Computers & Security Vol.20, No.7, pp.620-634, 2001
Copyright © 2001 Elsevier Science Limited
Printed in Great Britain. All rights reserved
0167-4048/01$20.00

Usability and Security An Appraisal of Usability Issues in Information Security Methods

E. Eugene Schultz[1], Robert W. Proctor[2], Mei-Ching Lien[2], Gavriel Salvendy[3]

[1]*Global Integrity Corporation, West Lafayette, IN 47906*
[2]*Dept. of Psychological Sciences, Purdue University, West Lafayette, IN 47907*
[3]*School of Industrial Engineering, Purdue University, West Lafayette, IN 47907*

Abstract

In the modern multi-user computer environment, Internet-capable network servers provide connectivity that allows a large portion of the user population to access information at the desktop from sources around the world. Because of the ease with which information can be accessed, computer security breaches may occur unless systems and restricted information stored therein are kept secure. Breaches of security can have serious consequences, including theft of confidential corporate documents, compromise of intellectual property, unauthorized modification of systems and data, denial of service, and others. Considerable research has been conducted on threats to security.

Numerous sophisticated security methods have been developed, many of which rely on individuals to implement and use them. However, these methods may not accomplish their intended objectives if they are not used properly. Despite the apparent influence of usability, surprisingly little research has been conducted on the trade-off between usability and the degree of security provided by various information security methods. In the present paper, we review the various information security methods that are used, appraise the usability issues, and develop a taxonomy to organize these issues. The intent is to make a strong case for the need for systematic usability analyses and for the development of usability metrics for information security.

An Appraisal of Usability Issues in Information Security Methods

Human Factors literature abounds with usability research and analysis concerning human performance in a large variety of different settings, including driving automobiles, interacting with graphics displays, manipulating controls at power generation plants, reading help messages on computer display terminals,

and many others (Salvendy, 1997). A notable exception is research and analysis regarding humans engaged in tasks designed to elevate the security of computing systems. Information security has come to be recognized as increasingly important because the Internet allows a potentially large number of unauthorized users to access and possibly alter information from around the world. The relative lack of research on the topic of usability with regard to security controls is somewhat surprising given that their appropriate use requires human interactions and users, broadly defined to include both end-users and system administrators, typically are considered to be a limiting factor in security. This dearth of research is even more surprising given that the study of human-computer interaction (HCI) has probably been the single most active area of research within the field of Human Factors in the last decade (see, e.g., Helander, Landauer, & Prabhu, 1997).

Human Factors in Information Security

The lack of consideration of Human Factors issues in information security is not in any way related to a scarcity of analyses of security-related threats to computers and networks. Threats to security on the Internet are, for example, many and varied (see, e.g., Bernstein, Bhimani, Schultz, & Siegel, 1996). These threats include eavesdropping on user sessions, masquerading as another user, manipulating data without authorization, misrouting communications, and repudiation (e.g., denying that a recently initiated electronic commerce transaction was indeed initiated), among others (see Table 1). Not surprisingly, considerable information concerning these and many other threats, and how to counter them, has been published.

A logical starting point in investigating the relationship between Human Factors and security-related controls is to define "information security" (sometimes also known by the older and now somewhat less used term, "computer security"). Information security in its most basic sense is protecting computer-related assets such as computers, networks, data, programs, and the hardware components (e.g., display terminals, printers, and pointing devices; Gollmann, 1999). "Information security controls" (also called "information security mechanisms") are techniques and procedures used to reduce the likelihood that security-related threats will result in unauthorized disclosure or possession of information, loss of integrity of systems and/or data, and disruption of availability and/or accessibility of systems (Parker, 1991). Numerous sophisticated security control mechanisms, examples of which are passwords used to log on to systems, file permissions, and cryptographic devices, exist today.

Many (if not most) security-related controls rely on individuals to implement and deploy them. Considering the case of file access permissions, default permissions on newly installed systems are generally not adequate from a security perspective. For example, in the Windows NT operating system, the critical SYSTEM32 directory within the C: partition by default allows FULL CONTROL to everyone.

Default permissions such as these must be changed if a system and its resources are even to be minimally secure. Any control measures may not, however, accomplish their intended objectives if they are improperly installed and/or maintained. A significant concern is that many security measures, examples of which are provided shortly, unduly inconvenience the user despite the fact that users' acceptance of security measures and their willingness and ability to follow required procedures is necessary if the measures are to be effective. However, not much is known about the effects of Human Factors on performance of tasks related to securing systems and networks.

Although people's usage of security mechanisms is currently neither well understood nor adequately documented, anecdotal evidence suggests that people's use of these mechanisms is far from optimal.

Users have long been regarded as the weak link in information security. Systems administrators, for example, must usually inspect system logs to determine whether or not unauthorized activity has occurred. Given that typical logs display line-after-line of monochrome text that is also uniform in size and font and arranged in columns, it is hardly any

surprise that studies such as one by Van Wyk (1994) indicate that system administrators detect fewer than 20% of break-ins into their systems. Despite this and similar evidence, very little research has been conducted on the usability issues associated with information security methods.

User Resistance to Information Security Measures

As important as it is, performance is not the only consideration in Human Factors design and analysis. Another equally important consideration is user acceptance, or, from a reverse perspective, user resistance towards systems with which they must interact (Turnage, 1990). In general, systems with poor usability design tend to evoke a greater degree of user resistance (e.g., Al-Ghatani & King, 1999; Markus, 1983). This resistance manifests itself in various manners, including passive resistance, negative verbal behaviour, reluctance to perform tasks, failure to pay sustained attention to tasks, actions that cause damage to system components, and many others (Martinko, Henry, & Zmud, 1996). Overcoming user resistance through Human Factors design, thus, is another critical goal within the field of Human Factors.

Information security professionals are well too aware that the controls they prescribe for systems often elicit less than enthusiastic reactions from both management and the user community. Although the reasons are numerous, we contend that a major reason for user resistance is poor usability design. When users must interact with systems to implement, use, and/or maintain security, the usability of the interaction tasks is a critical factor in determining user willingness to engage in these tasks. Consider, for example, a fictitious system for which numerous security-related controls have been implemented. One of the controls is geared towards continuously determining the identity of the user after the user has logged on to the system. Suppose that dialogue boxes that query the user for different pieces of personal information (e.g., social security number, mother's maiden name, and so forth) pop up on the user's display terminal at random intervals. Application of known principles of Human Factors would easily lead to the prediction that the layout of the dialogue box and the wording and visual appearance of the dialogue boxes would be critical to user acceptance. Perhaps more importantly, however, the fact that normal user interaction tasks are disrupted by the need to perform additional interaction tasks (e.g., filling in fields in the dialogue boxes) would also lead to the prediction that users would quickly develop a strong dislike for interacting with this system. This system would, in effect, favour security at the cost of human usability. Certain users might also object to interacting with this system because having to enter information about oneself and one's family borders on infringement of privacy.

Signs of user resistance abound in the area of information security. System administrators neglect reading system logs. Users set up and use methods of remote access that bypass security measures called "firewalls," barriers between networks that filter and manage incoming network traffic. Organizations too often buy biometric devices (e.g., fingerprint readers) with the intent of using them to provide strong assurance of user identity, only to scrap these devices shortly afterwards because of the sheer volume of complaints. Yet, despite "the writing on the wall," no one has systematically investigated the nature of the human interaction component in performing security-related tasks.

Purpose

The purpose of this paper is to draw attention to the need to develop usability methods and metrics for information security controls. Specifically, the need exists to:

Establish which security method is best to use for which individuals and which jobs;
Improve the usability of existing security methods;
Develop and validate an instrument to quantitatively increase the usability of each security method, and derive an index of usability for each scenario;
Develop and validate an artificial intelligence-based software tool that will aid designers and users in determining how and what needs to be

done to increase the index of usability for a particular security method utilized for a specific reason by individuals having a set of known abilities and attributes.

Usability studies of security methods should be geared towards two objectives, namely to: (1) Increase the willingness of individuals to use the method; and (2) Ensure that those who choose to use the security method can do so with greater ease, less time, fewer errors, and higher satisfaction than otherwise would have been possible. The studies will also have some impact on increasing the number of people who are willing to use the security methods.

This paper explores the nature of these tasks from a Human Factors perspective and then presents a comprehensive analysis to characterize the tasks. Based on examination of the primary types of security-related controls, we delineate the tasks that involve human users. We then create a taxonomy that defines the major types of security-related user interaction tasks and the Human Factors considerations applicable to each. Constructing such a taxonomy is a necessary step in developing an organized research program toward addressing usability issues in security-related tasks. We also attempt to identify any relevant costs versus benefits not only from a Human Factors design perspective, but also from an operations and systems performance point-of-view.

Usability Issues in Security-Related Tasks

The major types of security controls that exist today apply to the following areas: Identification and authentication of a user, application, or system, helping ensure that only authorized users obtain access to systems, or that users cannot repudiate electronic transactions, or also possibly that intruders can be detected. These areas, the types of security-related threats that each counters, and some salient usability issues are listed in Table 1. In this section, we describe some of the methods used to implement these security controls and elaborate the usability issues associated with each method.

Identification and Authentication

Identification, one of the most fundamental areas of information security measures, means confirming the user's identity. Failure to confirm the user identity invites catastrophe. Consider, for example, an electronic banking transaction that does not require reasonable proof of identity. As a consequence of the potential for catastrophe, identification has, in fact, become increasingly important with the advent of electronic business; non-repudiation measures that prevent someone who recently ordered merchandise from denying having placed an order after the merchandise arrives are one of the most potentially valuable protections available to electronic merchants.

Authentication is closely related to identification in that identification is the first part of authentication. Authentication, however, goes farther in that it generally requires additional steps at a minimum, entry of the user's account name or possibly some other type of information. Authentication, however, is distinguished from identification mainly by its purpose, namely to allow or deny access to systems and/or networks. The main purpose of identification, in contrast, is simply to ensure that users are who they claim to be.

Passwords

The most common identification-related user task is entering a password when the appropriate prompt is displayed on the user's terminal. For example, when a user comes back from a coffee break or lunch, a screen saver that requires a password entry may keep the user from immediately using the computer. Consider the major user behaviours in such a task:

1. Visually sighting the dialogue box and the prompts and input field within;
2. Using a pointing device to align the cursor/pointer to the correct location;
3. Homing both hands at the keyboard;
4. Entering a string of alphanumeric characters;
5. Clicking on <OK> or pressing the <ENTER> key.

Provided that the interaction sequence is reasonably simple and intuitive (as in the above example), it is straightforward to predict that users will be able to accomplish this task reasonably rapidly, with only minor errors, and without major user resistance hurdles. As noted for the login example discussed for user resistance, the exception would be when identification dialogue boxes are presented multiple times during any given log-on session. A security-usability trade-off occurs in this case; having users identify themselves multiple times during a log-on session greatly reduces the likelihood that someone who has broken into a user account will be able to stay logged on. On the other hand, this (not so advisable) procedure would disrupt other ongoing task interaction sequences substantially.

The most frequent types of expected errors in using a password include:

1. Data entry errors;
2. Failure to remember one's password. Password memory is, in fact, the area in which security-usability trade-offs are most likely to occur because the easier the password is to remember, the more likely it is to be guessed or cracked (De Alvare & Schultz, 1988);
3. Errors (e.g., homing fingers on the wrong keys on the keyboard, pressing the wrong mouse button) resulting from the need to use more than one interaction device, e.g., a mouse and keyboard, while performing the different steps of this user interaction sequence.

Given that errors such as typing in a wrong letter can occur with relative ease, a criterion must be developed for reducing errors or allowable steps to recover from errors. For example, a way to reduce errors, but at the expense of an elevation of security risk, is to allow users to see the password they are typing in, instead of having it masked by asterisks. With the masked passwords, entry errors will be of less consequence if users are provided with a method of recovery if a typographical error is made.

Although username-password entry is the most common authentication method, in part because so many possible passwords are easy to remember, it at best provides only a relatively weak authentication mechanism. An unauthorized user can gain entry by

accessing electronic messages that contain passwords, cracking passwords, or entering successive username-password combinations until successful (unless the system has limits on the number of incorrect log-ons). For example, Klein (1990) collected UNIX password files containing nearly 14,000 encrypted passwords. Following a simple strategy, he was able to identify approximately 25% of the passwords. Consequently, this method of authentication must be replaced by, or supplemented with, other methods if elevated security is required. These other methods allow identification of the user with more certainty, but at the cost of being more difficult to use.

More secure forms of password verification. To remedy the problem of users choosing easy-to-crack passwords, administrators have implemented computer-generated passwords, which are chosen to satisfy specified criteria such as that a mixture of letter and digits be included. However, more errors are made with computer-generated passwords because the users have trouble remembering them. As a compromise to the first two approaches, proactive password checking (Stallings, 1995) allows a user to select a password; the system then checks to see if the password satisfies certain criteria (e.g., at least one uppercase and one lowercase English alphabet character in addition to a digit). If it does not, the password selection is rejected, and the user must choose a new password. This process will continue until the password screening tool allows a selected password. Another way to strengthen security is to make the knowledge that the user must demonstrate be more detailed. An example is one-time passwords, for which the user has a list of passwords computed ahead of time, each of which is good only once.

Clearly, there are many usability issues associated with the requirement that the user maintain and act on detailed knowledge, with the foremost being confusion concerning the particular password to be entered on any given log-on attempt. In addition, extra requirements for passwords (such as restrictions in

length) and the like may impose unnatural constraints on the user.

Digital signature

A somewhat more sophisticated identification task is signing a document or message with a digital signature, a method that uses a cryptographic algorithm to provide assurance that the user who digitally signed the document or message is indeed that user. To digitally sign a document the user typically must select a menu item to invoke the digital signature function, then enter a password. The following is a typical user interaction sequence:

1. Verifying that the document to be signed is the "current object" among all displayed objects within the window;
2. Visually sighting a menu bar that contains security-related options;
3. Using a pointing device, pulling down the menu bar to a "digital signature" option;
4. Visually sighting the dialogue box and the prompts and input field within the box;
5. Homing both hands at the keyboard;
6. Entering a string of alphanumeric characters;
7. Clicking on <OK> or pressing the <ENTER> key.

Compared to the previous interaction sequence, the current one includes two additional steps on the user's part. This leads to the prediction that digitally signing documents will result in a somewhat longer task completion time, somewhat elevated error rate, and moderately higher user resistance. The most common errors are likely to include the following:

1. Data entry errors;
2. Failure to remember one's password;
3. Errors resulting from the need to use more than one interaction device while performing the different steps of this user interaction sequence;
4. Failure to remember or locate the relevant menu options;
5. Failure to select the desired document as the current object.

Interestingly, several vendors' products include a feature called "Remember Password" for digital signing. This feature allows a user who has previously signed a document during a log-on session to subsequently sign documents without having to enter a password. Note that this eliminates step 6 in the seven-step interaction sequence shown above. The result is reducing the user's time and effort to remember the password, but with potentially serious consequences for security. Digital signatures are one of the most important functions in today's computing systems. These signatures are legally binding in at least four States in the U.S., and the U.S. Congress is currently considering legislation to make them legally binding throughout the entire country. The "Remember Password" function allows another, unauthorized user to easily sign another, legitimate user's document if the legitimate user briefly leaves the terminal unattended[1]. This provides a good example of how security and usability are sometimes diametrically opposed.

Biometric measurements

Biometric security methods make use of individual differences in personal characteristics. Currently, face recognition, fingerprint recognition, hand geometry analysis, iris scan, retinal scan, signature recognition, voice recognition, and face thermogram are the most widely available biometric methods (Jain, Hong, & Pankanti, 2000).

Biometric security systems can be broken down into two sub-categories, physiologically based (e.g. fingerprints) and those with a behavioural component (e.g. keystroke patterns, signature, or voice properties). Physiological systems have historically tended to be expensive, obtrusive, and low in user acceptance (Millar, 1994), although several current vendor implementations of these systems are exceptions to this trend. They can be very user unfriendly; temporary physical changes, such as cuts, burns, or blisters on the finger, may prevent the user from being authenticated. Behaviourally-based systems rate higher in acceptability (Sherman, 1992), but measurement issues need to be resolved if such systems are to be used with much success. For example, Deane, Henderson,

Mahar, and Saliba (1995) concluded that anxiety may have sufficient effects to alter performance responses, as well as some physiological ones, resulting in increased security challenges and interruption of workflow.

In addition to issues such as the situational stability of biometric measures, there are also questions concerning whether measures that have not received much consideration to date may prove useful. For example, Barrelle, Laverty, Henderson, Gough, Wagner, and Hiron (1996) investigated the possibility that parameters of an individual's use of indirect control devices, such as a mouse or pen, could be used as a means of user authentication. They concluded that error of classification based on use of pointing devices alone was too high for this to be a viable method of user authentication. However, they suggested that it might be useful if integrated into a multi-modal security system.

There are many additional steps involved in use of a biometric authentication method. Consider, for example, a commonly used biometric authentication method, one that reads user fingerprints. Typical user interaction steps include:

1. Visually sighting a prompt on the display terminal that prompts the user to place a finger on the fingerprint reader;
2. Visually sighting the fingerprint reader;
3. Moving a hand towards the fingerprint reader until it is in close proximity;
4. Rotating the hand until the palm side is down;
5. Extending a finger until it fits over the fingerprint reader;
6. Visually sighting the display terminal for confirmation that the fingerprint read was successful;
7. Reading a prompt that begins the "normal" username-password entry sequence;
8. Repeat the steps involved in a normal username-password-based log-on.

Because of the additional steps involved in this rendition of an authentication interaction task, one can safely predict that task completion time, errors, and user resistance will increase substantially (see Proctor, Lien, Salvendy, & Schultz, 2000, for a more extended discussion of these factors). Most common errors are likely to include the following:

1. Performing the steps in the wrong order (e.g. by first trying to enter a username-password combination);
2. Failure to place a finger in the proper position in the fingerprint reader;
3. Placing the "wrong" finger (i.e., a finger with a cut, which is likely to render the fingerprint read invalid) in the fingerprint reader;
4. Failure to keep finger still enough to have the fingerprint read;
5. A timeout occurs because the user does not perform a step within a prescribed time interval.

Smart cards and token devices.

Smart cards are credit-card sized plastic cards that carry information via an embedded computer chip. Smart cards can be fairly convenient because they can reduce the amount of items that a person must carry and/or remember. An additional security advantage of the smart card is that the PIN can be verified securely off-line.

There are two types of smart cards: contact and contactless. Contact cards require the user to insert the card into a reader, whereas contactless cards require the card to be only in close proximity to the reader. Each type of smart card can be programmed with four security levels: "read only," "added only," "update only," or "no access." Access to the information can also be designated so information can be accessed by everyone (e.g. all users' names), the cardholder only (personal identification number (PIN) must be entered), or a third party (e.g. the card issuer).

Secure tokens are also small cards used to provide authentication through a "log-on challenge" in which users must first connect to a service provider and use an authentication token such as a number displayed on a special device in order to gain access to the system. The token is always displayed for a limited time; users must enter the correct input (e.g., the same numbers and or characters in the token) before each

token expires. Although this authentication method may increase security substantially, it can be a nightmare for the user community. Untold hours of users' time can be wasted because of the complicated user interaction tasks, timing complications, and other problems associated with this authentication technique. Usability of the artifacts is a key factor in determining the relative success of these security methods.

As with other additional authentication methods, use of smart cards or tokens may improve authentication security, but not without several significant liabilities. The most salient one from management's perspective is additional financial costs associated with procuring, implementing, and maintaining the requisite hardware and software that is generally required. From a Human Factors perspective, most of these additional methods also entail significant usability hurdles.

Consider a real-life example that recently adversely affected the first author's ability to remotely access his corporation's computers. This corporation adopted a type of token-based authentication in which users had to first connect to a local Internet service provider and authenticate using a token and username-password sequence. They next had to connect through a different authentication program to the corporation's network with still another token. The token number is always displayed in black on a gray background on a type of small key-chain mechanism that has no built-in illumination. Worse yet, each token expires after a short time interval.

Although this authentication mechanism has increased the security of remote dial-in connections substantially, it initially proved to be a nightmare for the user community. The complicated user interaction tasks, timing complications, and other problems associated with this new authentication mechanism at first lowered user productivity considerably.

Integrity, Confidentiality, and Availability of Data and Systems

Any information stored on computers is subject to being deleted, altered, or stolen. Each of these outcomes has an associated cost that may be quite high. Consequently, considerable effort in information security is devoted to protection of data and systems. Three types of security objectives are data integrity, confidentiality, and availability.

Data integrity

Deletion or alteration of information on a computer can occur either accidentally or deliberately. Information is also vulnerable to alteration or deletion as it is transmitted over a network. The potential financial loss resulting from security breaches in which integrity is compromised may be quite high. Consider, for example, the importance of integrity in life insurance customer databases and the fallout that unauthorized deletions and modifications would cause not only among customers, but also among potential customers. Consequently, considerable effort in information security is devoted to data integrity assurance. Alteration of Web page content has become a particular concern for several reasons: 1) liabilities associated with unexpected, unauthorized changes advertised prices, quantities, and other terms and conditions, and 2) potential public relations damage. The media and competitors too often make organizations that experience unauthorized Web page alterations look hapless and incompetent. As we have recently seen, even White House Web pages have been maliciously altered.

The most commonly security control to guard again unauthorized changes in data integrity is setting permissions or access control lists to restrict access particularly access that results in ability to write to and/or delete files and directories. Most of today's operating systems incorporate file systems that permit such access restrictions. Some, such as Windows NT, offer both graphical- and command line-based methods of changing file permissions. Consider, for example, the steps involved in the graphical user interaction method when an owner of a file wants to allow a group read-level access to the file:

1. Visually inspect the objects on the desktop to find the NT Explorer;
2. Using a pointing device, scroll through groups of objects until sighting the desired object;

3. Using a pointing device, highlight the desired object;
4. Visually scan the menu bar to locate "File";
5. Using a pointing device, drag down the File Menu until reaching the "Properties" selection;
6. Using a pointing device, click on "Properties";
7. Visually scan the tabs at the top of the new screen that appears to locate "Security";
8. Using a pointing device, click on "Security";
9. Of the options that will next appear on the screen, click on "Permissions" using a pointing device;
10. Using a pointing device, click on "Add";
11. Using a pointing device, scroll through the list of groups and users;
12. Highlight the group or click on the "Show Users" box;
13. Scroll down to correct user name and highlight;
14. Select type of access;
15. Scroll through "Type of Access";
16. Highlight the type of access desired and release;
17. Click "OK" on three different screens.

Another method used to combat the data integrity problem involves computation and storage of cryptographic codes for all of the data on the server. Periodically, these cryptographic codes are recomputed and compared to the originally computed codes. A mismatch signals that something has changed and that an immediate investigation is needed. Effective operation of this method requires updating any time that data are added, deleted, or modified, and higher levels of security demand ever increasing complexity (e.g. parts of the cryptographic codes stored on separate systems), both for the hardware and for the security administrator. The extent to which the complexity and workload imposed on the administrator can be kept within reasonable bounds is an issue that needs to be examined.

Data confidentiality.

Data confidentiality involves ensuring that sensitive or proprietary data are protected and not disclosed to unauthorized persons. This is of particular concern for Internet commerce, where a customer's credit card number is sent to a merchant's Web server. The major method in protecting data confidentiality is data encryption using an algorithm that transforms data in some systematic fashion to render it unreadable unless a "key" is applied to unscramble it. However, use of encryption methods typically requires the user to perform additional procedures. At a minimum, many encryption methods require users to remember the "key." Many provide little or no means of key management or of recovery if a "key" is ever lost or becomes corrupted. The result is that users have to expend effort often considerable effort in creating and using their own methods of key management and recovery. Making data confidentiality procedures as convenient as possible, thus, seems to be a necessary step in effective use of the methods.

Data availability

One consequence of an attack by an intruder is that data may be made temporarily or permanently unavailable. Temporary unavailability can create disruptions in services and business operations, as in the case of the White House Web pages mentioned above. The consequences of denial of service are considerably more variable than those of data destruction. Some kinds of denial of service attacks lead to temporary inconvenience; others (e.g. in billing systems) can result in major financial loss. A special concern is that data may be irreplaceable; at the very least, restoration of the data and remediation of the problem will often require considerable time and resources. Among the means for protecting data availability is implementing adequate back-up and fault tolerance procedures, which again places a large responsibility on the system administrators.

System Integrity

Another consequence of an attack by an intruder is that system parameters or executable programs may be altered without authorization. This kind of attack can be potentially very costly and disruptive if the unauthorized modification results in downtime, poor performance, or incorrect output of programs that process data. The problem of system integrity can be addressed by a combination of setting file and directory permissions properly, limiting privileges on any system, and running programs that perform integrity checking.

Detection of Intrusions

A properly executed Internet security program requires proper program management and administration. For example, vendor patches for security-related flaws in operating systems and application programs need to be installed promptly. If the system administrator neglects to install these patches or does so improperly, the risk of a security breach increases. The Human Factors literature abounds with knowledge concerning how systems can be designed to increase the probability that action is taken when needed. A systematic analysis of the administrator's tasks should provide us with insight into the factors that might encourage prompt patch installation.

A secure system must provide protection against intrusions and a method for detecting intrusions when the protection fails. Setting up the system for protection against intrusions typically involves setting up system and network defences to repel attacks (i.e., firewall software) and permissions or access control lists to permit or restrict information access to specific groups of users. This will help to ensure that unauthorized users will not be able to access files to read, write, and/or delete them. Most of the previous emphasis on maintaining security against intrusions has focused on detection of unauthorized access to systems and the files stored therein or corruption of files. Although this is important for maintaining security, recent interest involves prevention of non-invasive attacks such as network traffic flooding attacks.

These attacks are in some ways more hazardous than unauthorized system access, file corruption, and so forth because they can cause systems to lock-up and fail, which can be catastrophic for business. Down time for systems interrupts operational cycles, resulting in outcomes such as failing to bill customers on time, loss of interest, and others.

Without an appropriate security monitoring and audit program, remote intrusions and unauthorized accesses of databases likely will go unnoticed. Two methods for performing security monitoring and audit are system audit logging and use of intrusion detection systems.

System audit logging

System audit logs provide information about usage characteristics on a computer system. Regular inspection of the audit logs by system administrators can lead to detection of unauthorized usage patterns. The major problem with such logs is the massive amount of information they provide. System administrators must inspect system logs that are usually poorly formatted to determine whether or not unauthorized activity has occurred. Because typical logs display line-after-line of monochrome text that is also uniform in size and font and arranged in columns, it is difficult for system administrators to detect break-ins.

A key component in the success of system audit logging for intrusion detection is the ability of administrators to recognize patterns indicative of intrusion activity. Much is known about human pattern recognition, how to display information in such a way as to make the patterns easier to detect, and training humans to recognize specific patterns "automatically." (e.g. Aspillaga, 1996). Yet, this knowledge has for all practical purposes not yet been applied in the area of intrusion detection.

Intrusion detection systems

Intrusion detection systems recognize suspicious access and usage patterns, signaling security administrators with an alarm or less obtrusive indication that an unauthorized pattern has occurred. One problem with such systems is that the security administrators often do not have sufficient knowledge regarding setting up and using the systems optimally. Another problem is that the system or network administrator (or, possibly, an intrusion detection specialist) must ultimately decide whether an intrusion has taken place and also whether to report the intrusion.

As with system audit logging, the administrator's task involves pattern recognition —- a decision whether the suspicious pattern really is indicative of an intruder must be made. The task also falls within the category of "vigilance tasks," which have been widely studied in Human Factors (e.g. the administrators must monitor the system continuously for infrequently occurring "signals" of intrusion over long "vigils").

Factors such as a high false-alarm rate from the intrusion detection system can decrease the likelihood that a real intrusion will be detected by the administrators. Even if an apparent intrusion is detected, the administrators must decide whether to report the intrusion. Consequently, only a portion of the already small percentages of intrusions that are detected will be reported. This is one aspect of information security in which a systematic Human Factors analysis of the process clearly should be of significant benefit.

Implementing an information security policy. The initial step in maintaining security is creating an appropriate information security policy and getting senior management support for it (Bernstein, Bhimani, Schultz, and Siegel, 1996). This policy must then be implemented, with a major part of the implementation involving setting up the system appropriately. This is done mainly through setting of permissions or access control lists. For example, most of today's operating systems incorporate file systems that permit access to different levels of information for specific groups of users (e.g. system administrators versus users).

With a small number of computers, administrators should have little problem implementing procedures for setting file permissions. However, for companies that use hundreds or thousands of computers, setting file permissions can be a significant problem. In these cases, the administrators themselves usually do not set many of the permissions; they depend on their staff (in many cases, users) to do so. Therefore, administrators must make sure that their descriptions concerning setting file permissions are accurate and simple to follow. The extent to which the complexity and workload imposed on the system administrator can be kept within reasonable bounds is thus still another issue that needs to be examined.

Setting the system to identify intrusions is also important in maintaining security. There are two methods by which intrusions can be identified. The first is to have a human administrator monitor the system for deviations from normality; the second method is to have computer software check for reoccurring attacks and unfamiliar patterns of activity.

In addition to setting up measures against attempted intrusions, administrators must ensure that audit logs are not corrupted or altered. Security can be maintained by having human resources allocated to monitoring the system or by implementing an automated checking system with minimum administrative support. If human resources are used to monitor the system, then the system's activity must be recorded and presented in a readable and manageable manner. On the other hand, if an automated checking system is utilized, it must be programmed in an optimal and efficient manner. This may involve computation and storage of cryptographic codes for all of the data on the server. Periodically, these cryptographic codes are recomputed and compared to the originally computed codes. A mismatch signals that something has changed and that an immediate investigation is needed. Effective operation of this method requires updating any time that data are added, deleted, or modified, and higher levels of security demand ever increasing complexity (e.g. multiple copies of the cryptographic codes stored on separate systems), both for the hardware and the security administrator.

A High-Level Framework

Many different security control methods, several of which we have reviewed in this paper, have been developed to fill the increasing need for security of information in computer systems. These methods involve interactions between humans and computer hardware and software, yet they were largely developed with little regard for usability. Because the degree of security provided by the various security control methods typically depends on the actions of system administrators and end-users, security hinges on the usability of the methods. Consequently, there is a critical need for researchers in Human Factors and HCI to focus their efforts on systematic investigation of usability issues associated with information security-related tasks. But what would be an appropriate initial step?

A logical initial step in the investigation of usability is to develop a taxonomy of the domain of interest, in this case, information security methods, and to analyze the tasks that a user must perform for each category

within the taxonomy. Table 1 in this paper lists the major threats to security and the types of security control methods that have been developed to counter those threats. For each type of security control method, we have described the tasks that end-users and system administrators must perform and the information-processing demands that these tasks impose on them. We have also provided several illustrations of the sequences of actions in which the user must engage to perform a particular security-related task and the errors that might occur during the task performance. Given this foundation, a major step forward would be to relate information security tasks and methods, threats, and Human Factors issues together in a high-level framework to guide future research on this topic. A taxonomy can be used to provide the necessary level of conceptualization.

Table 2 includes a detailed taxonomy of the information security tasks, the risks associated with each, and the associated usability issues. This taxonomy highlights many of the usability concerns that need to be addressed in the field of information security. We feel it is important to make a distinction between usability issues pertaining to the end users and those pertaining to the system administrators. Many of these issues involve fundamental aspects of human performance, such as the high memory demand imposed by passwords that are difficult to crack. Others, such as how to display the large amount of data in system audit logs so that the information can readily be processed by the system administrator, are relatively unique to information security. In addition to specifying the usability issues for each control type, the taxonomy includes a broad range of general usability issues associated with displays and controls that apply to a range of information security methods.

Conclusion

Although this paper documents the need for sound human factors design and analysis in information security, it by no means resolves the many issues that need to be addressed. This paper has described the major types of security controls that currently exist, why these controls are necessary, and some of the usability issues associated with each. Although information security involves automated computing systems, protecting these systems necessarily involves human intervention both in judgment of appropriate courses of action and performance of interaction sequences that implement these courses of action. We have argued that failure to take into account the "human-in-the-loop" factor will result in security control measures that users will resist and that will result in task performance that is unnecessarily inefficient and error prone.

Where should we go from here? We recommend two important steps. First, we challenge the information security and human factors communities to build on the task taxonomy shown in Table 2. We are confident that understanding the major types of information security tasks and human usability factors is critical to future research and analysis. Second, we advocate developing better metrics. In the field of information security, developing suitable metrics is in and of itself a formidable challenge. Assigning numbers that accurately reflect the degree of security a system provides is no small challenge. Most existing metrics incorporate the system components such as system specification and configuration (Department of Defense, 1985). However, metrics that capture the "human-in-the-loop" component of the system have not yet been developed. Because the security that a particular method provides is heavily dependent on end-users' and administrators' interactions with computers, metrics regarding usability of security methods are necessary. Thus, a longer-term goal should be to develop usability metrics for alternative security methods and to incorporate these usability metrics into the metrics for system security. Without an effective way of incorporating usability into system-security metrics, system security metrics will not accurately reflect a system's actual security level.

References

Al-Ghatani, S. S., & King, M. (1999). Attitudes, satisfaction and usage: Factors contributing to each in the acceptance of information technology. *Behaviour & Information Technology, 18*, 277-297.

Aspillaga, M. (1996). Perceptual foundations in the design of visual displays. *Computers in Human Behaviour, 12*, 587-600.

Barrelle, K., Laverty, W., Henderson, R., Gough, J., Wagner, M., & Hiron, M. (1996). User verification through pointing characteristics: An exploration examination. *International Journal of Human-Computer Studies, 45*, 47-57.

Bernstein, T., Bhimani, A. B., Schultz, E. E., & Siegel, C. A. (1996). *Internet Security for Business*. New York: Wiley.

De Alvare, A. M., & Schultz, E. E. (1988) A framework for password selection. *USENIX Security Workshop Proceedings, 1*, 8-9.

Deane, F., Henderson, R., Mahar, D., & Saliba, A. (1995). Theoretical examination of the effects of anxiety and electronic performance monitoring on behavioural biometric security systems. *Interacting with Computers, 7*, 395-411.

Department of Defense (1985). *Trusted computer security evaluation criteria*. Publication number STD-001-85.

Gollmann, D. (1999) *Computer Security*. New York: Wiley.

Helander, M. G., Landauer, T. K., & Prabhu, P. V. (Eds.). (1997). *Handbook of human-computer interaction*. Amsterdam: Elsevier.

Jain, A., Hong, L., & Pankanti,, S. (2000). Biometric identification. *Communications of the ACM, 43(2)*, 91-98.

Klein, D. (1990). Foiling the cracker: A survey of, and improvements to, password security. *Proceedings, UNIX security workshop II.*

Markus, M. L. (1983). Power, politics, and MIS implementation. **Communications of the ACM, 26,** 430-444.

Martinko, M. J., Henry, J. W., & Zmud, R. W. (1996). An attributional explanation of individual resistance to the introduction of information technologies in the workplace. *Behaviour & Information Technology, 15*, 313-330.

Millar, B. (1994). Vital signs of identity. *IEEE Spectrum*, February, 22-28.

Parker, D. B. (1991) Restating the foundation of information security. *Proceedings of 14th National Computer Security Conference*, 480 – 493.

Proctor, R. W., Lien, M.-C., Salvendy, G., & Schultz, E. E. (2000, April). A task analysis of usability in third-party authentication. *Information Security Bulletin*, 49-56.

Salvendy, G. (Ed.) (1997). *Handbook of human factors and ergonomics* (2nd ed.). New York: Wiley.

Sherman, R. L. (1992). Biometrics futures. *Computers and Security, 11*, 128-133.

Stallings, W. (1995). *Network and internet security*. Englewood Cliffs, NJ: Prentice-Hall.

Turnage, J. J. (1990). The challenge of new workplace technology for psychology. *American Psychologist*, 45, 171-178.

Van Wyk, K. R. (1994). Threats to DoD Computer Systems. Paper presented at 23rd Information Integrity Institute Forum.

Warm, J. S.; Dember, W. N.; Hancock, P. A. (1996). Vigilance and workload in automated systems. In R. Parasuraman, M. Mouloua,

Mustapha (Eds.), *Automation and human performance: Theory and applications* (pp. 183-200). Mahwah, NJ: Erlbaum.

Table 1 – Major Types of Security-Related Controls, the Threats they Counter, and the Associated Usability Issues

CONTROL TYPE	TYPE OF THREAT COUNTERED	USABILITY ISSUES
Identification and Authentication	Masquerading as another user; repudiation	Willingness of users to adopt; ease of using method/device
Data Integrity	Unauthorized deletion and/or changes	Install and maintain appropriate software; control access rights and privileges
Data Confidentiality	Unauthorized disclosure and/or possession	Control access rights and privileges
Data Availability	Unauthorized deletion of data and/or the databases/ programs used to store and retrieve them; denial of service attacks	Protection provided by system manage backup media; ease of implementing
System Integrity	Unauthorized deletion and/or changes to system data/ configuration files; theft; denial of service attacks	Inspection by administrators; detection by software
Intrusion Detection	Unauthorized access to systems; denial of service attacks	Inspection by administrators; detection by software; ease of implementation

Table 2 – Taxonomy of Information Security Tasks and Related Human Usability Factors
The first column lists different information security methods and tasks. The potential risk or error associated with using each method is outlined in the second column. The third and fourth columns include descriptions of the demands imposed on the users and system administrators by each security method.

Nature of the Security Task	Risks or Errors	Users	Administrators
Identification and Authentication			
• Identify the user, such as password entry.	• Unauthorized access.	• Require user to maintain and act on knowledge that is sometimes detailed.	• Password administration • Need to create and reset passwords
• Determine user's location, such as caller ID, user callback, Global Position System (GPS).	• Weak authentication and easy to crack passwords.	• High memory demand. • May impose unnatural syntactic and other constraints (e.g. minimum password length)	
• Biometrics approach • Physiological based, such as fingerprints and retinal readers.	• Often not user friendly. • Cause user's anxiety and alter performance responses.	• These systems have to consider the anthropometric constraints. • User's anxiety increases when use those systems.	• Install and maintain the necessary software and hardware components. • More cost on setting the devices.
• Behavioural based, such as keystroke patterns, signature, or voice properties.	• For the fingerprint approach, if any cuts, burns, or blisters were created on the finger after user enrolled, the user may not be able to authenticate with it. • For the retinal readers, the design is not comfortable for certain populations, such as handicap people, shorter and taller persons.	• Although the biometrics approach provides the highest degree of security, it also makes it harder for the enrolled users to access their system.	• Increase security challenges and interrupt workflow when user's anxiety affects the system. • Avoid user's anxiety affecting the system. • The false acceptance rate (FAR) on failing to reject an imposter attempting to gain access to the system is lower. • Make sure the device (such as fingerprint scanner) captures good quality images of user's fingers to ensure accurate authentication and enrollments.
• Smart cards and token devices • Digital Signatures • Installing patches	• The problems associated with the secure ID token device like keychain include misplacing the token.	• These devices have to be designed to be ergonomic and easy to use (usability).	• Be able to provide flexible verification when the system fails to identify the users. • Install and maintain the necessary software and hardware components • Require periodic cleaning and maintenance of those devices.
Integrity, Confidentiality, and Availability of Data			
• Ensuring that data have not been modified or deleted. • Backups • Integrity verification – modify message digest technique • Encryption, cyclic redundancy checks (CRCs) • Setting files and directories • Time Stamping • Event logging • Anti-viral software	• Difficulty in accessing the information because of security controls. • Fail to install or maintain programs properly. • Data are subject to being deleted, altered, or stolen. • Failure to suitably manage encryption keys. • Format of audit logs and intrusion detection data difficult to read and comprehend.	• Be able to understand and recognize the commands and options. • Be able to use the commands in appropriate ways. • Attention resource demand. • Be able to remember keys used in digital signatures. • Have to inspect their files to ensure they have not been changed.	• Be able to guide the user step by step, suggesting viable options and hiding inappropriate actions. • Manage backups media and store backups properly. • Make sure the system has been installed correctly and be able to provide sufficient access permissions for users to access the data. • Maintain multiple copies of files on separate systems. • Maintaining privilege control.
Intrusion Detection			
• System audit logging • Intrusion detection systems • Tripwire	• Remote intrusions and unauthorized accesses of data.	• User's privacy can be compromised.	• Perform regular inspection of the audit logs. • Organize the massive amount of

Nature of the Security Task	Risks or Errors	Users	Administrators
• Monitoring	• Unable to detect the intrusion (e.g. due to inability to recognize patterns). • High false alarm rate can decrease the likelihood that a real intrusion will be detected • Disabling or modifying audit logging or intrusion detection software. • Disruption of system performance.		information logs provide. • Recognize complex patterns quickly and easily. • Monitor the system continuously for infrequently occurring "signals" of intrusion over long vigils. • Be able to immediately investigate after detecting intrusions. • Investigate anomalies. • Install and maintain appropriate software.
Responding to Intrusions		• N/A	
• Execution of incident response procedures • Use of automated response software. • Failure to correlate relevant data. • Correlated multiple sources of data. • Implement additional defensive measures.	• Failure to execute procedures properly • Automated response software fails to run or becomes corrupted.		• Diagnose status of systems and networks. • Determine source of incident(s). • Install and maintain appropriate software. • Take defensive actions, such as disconnecting from the network.
Assurance of Operational Continuity		• Report disruption of service.	
• Monitoring; need for continuous verification • Availability • Fault tolerance / redundant • Redundant firewalls	• Failure to monitor or to recognize disruption.		• Monitor status of systems and net works. • Determine source of disruption(s). • Install and maintain appropriate software and hardware. • Make back-ups. • Enable fault tolerance measures.
General Usability			
• Information display • Legible and readable (such as font size, luminance) • Clear and understandable • Concern with the visual angle and viewing area • Format of data displays facilitates rapid scanning and pattern recognition.	• Misrepresent the information or instruction and causes the errors. • Misleading	• Be able to understand the instructions and take the appropriate steps. • Acquire knowledge about the system by reading the instructions, performing training tutorials, imitating described routines, and practicing with the system.	• Provide clear and understandable procedures or instructions that match users' mental representations. • Use appropriate command syntax. • Confirmation of actions that could result in catastrophic consequences. • Provide software that requires relatively few and intuitive steps to install and maintain.
• Entry control • Keyboard devices and design • Touch screen, touch pads, and the mouse	• Impair performance and cause injuries	• Have to familiar with the files or functions that categorized by the administrator or system designer (minimize the searching time for the target or a particular function). • Have sufficient knowledge on the type of information displayed (such as a red circle indicates the key is invalid and a green circle indicates it is valid). • Be able to use the shortcuts that system provides to accomplish the task quicker.	• Provide the windows that include menu bar icons for the novices and command lines for the experienced users. Both methods have to come along with a clear description of their functions. • Use hierarchical display design and top-down knowledge processing to display the instructions and functions. • Provide keyboard shortcuts for most menu operations. • Design features on the system that provides immediate access to the functions regardless of which applications that users run. • Provide appropriate help menu, such as a general overview and instructions for all of the procedures users are likely to perform.