International Journal of Human Computer Studies Special Issue on HCI Research in Privacy and Security, 63(1-2), 5-24. 2005.

In the Eye of the Beholder: A Visualization-based Approach to Information System Security

Rogério de Paula, Xianghua Ding, Paul Dourish, Kari Nies, Ben Pillet,
David F. Redmiles, Jie Ren, Jennifer A. Rode and Roberto Silva Filho

Institute for Software Research
University of California, Irvine
Irvine, CA 92697-3425

{depaula, dingx, jpd, kari, bpillet, redmiles, jie, jen, rsilvafi}@ics.uci.edu

Abstract

Computer system security is traditionally regarded as a primarily technological concern; the fundamental questions to which security researchers address themselves are those of the mathematical guarantees that can be made for the performance of various communication and computational challenges. However, in our research, we focus on a different question. For us, the fundamental security question is one that end-users routinely encounter and resolve for themselves many times a day – the question of whether a system is secure enough for their immediate needs.

In this paper, we will describe our explorations of this issue. In particular, we will draw on three major elements of our research to date. The first is empirical investigation into everyday security practices, looking at how people manage security as a practical, day-to-day concern, and exploring the context in which security decisions are made. This empirical work provides a foundation for our reconsideration of the problems of security to a large degree as an interactional problem. The second is our systems approach, based on visualization and event-based architectures. This technical approach provides a broad platform for investigating security and interaction, based on a set of general principles. The third is our initial experiences in a prototype deployment of these mechanisms in an application for peer-to-peer file-sharing in face-to-face collaborative settings. We have been using this application as the basis of an initial evaluation of our technology in support of everyday security practices in collaborative workgroups.

1 Introduction

Networked computer systems are increasingly the site of people's work and activity. Millions of ordinary citizens conduct commercial transactions over the Internet, or manage their finances and pay their bills online; companies increasingly use the Internet to connect different offices, or form virtual teams to tackle mission-critical problems through entirely "virtual" interaction; For example, interaction between citizens and local and federal government agencies can increasingly be conducted electronically; and the 2004 national elections in Brazil and (to a much more limited extent) the U.S. saw the introduction of electronic voting, which will no doubt become more widespread.

However, these new opportunities have costs associated with them. Commercial, political and financial transactions involve disclosing sensitive information. The media regularly carry stories about hackers breaking into commercial servers, credit card fraud and identity theft. Many people are nervous about committing personal information to electronic information infrastructures. Even though modern PCs are powerful enough to offer strong cryptographic guarantees and high levels of security, these concerns remain.

The need for secure systems is broadly recognized, but most discussions of the "problem of security" focus on the foundational elements of information systems (such as network transmission and information storage) and the mechanisms available to system developers, integrators, and managers to ensure secure operation and management of data. Security, though, is a broader concern, and a problem for the end users of information systems as much as for their administrators. Participation in activities such as electronic commerce requires that people be able to trust the infrastructures that will deliver these services to them.

This is not quite the same as saying that we need more secure infrastructures. We believe that it is important to separate theoretical security (the level of secure communication and computation that is technically feasible) from effective security (the level of security that can practically be achieved in everyday settings). Levels of effective security are almost always lower than those of theoretical security. A number of reasons for this disparity have been identified, including poor implementations of key security algorithms (Kelsey et al., 1998), insecure programming techniques (Shankar et al., 2002; Wagner et al., 2000), insecure protocol design (Kemmerer et al., 1994; Schneier and Mudge, 1998), and inadequate operating systems support (Ames et al., 1983; Bernaschi et al., 2000).

One important source of the disparity, though, is problems around the extent to which users can comprehend and make effective use of security mechanisms. Approaches that attempt to make the provision of system security "automatic" or "transparent" essentially remove security from the domain of the end-user. However, in situations where only the end user can determine the appropriate use of information or the necessary levels of security, then this explicit disempowerment becomes problematic. We have been tackling these problems in the Swirl project. Here, rather than regarding the user as a potential security hole to be "routed around," we attempt, instead, to understand how to create systems in which security is a joint production of technical, human, and social resources.

We will begin by discussing some existing work in this area, before introducing our approach. We will briefly summarize the results of our empirical work and the conclusions that we draw from these investigations, before presenting our design approach and an example of a system based on this approach. We will then briefly discuss some early usage feedback.

2 Explorations in Information System Security

2.1 Previous Approaches

It is broadly recognized that one of the major challenges to the effective deployment of information security systems is getting people to use them correctly. Psychological acceptability is one of the design principles that Saltzer and Schroeder (1975) identify. Even beyond the domain of electronic information systems, there are many examples of the fact that overly complex security systems actually reduce effective security. For example, Kahn (1967), cited by Anderson (1993), suggests that Russian military disasters of the Second World War were partly due to the fact that Russian soldiers abandoned the official army cipher systems because they were too hard to use, and instead reverted to simpler systems that proved easier to crack. Scheiner (2000:373) sums up the situation: "Security measures that aren't understood by and agreed to by everyone don't work."

However, despite this broad understanding of the significant relationship between security and usability, little work has been carried out in this area to date. We discuss some exceptions here.

2.1.1 Usability of Security Software and Mechanisms

In a series of studies, researchers at University College, London have explored some of the interactions between usability and security (Adams and Sasse, 1999; Adams et al., 1997). They focused on user-visible elements of security systems, such as passwords. Although many information systems professionals regard users as being uninterested in the security of their systems (and, indeed, likely to circumvent it by choosing poor passwords, etc), Adams and Sasse's investigations demonstrate that users are certainly motivated to support the security of the system, but often unable to determine the security implications of their actions. The specific problems that they identify with passwords have also led to interesting design alternatives (Brostoff and Sasse, 2000; Dhamija and Perrig, 2000).

In some cases, the complexity of making security work is as much a matter of interface design as anything else. Whitten and Tygar (1999) present a usability analysis of PGP 5.0, demonstrating the difficulties that users have in completing experimental tasks (in their user study, only 3 out of 12 test subjects successfully completed a standard set of tasks using PGP to encrypt and decrypt email.) The problems that they uncovered were largely problems of interface design, and in particular the poor matching between user needs and the structure of the encryption technology provided to meet these needs.

Zurko and Simon (1996) explore similar concerns in their focus on "user-centered security". Like us, they are concerned that the inscrutability of conventional security mechanisms makes it less likely that users will employ them effectively. The approach they outline focuses on graphical interfaces and query mechanisms to MAP, an authorization engine. While this approach is clearly helpful, it is limited to a particular area of system security, and lacks the real-time feedback.

2.1.2 Control Over Security

One area at the intersection of usability and security that has received some attention is the role of access control in interactive and collaborative systems. For example, Dewan and Shen (Dewan and Shen, 1998; Shen and Dewan, 1992) have explored the use of access control and meta-access control models as a basis for describing and controlling degrees of information access and management in collaborative systems. This is not simply a technical matter, since the structure and behavior of these "internal" components can have a significant effect on the forms of interactivity and collaboration they can support (Greenberg and Marwood, 1994).

Many collaborative systems involve privacy issues and need to provide users with control over the disclosure of information. This has spurred a number of researchers to explore the development of privacy control systems that are tailored to the needs of end users. For instance, Dourish (1993) describes the relationship between three different security mechanisms for similar multimedia communication systems, each of which reflects assumptions and requirements of the different organizations in which they were developed. Bellotti and Sellen (1993) draw on experiences with multimedia and ubiquitous computing environments to identify the source of a number of potential privacy and security problems. Their primary concepts – disembodiment and dissociation – are both visibility problems, related to the disconnection between actors and actions that renders either actors invisible at the site of action, or actions invisible to the actor.

Based on their investigations of privacy problems in online transactions, Ackerman and colleagues propose the idea of privacy critics—semi-autonomous agents that monitor online action and can inform users about potential privacy threats and available countermeasures (Ackerman and Cranor, 1999; Ackerman et al., 1999). Again, this mechanism turns on the ability to render invisible threats visible.

One important related topic is control over the degree of security available. One of our criticisms of traditional security systems has been their "all or nothing" approach. However, there has been some work that attempts to characterize degrees of security provision, as embodied by the idea of "quality of security service" (Irvine and Levin, 2000; Spyropoulou et al., 2000). This builds on earlier work establishing a taxonomy of security service levels (Irvine and Levin, 1999). The fundamental insight is that organizations and applications need to trade-off different factors against each other, including security of various forms and degrees, in order to make effective use of available resources (Henning, 2000; Thomsen and Denz, 1997). While this work is directed towards resource management rather than user control, it begins to unpack the "security" black box and characterize degrees and qualities of security.

2.2 Our Approach: Theoretical and Practical Security

Our research was motivated by a series of examples from our own experience that illustrated the problems of effective security, even in technologically sophisticated environments. For instance:

- A research group designing a system for mobile code needed a security solution. A highly qualified academic security expert designed and implemented an elegant scheme based on SPKI/SDSI (Simple Public Key Infrastructure/Simple Distributed Security Infrastructure) in which the system servers would determine transaction rights based on cryptographically secure certificates exchanged over a Remote Procedure Call infrastructure secured using TLS (Transport Layer Security). However, in actual use, this resulted in a performance reduction by a factor of 5-10; consequently, everyone simply turned it off, rendering the system less secure than it had been in the first place.
- A research laboratory used "S/Key" [1] one-time pads to allow terminal access through a firewall host. Researchers would periodically use private passwords and local client programs to generate themselves new one-time password pads. However, the system was soon discontinued when it became clear that people could not tell whether their connections were secure enough to make it safe to generate the new pads.

Elsewhere, we have also observed security problems that emerge from the mismatch between theoretical and effective security, for instance:

- Norton's Anti-Virus software offers an option to check incoming email for viruses before you download it to your computer. The actual mechanism for doing this is not directly disclosed. When this option is turned on, the user's login and password are sent to a Norton server, which downloads the user's email and reads it, checking for viruses, before sending it on to the user. Inserting Norton's own servers as an intermediary makes great technical sense, allowing Norton to respond rapidly to new virus attacks. However, users are typically shocked to learn that their password and their email are being shared with Norton; it damages their trust in the system and in the software.

Each of these cases illustrates a problem at the intersection of security and interaction. They suggest that one important source of the disparity is problems in the extent to which users can comprehend and make effective use of security mechanisms.

As noted above, and as signified by this special issue, a growing number of researchers have been concerned with problems of usability and security. However, our concern is not simply with the usability of security mechanisms, but more broadly how security can manifest itself as part of people's interactions with and through information systems (as the examples here show).

[1] S/Key is a trademark of Bellcore.

Researchers in the HCI community have long argued that "usability" cannot be an afterthought in information system design; a system cannot be made usable merely by the addition of a graphical user interface, however pretty. Security researchers have made a similar argument about the design of secure systems; insecure systems cannot be turned into secure ones merely by the addition of a layer of encryption. Both of these argue, then, that security and usability need to be understood as a holistic design problem. A lick of "usability paint" will not cure the difficulty of making use of, say, X.509 certificates (an International Standard Organization standard that specifies the format of digital certificates) within a broader interactive setting, nor can security be simply integrated into usable applications. We need to look more broadly at how these problems emerge as part and parcel of everyday activity.

As a way of understanding this problem, we began by undertaking a small-scale investigation of end-user security practices, looking in particular at the context within which security decisions are made. This led to a formulation of a research approach based on two primary principles – the dynamic real-time visualization of system state, and the integration of action and configuration. Most recently, we have been developing and evaluating a prototype system for collaborative file sharing based on these principles. Our empirical investigations are reported in more detail elsewhere (Dourish et al., 2004); we will provide a brief overview here, to provide context for later discussion.

2.2.1 Empirical Investigation

Since our concern is with security as a matter of everyday practice, we conducted a brief investigation of everyday user practices around security, interviewing a total of 20 users drawn from a number of distinct groups across two sites. We present a very brief outline here of some key results that have informed subsequent research investigation; Dourish et al. (2004) provides a fuller report on our methods.

Optimization. Security is not, in general, an end in itself for most people in the course of their work; rather, while they have differing needs for security, secure practice is evaluated in terms of its relationship to the task at hand. In some circumstances, this may mean stepping outside the boundaries of conventional security practice, or finding alternative ways to make work secure that allow for more effective accomplishment of necessary activities. This is in line with previous investigations such as those of Westin (1967) and Ackerman et al. (1999), which have pointed to "pragmatism" as a dominant aspect of privacy practice and information security management, but it is not entirely the same as this previous work, since it reflects not a direct trade-off but rather the differential way in which security can be accommodated alongside other needs.

Contingent assessment. People report (and display) not just a wide range of attitudes towards appropriate security practice, but highly contingent behaviors, crafted in response to particular circumstances. That is, the balance between immediate needs and overall security concerns is one that is continually being evaluated and reassessed. There are a number of factors to this: first, the security implications of particular actions may change as a situation develops; second, the other factors affecting the appropriateness of engaging in those activities may change; and third, the relevance of security concerns to the user may also change.

Delegation. Information technology solutions, such as cryptographic network protocols and authentication are certainly some of the facilities on which people rely in order to work securely. Essentially, these are technological "agents" to which responsibility for security can be delegated (Latour, 1992). However, technology is merely one amongst a range of potential agents to whom this responsibility may be delegated. Others include specific individuals (e.g. a trusted family member or colleague who has "set things up" on someone's behalf); another organizational entity (such as a campus or corporate information management group, who are trusted to have ensured that things will work appropriately); or an institutional entity (such as a bank) that may be, by dint of its very nature, deemed trustworthy.

Embeddedness. The boundaries of information systems and the boundaries of working activity are not the same; working activities are embedded in broader frames of physical space, organizational structure, and working practice. A concern with security, then, may be distributed across all of these. Managing electronic information securely may involve the simultaneous coordination of physical, electronic and organizational resources. Similarly, appropriate physical, organizational and working configurations may render otherwise insecure electronic information practices as entirely appropriate and resilient to practicable lines of attack.

2.2.2 Design Implications

Our empirical investigation looked at a wide range of computer users, with different needs and working in different settings. As illustrated above, a number of common issues arose in our interviews. A broad summary would be that "security," both as a need and a practice, extends beyond the domain of the computer system itself.

Computer system security is traditionally regarded as a primarily technological concern; the fundamental questions to which security researchers address themselves are those of the mathematical guarantees that can be made for the performance of various communication and computational challenges. However, our empirical investigation reveals an alternative reading of the problem of security. Here, the fundamental security question is one that end-users routinely encounter and resolve for themselves many times a day – the question of whether a system is secure enough for their immediate needs. Security is evaluated and managed in a range of contexts – physical, personal, organizational, interactional, and more.

We draw four conclusions from these investigations.

First, security in practice is not an all-or-nothing matter. Our alternative security question – whether a system is secure enough for people's immediate need – is a quite different question from the traditional approach, and implies a different sort of answer. It suggests a continual tuning of the degree of security required, and a process of matching security to task. Too much security can be as much of a problem, as too little. Systems that inflexibly offer absolute security are likely as useless as those that offer none (but generally more difficult to configure and use).

Second, the distinction between "secure" and "insecure" settings is not an absolute matter that can be legislated in advance and resolved by the system, but rather a matter for user determination; decisions must be placed into the hands of end users, raising questions of organizational accountability, as noted by Weirich and Sasse (2001).

Third, visibility of mechanism plays a critical role. If the key problem is determining whether the current configuration of systems and services is secure enough for the task at hand, then it is critically important that security features and potential threats be visible so that this determination can be made. Hidden features of infrastructure, including mechanisms designed for secure computation, are inherently unavailable for this sort of examination.

Fourth, security in end-user applications is an end-to-end phenomenon, even though it arises out of the interactions between many different components (Blumenthal and Clark, 2001; Saltzer et al., 1984). Effective security potentially depends upon each application or infrastructure component involved, as well as on the relationships between those components. Although the end-to-end element is a known issue in traditional security circles, it is particularly problematic when we consider visibility and usability as central issues for security infrastructures. When we talk of "distributed applications" or "networked applications", we mean to include not simply the application, but the entire “slice” through the infrastructure needed for that application to work – client, server, network services, protocol implementations, etc.

3 Design Approach for Effective Security

Our goal in undertaking both a broad review of the literature and these empirical investigations has been to understand how best to approach the design of technologies supporting usable security. As we have noted, one design approach involves giving specific attention to the security features of a system, such as those components through which information encryption might be controlled, or through which privacy preferences might be expressed, and tackling the usability problems that typically beset those components. However, as seen from the conclusions listed above, our empirical studies have suggested an alternative approach. Our investigations into user security practices suggest that security concerns cannot be localized within those components of a system specifically designed to address security. In the everyday world, security is not a delineable set of actions, but rather is a pervasive aspect of the ways in which work gets done. Accordingly, our design approach has been to understand and support security as an intrinsic aspect of interactive systems.

In particular, our approach in Swirl is based on supporting informed decision-making. The central problem of security, for end users, is two-fold: it involves understanding the system’s configuration and state, and understanding their consequences for user action. People act through information technology, and so our goal is to help them understand how an information system might mediate their actions. This turns our attention away from traditional considerations of expression and enforcement and towards explication and engagement – how can we provide people with insight into the operation of a distributed system, and how can we couple those understandings to the actions that people take?

We have been exploring these questions through a series of prototypes designed to uncover, demonstrate, and evaluate a range of principles and specific designs (de Paula et al., 2005). In this account, we focus in particular on two design principles – visualizing system activity and integrating configuration and action.

Visualizing system activity gives users a means of understanding and assessing the consequences of their action. By providing dynamic feedback on relevant but hidden aspects of system activity, such as network traffic and configurations, we provide people with a means to understand the relationship between their actions and the technology configuration through which they are performed. It is important to note that this visualization does not take the form of the sorts of network monitoring that might be employed by system administrators or network managers. Clearly, end users neither understand nor care to understand the details of network operation, and so we cannot assume this level of technical expertise. Nonetheless, we find that people can understand and appreciate the temporal and structural correlations between their activities and the system’s behavior. A useful analogy is with driving; even without understanding the detailed operation of the car, a driver can still make use of the sound of the engine, the feel of the clutch, and the feel of the road through the wheel.

One of the primary challenges in designing visual accounts of system security is thus to achieve an appropriate level of expression or description. Clearly, the visual presentations provided must be expressive enough to be useful in making security-relevant decisions. However, at the same time, it is not our goal to provide people with large amounts of information, nor do we intend to require all users to understand all the relevant technical characteristics of security in their system. Our goal is nonetheless to provide people with information – visual depictions of the system’s action – which they can incorporate into their assessments through practice and experience, but which do not rely on complete technical descriptions of the system’s operation.

An important element of our strategy, then, is not to attempt to represent the users' intent, nor their interpretation of current threats. While user modeling approaches of this sort have achieved some degree of success in online applications such as web site personalization, we feel that the domain of user actions in networked systems is insufficiently constrained to apply this approach. Instead, our approach is to have the system present information that it can validly "talk about" – its own internal structure and action. The question to be addressed, then, is in what terms this account should be constructed.

Although there has been a certain amount of research investigating ways of visualizing distributed systems structure, behavior and performance, most of this work has been aimed at system managers and operators. Systems such as Pulsar (Finkel, 1997) or Planet MBone (Munzner et al., 1996) are designed to convey information to highly technical audiences. One exception is in the System Health project (Dourish et al., 2000), which monitored the activity of complex distributed systems in order to convey some understanding of the state of the system to end-users whose work might be affected by outages, slowdowns, and other mysterious "internal" events. However, this work was directed towards fairly general characterizations of systems, rather than focusing on an issue like security. In a more focused area, we anticipate being able to apply heuristics, which can inform a more specialized interpretation of events.

Our second principle is the integration of configuration and action. This reflects the concentration in our account of information practices that these practices are performed, not expressed (and indeed, that expression, when it arises, is itself performance.) We are concerned, then, with engaged action, and with the ways in which people express their security needs through everyday action (Dourish and Anderson, 2005). Studies of privacy and security practice have repeatedly demonstrated a disjunction between expressed needs or intents on the one hand, and actual practices on the other (e.g., Spiekermann et al., 2001). This suggests that approaches that are based on enforcing expressed constraints may be either overly rigid or ineffective. For example, this approach manifests itself, in current operating system designs, in a separation between a control panel where preferences are set, and some separate window or windows within which the activity of the system is performed. This separation is doubly problematic. Not only does it separate two coextensive forms of activity (the act of "sharing" being distributed across the preference window and the system window), but it also separates the expression of preferences from the occasion or situation in which those preferences are to be invoked. Conventional interfaces separate configuration and action in both space and time, although in everyday practice they are one and the same activity. Speaking and vocal modulation (e.g. intonation, volume, etc) are, for example, inseparable aspects of the same activity; similarly, our design approach seeks to make configuration and action part of the same interactional space.

4 Applying The Principles

The two principles – visualizing system state, and integrating configuration and action – are broadly applicable. They have informed the design of a number of prototypes, and are part of a developing design "vocabulary" that is the primary focus of our work. In order to show how we have used them, we will spend some time discussing our most recent application design.

Our current testbed for experimentation is an application called Impromptu. Impromptu is a collaborative peer-to-peer file sharing application for small group synchronous and collocated interaction. Informally, Impromptu can be thought of as an application designed to augment face-to-face meetings by providing a shared data space with zero set-up costs.

Impromptu provides a visual client interface designed to support our two major principles – the integration of configuration and action, dynamic visualization of activity. Figure 1 depicts the Impromptu client interface. The primary interface feature is the circular "pie" corresponding to the shared workspace as a whole in which each "slice" corresponds to a single user's area of the shared workspace. These areas expand and contract as users arrive and leave. Files, represented by labeled dots, are placed in and around the circular region. Each area is tagged, on the pie's perimeter, with a unique color for each user; this color is also associated with that user's files, and with indicators of that user's activity. The organization and orientation of this circular region are consistent for all users, so that informal references (e.g. to "left", "right," or "top corner") can be oriented towards by all (Tatar et al., 1991).

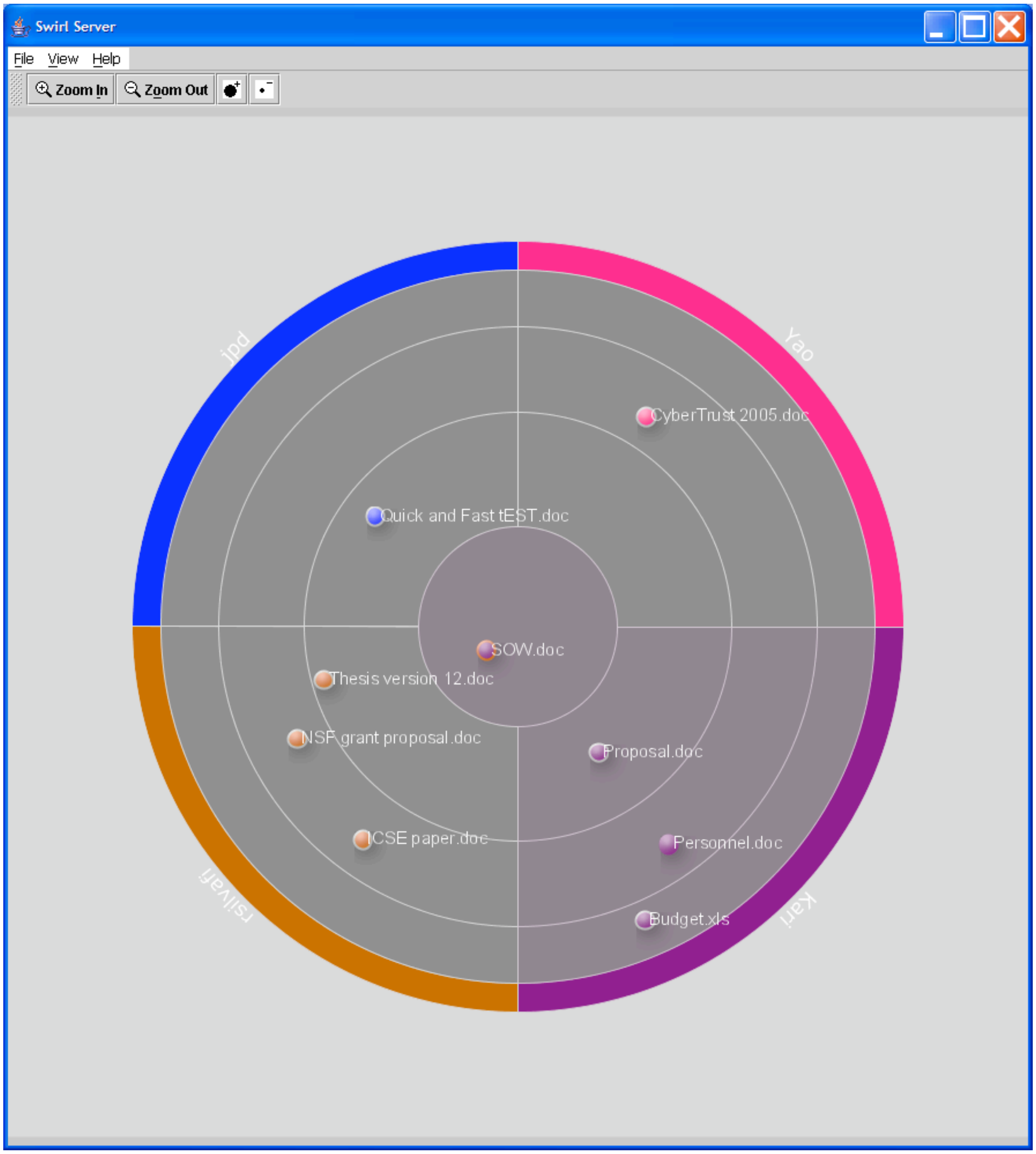

Figure 1: Impromptu Client Interface. Spatial organization integrations configuration and action; color dynamics are used to provide real-time feedback on activities.

The interface is separated into multiple concentric regions; the basic metaphor is that the closer the files are to the center, the "more shared" they are. Various degrees of sharing might be implemented. The particular mappings we have been using are that files outside the circle are not shared at all, but available to the local user; files in the outer region are visible but not readable or writable to others; files in the next region are readable but not writable; in the next, readable and writable; and in the center, readable, writable, and available persistently. Persistent access means that, even when someone leaves the session, his or her files remain accessible to others in the group; by default, files are non-persistent, meaning that when the user leaves the session, their files will disappear from others' interfaces. The provision of persistence serves two functions here, one pragmatic and one research-oriented. The pragmatic motivation is that persistence is a necessary feature of many of our usage scenarios (e.g. information sharing in group meetings); the research motivation is that we wanted to be sure that our different "sharing degrees" did not simply correspond to conventional file access rights. File access is managed by moving the files between the levels. People can only control the accessibility of their own files; moving files onto and off other people's segments initiates a copy operation, if access rights allow. This direct coupling between location and sharing reflects the principle of integration of configuration and action. The mechanisms by which files are shared and by which their sharing is controlled are not separate; rather, they are one and the same.

The dynamics of the interface reflect its concern with the visualization of internal actions. Individual activities are reflected quickly to the group as a whole, for two reasons – first, this ensures that everyone can see potentially consequential actions, and second, it provides individuals with direct visual feedback on the ways in which their own actions are seen by others. This is an important consideration in developing an understanding of the consequences of action. Further, the dots that represent files do more than that; they also represent activities over those files. So, for example, remote file accesses to local files cause the icons for the files to blink in colors that indicate the identity of the user accessing them. This dynamic visual display draws attention to current activity and allows for a quick overview of access patterns.

Since Impromptu is designed as a testbed for principles and design approaches with broader applicability, it is based on a combination of open standards including WebDAV (Web-based Distributed Authoring and Versioning (Goland et al., 1999) - an IETF standard for collaborative editing via extended Web protocols) and the IETF Zeroconf protocols for service discovery (see below). Overall, Impromptu's infrastructure provides a shared working space available across a wide range of system platforms, supporting multiple degrees of sharing, with no pre-configuration. Either using wired or wireless network connections, or 802.11 network cards in "ad hoc" mode (allowing them to communicate directly without the use of an access point), it also operates even when disconnected from the public Internet.

Previous research suggests that peer-to-peer file-sharing may be a fruitful area for exploring design principles. Good and Krekelberg (Good and Krekelberg, 2003) present an analysis of the use of the KaZaa filesharing client. In their study, a number of significant design problems emerge, resulting in inadvertent sharing of sensitive information in a user trial. In particular, they vividly illustrate the central problem to which our principles have been addressed – the difficulty people have in assessing the consequences of activities within the system. In addition to these security considerations, a number of other design constraints pertain. Impromptu was designed to augment face-to-face collaboration by creating a collaborative file space where a group of users can share and exchange information more easily than they might do by, for example, exchanging files using a flash memory drive. This scenario implies four significant constraints. The first is that setting up a collaborative file space should require essentially zero configuration; the overhead must be nil, or close to nil, in order for the application to be effective. The second is that, since sharing is ad hoc, it should require no prior registration of relevant parties. The third is that it should ideally be operable with no fixed infrastructure; it should not require, for example, connection to the public Internet. The fourth is that it should operate on a wide number of platforms.

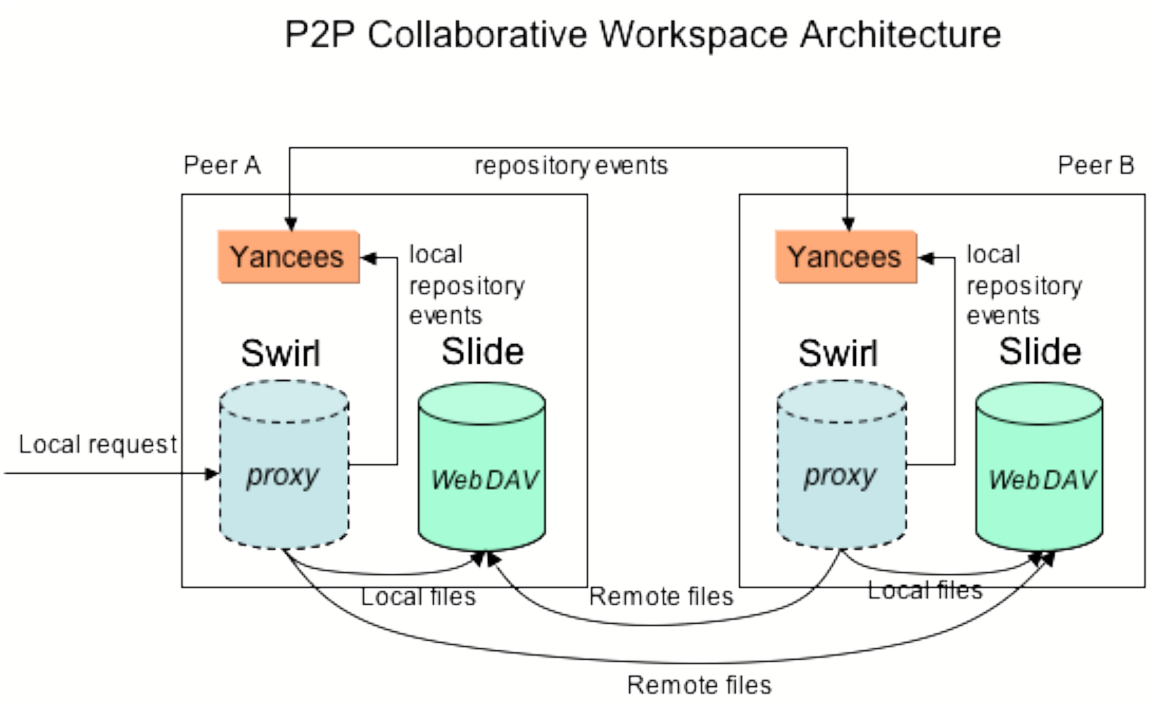

Figure 2: Impromptu Architecture

The Impromptu architecture is illustrated in Figure 2. It uses the YANCEES (Silva Filho et al., 2003) event service, configured for a peer-to-peer setting, to maintain the client Pie views in sync by informing each client of events taking place on the others. The system infrastructure is based on a two-tier WebDAV server. Each user's client has its own WebDAV server, which manages access to shared files. The proxy (see Figure 2) stitchs these separate servers together and creates, on each client, a unified virtual shared space, which is itself managed and accessed through the WebDAV protocol. WebDAV is broadly accessible across platforms both through Web interfaces and also through native filesystem interfaces on a range of systems including Windows, MacOS X, and Linux. In our federated WebDAV model, there is no central server; the system operates entirely as a peer-to-peer architecture in which each "client" is, essentially, also a server and in which no server has a uniquely distinguished role. Shared files, then, are distributed across the set of clients that make up a session, and so when a user leaves, their files disappear from the workspace. When users leave the system, all their persistently shared files are automatically allocated to another machine. In this way, a session persists through multiple arrivals and departures until, finally, there is no Impromptu client running.

One particular challenge in a peer-to-peer workspace implementation is the identification and management of peers that are constantly arriving and departing from the network. We accomplish this using an implementation of the IETF Zeroconf protocols. Zeroconf is a set of protocols that implement peer discovery, address allocation, name resolution, and related services over the TCP/IP protocols. This allows Impromptu clients to find each other automatically with no previous configuration or user intervention. Whenever someone runs Impromptu, it automatically finds and joins other Impromptu clients on the same network. Accordingly, the questions of who is connected, who else might be unconnected but "lurking," etc., are ones that the interface should illuminate. It is worth noting that, as peer-to-peer applications based on these sorts of protocols become more common, they open up both new security opportunities and new security challenges (Voida et al., 2005).

Event architectures are particularly appropriate for our approach to usable security (Dourish and Redmiles, 2002). They provide an integration platform for sharing and visualizing “end-to-end” security-related information. Moreover, through event correlation and analysis, they can be used to detect high-level patterns arising out of sequences of low-level events. The YANCEES event architecture integrates those components together through a common event bus. Event-based infrastructures provide an effective mechanism for flexible, loosely coupled distributed systems integration. Note that by event-based systems here, we refer to a particular style of distributed software architecture in which “events” are data structures that flow through a collective software bus to coordinate the activity of multiple components; this is not the event detection associated with, for instance, intrusion detection systems (Denning, 1987; Lunt and Jagannathan, 1988). YANCEES is the latest in a line of event-based architectures developed by our research group; it is a versatile infrastructure designed for flexibility in extensibility and configurability of its functionality. It also interoperates with a range of other event-based infrastructures such as CASSIUS (Kantor and Redmiles, 2001), Siena (Carzaniga et al., 2001) and Elvin (Segall and Arnold, 1997), and provides a pluggable architecture which can support additional services such as event persistence, event sequence detection, and other features that may be needed by different applications.

5 Initial Usage Experiences

A formal evaluation exercise is ongoing, but it is useful to reflect on some initial experiences introducing people to the use of Impromptu. We conducted three informal pilots, in which pairs of users drawn from our department worked together on a self-selected, real-world task. The goal was to determine the extent to which the design principles that we had adopted facilitated the interpretation of the system in terms of security concerns and impacts on action. Although very preliminary, our experiences provide support for our design approach. Four particular issues stand out from these initial usage experiences.

First, we noted that each group that tried Impromptu used it in a different style and managed their security needs in different ways. Some groups adopted highly integrated working style, while others used Impromptu more as a means to coordinate separate activities. Some shared information to the highest degree, while others used the sharing levels more selectively. As we had hoped, then, Impromptu did not seem to impose a particular working style on the groups using it, but rather provided an open framework supporting different styles of collaboration. This observation confirms our empirical observation about the variety of security practice, and therefore the impracticality of approaches that presume that security needs can be specified within a uniform framework (Dourish et al., 2004).

Second, the integration of action and configuration creates a strong sense of embodiment and sharing. People respond to the shared space of the Impromptu interface as a shared and active space, and the objects within it as truly shared and seamlessly available. In fact, this sense of sharing pervaded not just their use of Impromptu but also their use of other applications running alongside it. Some users expected that, when they opened up shared files in Microsoft Word, their actions in Word would also be shared; that changes to the file made by one user would be immediately visible to other users at the same time, as though Word were a synchronous collaborative tool. We attribute this sense of seamless sharing to the very direct interaction metaphor that follows from the action/configuration link that we have been exploring. Clearly, this has some significant implications for interpretations of security when actions cross application boundaries; those boundaries need to be clearer, so that people can have some understanding of the different policies at work in the different applications that might be used to perform a single task.

Third, we were pleased to see that the interface provides people with a strong sense of the presence of others, critical for interpreting and understanding potential security problems or understanding the limits of information disclosure. During some of our trials, we had an unknown third party surprise participants and appear in the interface. This arrival was clearly visible in the interface, and was apparent to people glancing at the Impromptu window. Further, people's responses indicated that they were, first, clearly aware of the consequences for their own activities, and, second, able to take action in response by, for example, taking shared files and moving them to a read-only space to prevent the new arrival from gaining full access to the content of their work.

Fourth, we can see that it is important to understand aspects of the context in which the system may be put to use. Our previous research had confirmed that security is not confined to the system itself, but rather is spread across the system and the contexts within which it is used. There are two relevant contexts – a physical context and a working context. The physical context of use is face to face collaboration; Impromptu was designed not to support distance or distributed collaboration, but rather as an adjunct to face to face work, permitting people to share information more easily than they might do using other physically co-present mechanisms (e.g. flash drives.) People talked to each other a great deal while using Impromptu, commenting on their actions, describing their plans, and of course talking about the work that they were doing. The use of Impromptu as a support, rather than a replacement, for face-to-face interaction is clearly important in the design. The working context is slightly more problematic. File sharing is rarely an end in itself; it is a means to support other working activities. Impromptu, then, is expected to be used alongside other applications. In our early trials, we noted that these other applications would sometimes obscure the Impromptu system, making it harder to notice changes and updates. We are looking, therefore, at a range of ways of conveying information about shared activities to people, not only through a dedicated interface but also through ancillary displays that can augment other interfaces.

Impromptu is a not a generalized solution by any means; as an application, it is designed to support a narrow and particular range of uses. Similarly, collaborative peer-to-peer file sharing is not the primary focus of our research. However, Impromptu provides us with a useful testbed for a set of visual and design techniques to be applied. In particular, here, we have emphasized the ways in which real-time visualization of system state and integrated configuration and action can create an embodied experience of information sharing and so help people both to develop understandings of activity within a complex distributed system and then to understand and anticipate the consequences of their actions within that system. These principles are part of an emerging design vocabulary supporting an alternative interpretation of the problems of usability and security.

6 Conclusion

Computer and communication security has been an important research topic for decades. However, the pressing concern at the moment is not simply with advancing the state of the art in theoretical security, but with being able to incorporate powerful security technology into the kinds of networked computational environments that more and more people rely on every day. We see the problem of creating a trustable infrastructure – one that end users can see is visibly trustworthy – as a major problem for both the security and the HCI research communities.

Our empirical investigations suggest that this is more than simply a usability problem; rather, we need to think about the ways that security problems manifest themselves in people's work, and how they incorporate security practices into their everyday activities. In other words, we think of security as a joint production of system and user. This focuses our attention on a rather different design challenge; how can we provide people with the tools and facilities that they need to understand and dynamically control security as a part of their existing interaction? Rather than providing mechanisms that take security decisions away from people, we want to develop open and flexible frameworks that allow them to understand the consequences of their actions and develop new forms of practice.

Two design principles have been at the center of our approach. The first is dynamic visualization of system activity, providing people with a means of understanding a system's action, especially when that system is a distributed coalition of computing elements. The second is integration of action and configuration, removing the artificial separation between control over information and its use. In these, we are attempting to build interactive systems that acknowledge how these problems are handled in the everyday world – not simply as matters of privacy or security, but as collective information practices (Dourish and Anderson, 2005; Palen and Dourish, 2003).

Technologically, we have looked towards visualization and event-based architectures as one way to approach this general problem, embodied in Impromptu prototype, an experimental platform for security interaction. More broadly, though, we are interested in the range of ways in which we can provide technological support for an alternative security "problem" – not "what mathematical guarantees can be made about this system" but "is this system secure enough for what I want to do now?"

7 Acknowledgements

This work was supported in part by the National Science Foundation under awards 0133749, 0205724 and 0326105, and by a grant from Intel Corporation.

8 References

Ackerman, M.S. and Cranor, L., 1999. Privacy critics: UI components to safeguard users' privacy. CHI '99 extended abstracts on Human factors in computing systems. ACM Press, Pittsburgh, Pennsylvania, 258-259.

Ackerman, M.S., Cranor, L.F. and Reagle, J., 1999. Privacy in e-commerce: examining user scenarios and privacy preferences. Proceedings of the 1st ACM conference on Electronic commerce. ACM Press, Denver, Colorado, United States, 1-8.

Adams, A. and Sasse, M.A., 1999. Users are not the enemy: why users compromise security mechanisms and how to take remedial measures. Commun. ACM, 42(12): 40-46.

Adams, A., Sasse, M.A. and Lunt, P., 1997. Making Passwords Secure and Usable. Proceedings of HCI on People and Computers XII. Springer-Verlag, 1-19.

Ames, S., Gasser, M. and Schell, R., 1983. Security Kernel Design and Implementation: An Introduction, IEEE Computer, 14-22.

Anderson, R., 1993. Why cryptosystems fail. Proceedings of the 1st ACM conference on Computer and communications security. ACM Press, Fairfax, Virginia, United States, 215-227.

Bellotti, V. and Sellen, A., 1993. Design for privacy in ubiquitous environments. In: G. de Michelis, C. Simone and K. Schmidt (Editors), Proceedings of the Third European Conference on Computer-Supported Cooperative Work ECSCW'93. Kluwer, Milan, Italy, 77-92.

Bernaschi, M., Gabrielli, E. and Mancini, L.V., 2000. Operating system enhancements to prevent the misuse of system calls. Proceedings of the 7th ACM conference on Computer and communications security. ACM Press, Athens, Greece, 174-183.

Blumenthal, M.S. and Clark, D.D., 2001. Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world. ACM Transactions Internet Technology, 1(1): 70-109.

Brostoff, S. and Sasse, M.A., 2000. Are Passfaces more usable than passwords? A field trial investigation. In: S. McDonald, Y. Waern and G. Cockton (Editors), People and Computers XIV - Usability or Else! Proceedings of HCI 2000. Springer, 405-424.

Carzaniga, A., Rosenblum, D.S. and Wolf, A.L., 2001. Design and evaluation of a wide-area event notification service. ACM Transactions Computer Systems, 19(3): 332-383.

de Paula, R., Ding, X., Dourish, P., Nies, K., Pillet, B., Redmiles, D., Ren, J., Rode, J. A. and Silva Filho, R., 2005. Two Experiences Designing for Effective Security. Institute for Software Research, Irvine, CA. Working paper.

Denning, D.E., 1987. An intrusion-detection model. IEEE Trans. Softw. Eng., 13(2): 222-232.

Dewan, P. and Shen, H., 1998. Flexible meta access-control for collaborative applications. Proceedings of the 1998 ACM conference on Computer supported cooperative work. ACM Press, Seattle, Washington, United States, 247-256.

Dhamija, R. and Perrig, A., 2000. Deja Vu: A User Study. Using Images for Authentication, Proceedings of the 9th USENIX Security Symposium, Denver, CO.

Dourish, P., 1993. Culture and control in a media space, Proceedings European Conference Computer-Supported Cooperative Work ECSCW'93. Kluwer, 125-137.

Dourish, P. and Anderson, K., 2005. Privacy, Security… and Risk and Danger and Secrecy and Trust and Identity and Morality and Power: Understanding Collective Information Practices. Irvine, CA: Institute for Software Research. Technical Report UCI-ISR-05-1.

Dourish, P., Grinter, E., de la Flor, J. D. and Joseph, M., 2004. Security in the wild: user strategies for managing security as an everyday, practical problem. Personal Ubiquitous Computing, 8(6): 391-401.

Dourish, P. and Redmiles, D., 2002. An approach to usable security based on event monitoring and visualization. Proceedings of the 2002 workshop on New security paradigms. ACM Press, Virginia Beach, Virginia, 75-81.

Dourish, P., Swinehart, D.C. and Theimer, M., 2000. The Doctor Is In: Helping End Users Understand the Health of Distributed Systems. Proceedings of the 11th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Services Management in Intelligent Networks. Springer-Verlag, 157-168.

Finkel, R.A., 1997. Pulsar: an extensible tool for monitoring large Unix sites. Softw. Pract. Exper., 27(10): 1163-1176.

Goland, Y., Whitehead, E., Faizi, A., Carter, S. and Jensen, D., 1999. HTTP Extensions for Distributed Authoring - WEBDAV. Internet Engineering Task Force, Internet Proposed Standard Request for Comments 2518, February.

Good, N.S. and Krekelberg, A., 2003. Usability and privacy: a study of Kazaa P2P file-sharing. Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, Ft. Lauderdale, Florida, USA, 137-144.

Greenberg, S. and Marwood, D., 1994. Real time groupware as a distributed system: concurrency control and its effect on the interface. Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM Press, Chapel Hill, North Carolina, United States, 207-217.

Henning, R.R., 2000. Security service level agreements: quantifiable security for the enterprise? Proceedings of the 1999 workshop on New security paradigms. ACM Press, Caledon Hills, Ontario, Canada, 54-60.

Irvine, C. and Levin, T., 1999. Toward a Taxonomy and Costing Method for Security Services. Proceedings of the 15th Annual Computer Security Applications Conference. IEEE Computer Society, p. 183.

Irvine, C. and Levin, T., 2000. Quality of security service. Proceedings of the 2000 workshop on New security paradigms. ACM Press, Ballycotton, County Cork, Ireland, 91-99.

Kahn, D., 1967. The Codebreakers, The Story of Secret Writing. Macmillan, New York, NY.

Kantor, M. and Redmiles, D., 2001. Creating an Infrastructure for Ubiquitous Awareness, Eighth IFIP TC 13 Conference on Human-Computer Interaction INTERACT 2001, Tokyo, Japan, 431-438.

Kelsey, J., Schneier, B., Wagner, D. and Hall, C., 1998. Cryptanalytic Attacks on Pseudorandom Number Generators. Proceedings of the 5th International Workshop on Fast Software Encryption. Springer-Verlag, 168-188.

Kemmerer, R., Meadows, C. and Millen, J., 1994. Three Systems for Cryptographic Protocol Analysis. Journal of Cryptology, 7(2): 79-130.

Latour, B., 1992. Where are the Missing Masses? The Sociology of a Few Mundane Artifacts. In: W.E. Bijker and J. Law (Editors), Shaping Technology/Building Society. MIT Press, Cambridge, MA.

Lunt, T.F. and Jagannathan, R., 1988. A Prototype Real-Time Intrusion-Detection Export System, Proc. IEEE Symposium on Security and Privacy. IEEE, New York, 59-66.

Munzner, T., Hoffman, E., Claffy, K. and Fenner, B., 1996. Visualizing the global topology of the MBone. Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96). IEEE Computer Society, 85.

Palen, L. and Dourish, P., 2003. Unpacking "privacy" for a networked world. Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, Ft. Lauderdale, Florida, USA, 129-136.

Saltzer, J.H., Reed, D.P. and Clark, D.D., 1984. End-to-end arguments in system design. ACM Transactions on Computer Systems, 2(4): 277-288.

Saltzer, J.H. and Schroeder, M.D., 1975. The protection of information in computer systems. Proc. IEEE, 63(9): 1278-1308.

Schneier, B., 2000. Secrets & Lies: Digital Security in a Networked World. John Wiley & Sons, Inc., p. 304.

Schneier, B. and Mudge, 1998. Cryptanalysis of Microsoft's point-to-point tunneling protocol (PPTP). Proceedings of the 5th ACM conference on Computer and communications security. ACM Press, San Francisco, California, United States, 132-141.

Segall, B. and Arnold, D., 1997. Elvin has left the building: A publish/subscribe notification service with quenching, Proceedings AUUG'97, Brisbane, Australia.

Shankar, U., Talwar, K., Foster, J.S. and Wagner, D., 2002. Detecting Format String Vulnerabilities with Type Qualifiers, Proceedings 10th USENIX Security Symposium (USENIX'02), Washington DC, 201-220.

Shen, H. and Dewan, P., 1992. Access control for collaborative environments. Proceedings of the 1992 ACM conference on Computer-supported cooperative work. ACM Press, Toronto, Ontario, Canada, 51-58.

Silva Filho, R., de Souza, C.R.B. and Redmiles, D., 2003. The Design of a Configurable Extensible and Dynamic Notification Service, Proceedings Second International Workshop on Distributed Event-Based Systems (DEBS'03).

Spiekermann, S., Grossklags, J. and Berendt, B., 2001. E-privacy in 2nd generation E-commerce: privacy preferences versus actual behavior. Proceedings of the 3rd ACM conference on Electronic Commerce. ACM Press, Tampa, Florida, USA, 38-47.

Spyropoulou, E., Levin, T. and Irvine, C., 2000. Calculating costs for quality of security service. Proceedings of the 16th Annual Computer Security Applications Conference. IEEE Computer Society, 334.

Tatar, D.G., Foster, G. and Bobrow, D.G., 1991. Design for conversation: lessons from Cognoter. Int. J. Man-Mach. Stud., 34(2): 185-209.

Thomsen, D. and Denz, M., 1997. Incremental assurance for multilevel applications. Proceedings of the 13th Annual Computer Security Applications Conference. IEEE Computer Society, p. 81.

Voida, A., Grinter, R., Ducheneaut, N., Edwards, K. and Newman, M., 2005. Listening In: Practices Surrounding iTunes Music Sharing, Proc. ACM Conf. Human Factors in Computing Systems CHI 2005. ACM, Portland, OR.

Wagner, D., Foster, J.S., Brewer, E.A. and Aiken, A., 2000. A first step towards automated detection of buffer overrun vulnerabilities, Network and Distributed Systems Security Symposium.

Weirich, D. and Sasse, M.A., 2001. Pretty good persuasion: a first step towards effective password security in the real world. Proceedings of the 2001 workshop on New security paradigms. ACM Press, Cloudcroft, New Mexico, 137-143.

Westin, A.F., 1967. Privacy and Freedom. Atheneum, New York, NY.

Whitten, A. and Tygar, J.D., 1999. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, Proceedings Ninth USENIX Security Symposium.

Zurko, M.E. and Simon, R.T., 1996. User-centered security. Proceedings of the 1996 workshop on New security paradigms. ACM Press, Lake Arrowhead, California, United States, 27-33.