Even Experts Deserve Usable Security: Design guidelines for security management systems

Sonia Chiasson
Human Oriented Technology Lab & School of Computer Science
Carleton University
chiasson@scs.carleton.ca

Robert Biddle
Human Oriented Technology Lab
Carleton University
robert_biddle@carleton.ca

Anil Somayaji
School of Computer Science
Carleton University
soma@scs.carleton.ca

ABSTRACT
Contrary to end-users, security is a primary task for those charged with the security of system or network. Despite the importance of the task, little is known about how to effectively design interfaces for security management systems. Usability problems in these systems can lead to security vulnerabilities because administrators may miss an attack altogether or misdiagnose it. We examined four different design approaches in order to devise a preliminary set of design guidelines for security management systems.

Categories and Subject Descriptors
H.5.2 [**Interfaces and Representation**]: User Interfaces – *Graphical user interfaces*; K.6.5 [**Computing Milieux**]: Security and Protection.

General Terms
Design, Security, Human Factors.

Keywords
Usable security, security management, user interface, design principles.

1. INTRODUCTION
The field of usable security has primarily focused on designing user interfaces for end-users. These users have little knowledge about computer security and their focus is not on completing security tasks. In contrast, security professionals or system administrators have much more domain and system-specific knowledge. Maintaining the security of the systems within their care is a main priority. Since these are typically advanced computer users, little effort is spent on the design of usable interfaces for this group. Interactions usually consist of writing rules in a given scripting or programming language, sifting through very large amounts of textual output, or interpreting information that requires in-depth domain knowledge of computer security, systems, and networks. Designers concern themselves with measures such as false positive rates, acknowledging that the user can only be expected to sift through and respond to a certain amount of data, but this usually does not translate into further work on how to best design the user interface.

The area of computer security poses a unique set of challenges for interface design because attackers will be trying to avoid detection. There is no predefined notion of exactly what information the interface should be conveying or how to present it in the most meaningful way since each new attack will look somewhat different and require a different approach to detect it. The challenge becomes how to provide enough detail so that attacks can be detected without overwhelming administrators with information.

Designing usable interfaces is a matter of security since humans are more likely to make mistakes in cases where they must process and interpret multiple alarms, indicators, and other sources of data [11]. The current interfaces rely on the administrator to know what to look for, how to find it, and how to assimilate all of the clues in order to detect and analyze a potential attack. When interfaces are too cumbersome, present too little or misleading information, or overwhelm the administrator with too much information, then security will suffer.

This paper provides a first attempt at defining a set of user interface design guidelines and approaches for security management systems. While these will need to be further evaluated through user testing before they can be accepted as valid with any certainty, this initial exploration represents an important first step forming guidelines for usable security management systems. It may also prove useful in suggesting additional guidelines for general usable security interfaces.

Section 2 provides an overview of existing interfaces for security management systems. Relevant design approaches for user interfaces are described in Section 3, while the proposed design guidelines for security management systems are presented in Section 4. Section 5 offers concluding remarks.

2. BACKGROUND
In practice, interfaces specifically for security monitoring are often not even available. Instead, administrators must re-purpose existing network or system monitoring tools in an attempt to detect and diagnose security problems. These monitoring systems are not especially usable even for their intended purpose, much less for security analysis. For example, IBM's Tivoli [14] network monitoring system is designed for monitoring performance and usage of enterprise networks. It has a graphical interface and offers some visualizations, but still appears awkward for users. Performing security monitoring through the system may be even more challenging.

To appear in the Workshop on Usable IT Security Management (USM'07), held in conjunction with the Symposium on Usable Privacy and Security (SOUPS 2007) in July 2007.

Version: April 22, 2007.

Attempts at creating more advanced interfaces have come primarily from the field of information visualization. Security monitoring interfaces have been devised to visualize large data logs [1][5][15]. These visualizations typically allow the user to navigate within the data set, either by filtering, rotating, or otherwise manipulating the data in order to detect abnormal behaviour. These interfaces are intended for security administrators needing to process large amounts of data such as network traffic, system logs, and intrusion detection alarms. While offering an improvement over going through text logs, they still require extensive domain knowledge and skills in interpreting these complex visualizations.

A primary goal of security monitoring is to detect attacks against the system or network being monitored [7]. However, existing user interfaces for intrusion detection systems (IDSs) have also been fairly primitive. Most of the focus has been the technical aspects of intrusion detection while the user interface has been a secondary concern. In fact, most research papers discussing proposed IDSs barely acknowledge the existence of a user interface at all.

3. DESIGN APPROACHES

Besides general human-computer interaction (HCI) and intrusion detection literature, there are several specific areas of research that contain useful information in developing user interface design guidelines for security management systems. This section introduces these areas and summarizes the most relevant design principles.

3.1 Usable Security

The field of usable security recognizes that to be secure, a system must be usable. Even the most technically secure system will fail in practice if the intended users cannot or will not use it properly. The usable security community has done work in developing effective end-user interfaces for anti-phishing software [8], password managers [4], and other areas of security [6]. However, being a relatively young field, there are no well-developed theoretical frameworks or models to provide an overall definition of what makes a security interface usable.

As a starting point, Whitten and Tygar [19] proposed a set of usability guidelines for security interfaces. We extended these [4] and our combined guidelines suggest that users should:

1. be reliably made aware of the security tasks they must perform;
2. be able to figure out how to successfully perform those tasks;
3. not make dangerous errors;
4. be sufficiently comfortable with the interface to continue using it;
5. be able to tell when their task has been completed; and
6. have sufficient feedback to accurately determine the current state of the system.

Most of these guidelines hint at an overarching theme in usable security: the need for effective mental models [16]. A mental model is an understanding of how the system works, typically based on previous experience, the system's user interface, and the user's previous interaction with the system. Users need a workable mental model of the system in order to perform their tasks successfully. The mental model may not accurately reflect all of the technical details of the system but should provide a means of predicting observable system behaviour and the consequences of user actions.

As with other usable security interfaces, one of the most critical design goals of security management interfaces must be fostering an effective mental model of the system for administrators. However in order for security professionals to successfully accomplish their tasks, the mental model must provide a clear picture of the underlying system being monitored because these administrators will often need to diagnose and respond to unusual and unexpected security situations. This will prove challenging in an information-dense system but must be addressed.

3.2 Ecological Interface Design

Vicente [17][18] proposed the Ecological Interface Design (EID) theoretical framework for designing complex socio-technical systems. EID has primarily been applied to large-scale systems of critical importance such as controlling power plants, aviation systems, and more recently for computer network management [2]. The framework is based on well-established psychological research showing how humans process information and problem-solve. Contrary to other user interface approaches which try to shield users from the intricate details of the system, EID advocates allowing the revelation of as much detail as possible so that when unexpected events occur, users can gain a clear understanding of the state of the system and troubleshoot effectively. The belief is that users' mental model should accurately reflect the actual system being controlled so that users can most effectively perform their tasks. The expected users in these cases are already knowledgeable in the domain in which they are working.

The principle idea behind EID is to design the interface according to a 5-level abstraction hierarchy, with the topmost level giving a general overview of the system state and providing progressively more detail with each level. The lowest level provides a representation of the physical layout, components, and sensors controlled by the system. Users operate at different levels in the hierarchy, as required for their tasks, but when unexpected events occur, they can quickly move up to gain an understanding of the overall seriousness of the problem or drill down in order to troubleshoot and diagnose the issue.

There are parallels between the concepts of EID and object-oriented design. Each higher level in the hierarchy offers an encapsulated view of the system, but in the case of EID, revelation of the encapsulation is allowed and encouraged. Users are not restricted to a single encapsulated view of the system but should be able to easily access and modify the subordinate objects then move back up the hierarchy to see the consequences of their actions.

An EID interface usually avoids trying to make diagnoses for users (again contrary to conventional HCI principles) because the basic premise is that designers will be unable to predict every possible event and as such the system may offer an incorrect or misleading interpretation. The systems are intended to foster accurate mental models and support users in making a diagnosis by providing as much information as possible without offering an "opinion" about the cause of unexpected events.

EID offers an interesting model for security management systems as the parallels between security and the other application

domains are evident. It may offer a means of making available all of the details necessary for diagnosis without overwhelming administrators.

3.3 Social Navigation

Social navigation [9] is based on the human tendency to use cues from other people in order to make decisions about our own behaviour. People use social navigation on a daily basis. In the online world, this is translated into actions such as using reviews and recommender systems on web pages to decide whether to buy a certain product or checking the status of friends on an instant messenger before disturbing them. Advocates of social navigation also suggest using more subtle cues like having some indication of the collective activities of previous users (similar to how a recipe book falls open to a family favourite because that page has been read so often) to give an indication of what people actually do rather than simply what designers intended [13].

DiGioia and Dourish [10] discuss using social navigation as a tool in usable security to show the history of a user's actions, patterns of conventional use, and activities of others within a system. They demonstrate their approach in the context of making file sharing decisions in a peer-to-peer network.

Administrators currently use social navigation when they turn to online forums and mailing lists to discuss the latest security vulnerabilities and how to address them, their success at applying new patches, and to get advice from others who have dealt with similar circumstances. Including social navigation directly into the security management interfaces could facilitate this process and even improve it because the system could allow for direct comparison of two events to see if they really are instances of the same phenomena rather than relying on administrators to describe and compare what is happening. Specifically, the interfaces could help with *filtering* [10] by helping users make informed decisions based on aggregate behaviour. The *quality* [10] of information can also be assessed by providing information about who performed which actions, who is recommending each course of action, and relying more heavily on the behaviour and recommendations of trusted experts.

3.4 Persuasive Technology

Persuasive technology [12] is a new area of human-computer interaction that looks at how computers can motivate and influence users to behave in a desired way. Motivating users to behave securely is a commonly cited goal in computer security and as such, persuasive technology may provide valuable insight into how to design better security interfaces [3]. The principles most likely to help achieve the goals of usable security:

1. Principle of Reduction: Making the desired path one of least resistance.
2. Principle of Reciprocity: Harnessing the human tendency to return favours.
3. Principle of Expertise: Incorporating signs of expertise such as experience, knowledge, and competence to gain credibility with the users.
4. Principle of Conditioning: Using positive reinforcement to encourage the desired behaviour.

A common mistake made by designers of security interfaces is to make the interface "invisible", with the belief that an invisible interface is the least obtrusive and therefore most usable. While this does include security in the path of least resistance, it can cause usability problems because it typically translates into no feedback to users. A password manager is one example, since intuitively having only one password should be easier than having to remember multiple passwords, but usability tests [4] show that current interfaces have major usability issues which affect the security of the systems.

Products such as Norton Security and McAfee Security successfully apply persuasive principles. For example, users are reminded of how many viruses were stopped, in hope that users will feel a need to return the favour and keep their system up-to-date. The interfaces also promote credibility by displaying messages when updating virus definitions and alerting users whenever a virus is detected and removed. More than simply keeping users informed, these are meant to instill confidence that the software is protecting the computer.

Such are alerts are more likely to annoy rather than reassure expert users; however persuasive technology could offer strategies for helping administrators understand the severity of threats and ensuring that critical issues are promptly addressed.

4. PROPOSED DESIGN PRINCIPLES

Considering these four approaches to interface design, we propose the following initial set of design guidelines for security management interfaces.

1. Administrators should reliably and promptly be made aware of the security tasks they must perform;
2. Administrators should be able to figure out how to successfully perform those tasks;
3. Administrators should be able to tell when their task has been completed;
4. Administrators should have sufficient feedback to accurately determine the current state of the system and the consequences of their actions;
5. Administrators should be able to revert to a previous system state if a security decision has unintended consequences;
6. Administrators should be able to form an accurate and meaningful mental model of the system they are protecting;
7. Administrators should be able to easily examine the system from different levels of encapsulation in order to gain an overall perspective and be able to effectively diagnose specific problems;
8. The interface should facilitate interpretation and diagnosis of potential security threats
9. Administrators should be able to easily seek advice and take advantage of community knowledge to make security decisions;
10. The interface should encourage administrators to address critical issues in a timely fashion;

These design principles seek to address a few important characteristics of this particular design space. First, they acknowledge that the user will need to be making important decisions and needs to be supported in this process. Most of the

interactions will occur due to unexpected events that the system cannot deal with on its own, and as such it should try to provide clear, relevant, and sufficient information so that the user can accurately diagnose and address the problem.

Furthermore, it is to be expected that users will occasionally make mistakes when dealing with these novel situations so the system must allow users to easily revert to a previous state. Such mistakes differ from the "dangerous errors" addressed in usable security because these mistakes may not be possible to predict (whereas dangerous errors such as entering a password in a phishing site are always considered bad). For example, botched upgrades through security patches can lead to unstable systems that need to be rolled back. Occasional mistakes are unavoidable and thus the systems must be flexible enough so that recovery is possible.

When faced with a new security threat, it is likely that others are also being similarly attacked. The interface should support and facilitate interaction within the security community not only to more quickly analyze a new threat and determine appropriate counter-measures, but also to facilitate propagation of such security measures. Social navigation could also be used to provide trusted feedback about what steps others have taken in similar situations, and could be further customized by defining a specific group of trusted sources from which to gather information. Integrating the communication and social navigation into the system could be faster, have less noise, and be harder to spoof than current ad hoc methods.

Security systems still generate a sufficiently large number of false alarms to potentially lure administrators into ignoring alarms or deeming them as non-urgent, or otherwise lead to situations where it is impossible to address all alarms. This may result in unnecessarily vulnerable systems. The interface should attempt to recognize such situations and encourage the administrators to take corrective action. The interface should alert administrators if the majority of other security professionals have taken some preventative measure that has yet to be addressed in the current system, especially if related to a severe threat given the specific system configuration.

It should be noted however that persuasive technology should be limited to helping administrators promptly address important security issues rather than trying to influence and guide them during diagnosis since the system may inadvertently lead the administrator to misdiagnose significant problems.

5. CONCLUSION

End-users are the main concern for the field of usable security, but interfaces for security professionals are also important because the consequences of usability problems can potentially leave entire networks vulnerable to attack. Knowledge of how to design for end-users can help in designing interfaces for security experts; however unique challenges remain as the two groups are very different in terms of domain knowledge, responsibility, amount of information they must process, and consequences of their actions.

We have examined several design approaches in order to devise an aggregate model for security management systems. A set of ten design guidelines are proposed based on this aggregate approach. While these will need to be further examined and evaluated, they present a first attempt at defining guidelines for the design of security management systems.

REFERENCES

[1] Abdullah, K., et al. *IDS RainStorm: Visualizing IDS Alarms.* ACM Workshop on Visualization for Computer Security (VizSec), October 2005.

[2] Burns, C.M., Kuo, J., and Ng, S. *Ecological Interface Design: a new approach for visualizing network management*. Computer Networks, v.43, Elsevier, 2003.

[3] Chiasson, S. and Biddle, R. *Persuading Users to Behave Securely.* 2nd Conference on Persuasive Technology, April 2007 (poster).

[4] Chiasson, S., van Oorschot, P.C., Biddle, R. *A Usability Study and Critique of Two Password Managers.* 15th USENIX Security Symposium, August 2006.

[5] Conti, G., Ahamad, M., Stasko, J. *Attacking Information Visualization System Usability: Overloading and Deceiving the Human*. Symposium on Usable Privacy and Security (SOUPS), July 2005.

[6] Cranor, L.F. and Garfinkel, S. (eds). *Security and Usability: Designing Secure Systems that People Can Use.* O'Reilley Media Inc, Sebastopol, CA, 2005.

[7] Denning, D.E. *An intrusion detection model*. IEEE Trans. on Software Engineering, 1987.

[8] Dhamija, R., Tygar, J.D., and Hearst, M. *Why Phishing Works*. SIGCHI conference on Human Factors in Computing Systems (CHI), April 2006.

[9] Dieberger, A., et al. *Social Navigation: Techniques for Building More Usable Systems*. ACM Interactions, v.7(6), November-December, 2000.

[10] DiGioia, P. and Dourish, P. *Social Navigation as a Model for Usable Security*. Symposium on Usable Privacy and Security (SOUPS), July 2005.

[11] Endsley, M.R. and Garland, D.J. (editors) *Situation Awareness Analysis and Measurement*. Lawrence Erlbaum Associates, 2000.

[12] Fogg, B.J. Persuasive *Technologies: Using Computers to Change What We Think and Do*. Morgan Kaufmann Publishers, San Francisco, CA, 2003.

[13] Hill, W.C, et al. *Edit wear and read wear*. SIGCHI conference on Human Factors in Computing Systems (CHI), 1992.

[14] IBM. Tivoli software. www-306.ibm.com/software/tivoli/ Last accessed April 2007.

[15] Komlodi, A., et al. *A user-centered look at glyph-based security visualization.* ACM Workshop on Visualization for Computer Security (VizSec), October 2005.

[16] Norman, D.A. *The Design of Everyday Things*. Doubleday: New York, NY, 1988.

[17] Vicente, K.J. and Rasmussen, J. *Ecological Interface Design: Theoretical Foundations*. IEEE Trans. on Systems, Man, and Cybernetics, volume 22(4), July/August 1992.

[18] Vicente, K.J. *Ecological Interface Design: Progress and Challenges*. Human Factors, volume 44(1), Spring 2002.

[19] Whitten, A. and Tygar, J.D. *Why Johnny Can't Encrypt: A usability evaluation of PGP 5.0*. 8th USENIX Security Symposium, August 1999.