$perf[as.character(myod$group)=="control"])

        Pearson's product-moment correlation

data:  myod$post_raw[as.character(myod$group) == "control"] and
myod$perf[as.character(myod$group) == "control"]
t = 1.0116, df = 9, p-value = 0.3381
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3468203  0.7715228
sample estimates:
      cor
0.3195332
```

Test 1(b). Test of correlation between control group members' knowledge and performance

```
> cor.test(myod$post_raw[as.character(myod$group)=="test"],
myod$perf[as.character(myod$group)=="test"])

        Pearson's product-moment correlation

data:  myod$post_raw[as.character(myod$group) == "test"] and
myod$perf[as.character(myod$group) == "test"]
t = -0.5227, df = 9, p-value = 0.6138
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6994943  0.4773867
sample estimates:
       cor
-0.1716418
```

Test 1(c). Test of correlation between test group members' knowledge and performance

Test 2. By performing a t-test, we have found that between the two groups, test and control, there is a significant difference in the time takes to complete the tasks at the 99.9% confidence level, and the p-value is 0.00045.

```
> t.test(time$Test, time$Control)

        Welch Two Sample t-test

data:  time$Test and time$Control
t = -4.813, df = 11.736, p-value = 0.0004515
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -28.53464 -10.71990
sample estimates:
mean of x mean of y 
 10.27273  29.90000
```

Test 2. Two-sided t-test on difference in time between test group and control group

Test 3. Further test shows that the time the test group takes to complete the tasks is significantly less than that of the control group at the 99.9% confidence level, with a p-value of 0.0002.

```
> t.test(time$Test, time$Control, alternative="less")

        Welch Two Sample t-test

data:  time$Test and time$Control
t = -4.813, df = 11.736, p-value = 0.0002257
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
     -Inf -12.3455
sample estimates:
mean of x mean of y 
 10.27273  29.90000
```

Test 3. One-sided t-test on time used between test group and control group

Test 4. Since Task 5 seemed to be the most demanding as well as time-consuming task among all, and many participants (especially those in Control group) were having difficulties finishing the task, we would like to perform a test to see if there is any correlation between participants' knowledge and their performance on this task. As it turned out, there was no significant correlation between their knowledge and their performance.

```
> cor.test(myod$post_raw[as.character(myod$group)=="control"],
myod$services[as.character(myod$group)=="control"])


        Pearson's product-moment correlation


data:  myod$post_raw[as.character(myod$group) == "control"] and
myod$services[as.character(myod$group) == "control"]
t = 1.5881, df = 9, p-value = 0.1467
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1835230  0.8337393
sample estimates:
     cor
0.467859
```

Test 4(a). Test of correlation between control group members' knowledge and their performance on Task 5

```
> cor.test(myod$post_raw[as.character(myod$group)=="test"],
myod$services[as.character(myod$group)=="test"])


        Pearson's product-moment correlation


data:  myod$post_raw[as.character(myod$group) == "test"] and
myod$services[as.character(myod$group) == "test"]
t = -0.9718, df = 9, p-value = 0.3565
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7663696  0.3578633
sample estimates:
       cor
-0.3081669
```

Test 4(b). Test of correlation between test group members' knowledge and their performance on Task 5

```
> cor.test(myod$post_raw, myod$services)


        Pearson's product-moment correlation


data:  myod$post_raw and myod$services
t = 1.0151, df = 20, p-value = 0.3222
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2208702  0.5880777
```

```
sample estimates:
      cor 
0.2213506 
```

Test 4(c). Test of correlation between all participants' knowledge and their performance on Task 5

Test 5. We also wanted to see how participants perform using Windows toolset versus our Easy System Config while performing Task 5, so we performed a t-test comparing their performances on Task 5. The results indicate that the participants who use Windows tools perform worse than those who use our Easy System Config tool.

```
> t.test(myod$services[as.character(myod$group)=="control"], 
myod$services[as.character(myod$group)=="test"], alternative="less")


        Welch Two Sample t-test


data:  myod$services[as.character(myod$group) == "control"] and 
myod$services[as.character(myod$group) == "test"] 
t = -6.0822, df = 11.264, p-value = 3.589e-05
alternative hypothesis: true difference in means is less than 0 
95 percent confidence interval:
      -Inf -29.43280 
sample estimates:
mean of x mean of y 
 22.36364  64.09091 
```

Test 5. Two-sided t-test of difference on Task 5 performance between control group and test group

Test 6. Before the tasks begin, each participant is asked to fill out a survey to indicate the level of computer knowledge they think they have. From their results, we have also summarized a score to measure how well each participant actually performs. A set of t-tests show that there is indeed some significant differences between how knowledgeable they think they are and how knowledgeable they actually are, and all of which are significant at the confidence level of 99%.

```
> t.test(prepost$Pre, prepost$Post)


        Welch Two Sample t-test


data:  prepost$Pre and prepost$Post 
t = 4.9818, df = 39.105, p-value = 1.319e-05
alternative hypothesis: true difference in means is not equal to 0 
95 percent confidence interval:
 11.65079 27.57648 
sample estimates:
mean of x mean of y 
 64.72727  45.11364 
```

Test 6 (a). Two-sided t-test of difference between participants' self-assessed and actual levels of knowledge

```
> t.test(prepost.test$Pre, prepost.test$Post)
```

```
        Welch Two Sample t-test

data:  prepost.test$Pre and prepost.test$Post
t = 3.5229, df = 15.694, p-value = 0.002896
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  8.270972 33.365391
sample estimates:
mean of x mean of y
 66.36364  45.54545
```

Test 6(b). Two-sided t-test of difference between test group members' self-assessed and actual levels of knowledge

```
> t.test(prepost.ctrl$Pre, prepost.ctrl$Post)

        Welch Two Sample t-test

data:  prepost.ctrl$Pre and prepost.ctrl$Post
t = 3.3831, df = 20, p-value = 0.002954
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  7.058396 29.759786
sample estimates:
mean of x mean of y
 63.09091  44.68182
```

Test 6(c). Two-sided t-test of difference between control group members' self-assessed and actual levels of knowledge

Test 7. We also performed a test to compare the actual knowledge level between those in Control group and those in Test group. As we hoped, there was no significant difference between the two, so in other word, the two groups have the same level of knowledge regarding computers.

```
> t.test(myod$post_raw[as.character(myod$group)=="control"],
myod$post_raw[as.character(myod$group)=="test"])

        Welch Two Sample t-test

data:  myod$post_raw[as.character(myod$group) == "control"] and
myod$post_raw[as.character(myod$group) == "test"]
t = -0.4745, df = 19.33, p-value = 0.6404
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.862532  4.953441
sample estimates:
```

```
mean of x mean of y
 19.63636  21.09091
```

Test 7. Two-sided t-test on levels of knowledge between control group and test group

Test 8. Then we would like to see how each group of participants perform in different types of tasks, where the types are defined in Table 1. The t-test of their performance on account creation shows that Test group performs significantly better than Control group, and the p-value is 7.75e-05, which is almost 0.

```
> t.test(myod$rate_1[as.character(myod$group)=="control"],
myod$rate_1[as.character(myod$group)=="test"], alternative="less")


        Welch Two Sample t-test


data:  myod$rate_1[as.character(myod$group) == "control"] and
myod$rate_1[as.character(myod$group) == "test"]
t = -4.6483, df = 20, p-value = 7.746e-05
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.6289611
sample estimates:
mean of x mean of y
0.3636364 1.3636364
```

Test 8. One-sided t-test on Task 1 performance between control group and test group

Test 9. For tasks related to which account they would use to login, we have combined Task 2a and Task 3. Once again, the t-test results indicate that Test group performs significantly better than Control group, with a p-value of 0.0056.

```
> t.test(myod$login[as.character(myod$group)=="control"],
myod$login[as.character(myod$group)=="test"], alternative="less")


        Welch Two Sample t-test


data:  myod$login[as.character(myod$group) == "control"] and
myod$login[as.character(myod$group) == "test"]
t = -3.1053, df = 10, p-value = 0.005575
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.6812707
sample estimates:
mean of x mean of y
 0.000000  1.636364
```

Test 9. One-sided t-test on login behavior (Task 2a and 3 combined) between control group and test group

Test 10. Since there were a few incidents where the participants suspected that our researcher was actually not a participant, we wanted to see if it had any effects on their performance. We are only able to perform the t-test on the control group because only 1 participant from the test group raised suspicion. The test results showed that there was some significant difference in Task1 with a p-value of 0.03002, but for tasks 2a, 2b, 3, and 5, there did not seem to be any significant differences.

```
> t.test(tt$rate_1[as.character(tt$group.1)=="x" &
as.character(tt$group)=="control"],tt$rate_1[as.character(tt$group.1)=="y" &
as.character(tt$group)=="control"])


        Welch Two Sample t-test


data:  tt$rate_1[as.character(tt$group.1) == "x" & as.character(tt$group) ==
and tt$rate_1[as.character(tt$group.1) == "y" & as.character(tt$group) ==
"control"] and     "control"]
t = -2.8284, df = 6, p-value = 0.03002
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.06577942 -0.07707772
sample estimates:
mean of x mean of y
0.0000000 0.5714286
```

Test 10(a). Two-sided t-test on performance on Task 1 between suspecting participants and non-suspecting participants in control group

```
> t.test(tt$total_time[as.character(tt$group.1)=="x" &
as.character(tt$group)=="control"],
tt$total_time[as.character(tt$group.1)=="y" &
as.character(tt$group)=="control"])


        Welch Two Sample t-test


data:  tt$total_time[as.character(tt$group.1) == "x" & as.character(tt$group)
==  and tt$total_time[as.character(tt$group.1) == "y" & as.character(tt$group)
==      "control"] and     "control"]
t = -1.5511, df = 5.636, p-value = 0.175
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -37.179082   8.607653
sample estimates:
mean of x mean of y
 36.00000  50.28571
```

Test 10(b). Two-sided t-test on performance on tasks 2a, 2b, 3, and 5 between suspecting participants and non-suspecting participants in control group

REFERENCES

[1] Balfanz, D., et al. In search of usable security – five lessons from the field. *IEEE Jour. on Security and Privacy*, 2(5), 2004.

[2] Balfanz, D., et al. Network-in-a-box: how to set up a secure wireless network in under a minute. *13th USENIX Security Symp.*, 2004.

[3] Dhamija, R., and Tygar, J. D. The battle against phishing: dynamic security skins. *Symp.on Usable Privacy and Security*, 2005.

[4] Dhamija, R., et al. Why phishing works. *Proc. of the Conf. on Human Factors in Computing Systems*, 2006.

[5] Dourish, P., and Redmiles, D. *Approach to usable security based on event monitoring and visualization*. ISBN: 1-58113-598-X

[6] Egelman, S., et al. Studying the impact of privacy information on online purchase decisions. *Wkshp on Privacy and HCI: Methodologies for Studying Privacy Issues at CHI 06*, 2006.

[7] Garfinkel, S. L., and Miller, R. C. Johnny 2: a user test of key continuity management with S/MIME and outlook express. *Symp. on Usable Privacy and Security (SOUPS)*, 2005.

[8] Garfinkel, S. L., et al. How to make secure email easier to use. *CHI 2005: Technology, Safety, Community*, Portland, Oregon, 2005.

[9] Garfinkel, S. L., et al. Views, reactions and impact of digitally-signed mail in e-commerce. *Financial Cryptography and Data Security 9th Int'l Conf.*, 2005, Roseau, the Commonwealth of Dominica.

[10] Geng, W., et al. Usable firewall configuration. *Proc. of the 3rd Ann. Conf. on Privacy, Security and Trust (PST'05)*, 2005.

[11] Good, N. S., and Krekelberg, A. Usability and privacy: a study of kazaa P2P file-sharing. In *Proc. of the ACM Conf. on Human Factors in Computing Systems (CHI 2003)*, Ft. Lauderdale, Florida.

[12] Hardee, J. B., et al. To download or not to download: an examination of computer security decision making. *Interaction,* 13(3): 32-37, May 2006. ACM Press.

[13] Jackson, C., et al. *An evaluation of extended validation and picture-in-picture phishing attacks*, http://www.usablesecurity.org/papers/jackson.pdf

[14] Kumaraguru, P., et al. Protecting people from phishing: the design and evaluation of an embedded training email system. Tech. Report CMU-CyLab-06-017, CyLab, Carnegie Mellon University, November 2006.

[15] Nielsen, J. *Usability engineering.* Academic Press, 1993.

[16] Norman, D. A. Design rules based on analyses of human error. *Comm. of the ACM*, 26, April 1983.

[17] Orne, M. T. On the social psychology of the psychological experiment: with particular reference to demand characteristics and their implications. *Prevention & Treatment,* 5(1), Oct. 2002.

[18] Saltzer, J. H., and Schroeder, M. D. The protection of information in computer systems. *Proc. of the IEEE,* 64: 1278-1308, Sept. 1975.

[19] Sasse, M. A. *Eliciting and describing user's models of computer systems*. PhD thesis, School of Computer Science, University of Birmingham, April 1997.

[20] Schechter, S. E., et al. The emperor's new security indicators: an evaluation of website authentication and the effect of role playing on usability studies. To appear in the *2007 IEEE Symp. on Security and Privacy*, Oakland, CA, 2007.

[21] Smetters, D. K., and Grinter, R. E. Moving from the design of usable security technologies to the design of useful secure applications. *New Security Paradigms Wkshp*, 2002.

[22] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In *8th USENIX Security Symp.*,pp. 169-184, 1999.

[23] Wool, A. The use and usability of direction-based filtering in firewalls. *Computers & Security,* 23(6): 459-468. Sept. 2004, Elsevier Science B.V.

[24] Wu, M., et al. Do security toolbars actually prevent phishing attacks? In M. Jakobsson and S. Myers, eds, *Phishing and counter-measures: understanding the increasing problem of electronic identity theft.* Wiley, 2006, to appear.

[25] Wu, M., et al. Web Wallet: Preventing phishing attacks by revealing user intentions. In *Proc. of the 2006 Symp. on Usable Privacy and Security*, 2006, Pittsburgh, PA.

[26] Yee, K. P. User interaction design for secure systems. In *Proc of the 4th Int'l Conf. on Information and Communications Security (LNCS 2513)*, pp. 278 – 290. Springer-Verlag, 2002.