Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 7 (2012) pp. 366-376
© MIR Labs, www.mirlabs.net/jias/index.html

Usability, Security and Healthcare Systems: Design, Challenges and Perspectives

Tanya Ann Baklanoff and Anish Abraham Padath*

2717 49th Ave. S.W.
Seattle, WA USA
tbklnff@gmail.com

*Machine Intelligence Research Labs (MIR Labs),
Auburn, WA , USA
anishap@acm.org

Abstract: It's well acknowledged that the field of usable security recognizes that to be secured, a system must be usable. Even the most highly secured application system will fail in practice if the intended users cannot use it properly. In this research we focus on commercial of the shelf application end users. Security professionals also have an important role because the consequences of usability problems can potentially leave the entire networks vulnerable to attack. We believe that techniques are needed for building application systems that are usable, scalable and secure. After all it's not the responsibility of the end users to make sure an application is secured but the developer's job to make it usable and secure. So the progress in usability can contribute to the development and maintenance of dependable application system, especially if they can assist with scalability. Usability is an example of two-way interdependence; a system that is not scalable and not dependable is likely to be difficult to use, and vice versa. A system that is not usable is not likely to be dependable. This article aims to provide guidelines for designing and evaluating applications for security, usability, and scalability.

Key Words: *Usability, Security, Scalability, Healthcare, HIPPA, HITECH*

1. Introduction

Historically, security has been seen as an inconvenience, preventing users from doing what they want. However, many usability experts state that security features should not force users through complex steps. Instead, there are better ways to both enforce security and make the systems more usable. Usable security is a field of study that examines the interrelation between security, usability, and scalability in software systems. It is a combination of multiple domains; information security, user centered design, and system development. All of these areas work together to provide a framework for usable security.

1.1 Technology and Users

As Anne Adams and M. Angela Sasse stated, users are not the enemy. A simple security example of passwords show that today users have multiple accounts for work, home, and school. Add to the fact that most systems require users to have passwords with at least than 8 characters, including upper, lower, numeric, and special characters. With some systems requiring password changes every 30-60 days, it is no wonder users are tempted to write passwords down, or try to use easy to remember passwords on less restrictive systems.

A number of studies show that people perceive security as an inconvenience, preventing them from doing what they want. If a system security feature is too complex, too disruptive, and uses unfamiliar terms, users have been known to compromise a system through mis-configuration. If security feature gets in the way of getting to the primary task, interrupts work, or affect performance, users have been known to disable or bypass security settings, not understanding the impact of their actions. Security is not their primary goal; they just want to finish their job [1].

Jerome Saltzer and Michael Schroeder, two pioneers in usable security, state that security mechanisms should not impose unreasonable difficulty in performing a task. "It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanism he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors" [2].

While security awareness education can help, it is only part of the equation. Many people do not want to spend hours going through manuals or researching online sources to figure out how to configure firewall or encryption settings [3]. The best approach is to design secure systems that are transparent to the user, yet at the same time give an indication that the security feature is working, and prevent the user from making errors that would compromise security.

2. Significance of the research

The goal of any security system or feature is to provide Confidentiality, Integrity, and Availability (CIA). Usability can support these CIA triads by preventing accidental misuse. This research study was focused on the interrelation between security and usability and looks at a few examples in healthcare systems. In the first part of the paper we reviewed the literatures of security and usability and impact on the end user from various researchers. The second part we discussed how usable security impact health care systems and a few problem scenarios that end users face in day-to-day work place. The third part of the paper includes interviews with healthcare, security experts, and IT professionals on the state of usability, scalability and security. Finally, we analyzed and evaluate the findings in usability and security and proposed recommendations, future directions to develop a better user friendly and secured application system.

2.1 Perception of usable security

Security experts have largely ignored usability issues both because they often failed to recognize the importance of human factors and because they lacked the expertise to address them. Security professionals need to focus on usability. When we look at security from the user's perspective Jaferian et al [4] points out that controls must be easy to use. It must not require any training, and it must not get in the way of legitimate use. Ideally, the control should be invisible to end-users but it could be visible to super users. These criteria are often difficult to meet, but they are essential in healthcare systems. The alternative is that we end up with a system that is ineffective or not user friendly. The bottom line is that users need to get their jobs done. Anything that gets in the way of this is frustrating and something to be avoided. Many researches in usable security is pointing to increasing agreement in designing secure systems that are usable and scalable, but less agreement about how to reach this goal.

2.2 Building trustworthy systems that are usable

It is evident from research that there is not a large body of security-specific user interface design techniques. This is what the Whitten and Tygar [5] depict in their findings. According to them, there are no recognized exemplars of good user interface design for security, and human-computer interaction (HCI). Even though there has been research on security and HCI, nothing was focused exclusively on security applications. This study is of interest not only because of the findings that we reach, but also because it can serve as an example of how to evaluate the usability, scalability and security of commercial off the shelf application packages. The development of security oriented user interface design techniques requires expertise in security as well as in HCI. Because security concepts are often not easy to understand, and because they must be used flawlessly, an HCI expert who is unskilled in security is likely to make a system where the security mechanisms are not used in exactly the correct manner. This will lead to creating systems that are not usable, scalable and secured.

3. Literature Review

Garfinkel [6] states that, "there are of course no set of rules, principles or formalisms that, when followed, are guaranteed to produce usable computer systems. If such rules existed, we would almost certainly all be using them, and the usability problem would be solved." It's evident that such usable system doesn't exist because developers are not spending time on user interface design or they don't pay attention to usability problems. Payne and Edwards [7] gives a historical relation of design research into the stress between making information secure and keeping the computer systems useful for the tasks they support. They took examples from technologies for end user authentication and email encryption. They also illustrate how usable security isn't exclusively a matter of making interfaces to security measures usable but might also involve deeper structural considerations and the understandings that people bring to security. Recent advances in the research of usable security have produced many new security mechanisms that improve usability. In Parkin et al. [8] we can see that, if the user cannot be expected to understand how to work with the system, the meta-task responsibility is not properly fulfilled. Many situations in security have a similar structure, such as when password requirements are too complicated. With the increase in web applications, email and online banking services, users now have even more passwords today than ever before.

Security experts play a crucial role in the provision of usable and effective security. In [9] researchers recognized that one of the major challenges to the effective deployment of information security systems is getting people to use them correctly. 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. Security is usually a secondary goal. As per [10] the problem is that existing design methods for secure systems do not address goals like systematic process of software engineering, up-to-date knowledge of security threats and do not provide enough support for the developers to realistically achieve them. People do not generally sit down at their computers wanting to manage their security; rather, they want to send email, browse web pages, or download software, and they want security in place to protect them while they do those things. Chiasson et al. [11] states that 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 behavior and the consequences of user actions. Systems need to make users aware of and where necessary, supply them with guidance on the security tasks they need to perform. Another part of this guidance is recommendations support where users are unsure of decisions and their implications [12].

In [13] researchers analyzed a number of collected user stories to understand what happens when everyday users encounter security dependent technologies. They pointed out that there are significant differences between being secure and having a secure experience, and conclude that classical usable security, focus on people's immediate security experience. According to [14], the first and most important approach to building usable security technology is to attempt to build what we call implicit security into applications to unify the often "separate but (un) equal" views the user is forced to have of applications goals and security operations. In fact, security lives in separate parallel universe that the user must act on in addition to whatever actions are needed to "directly" accomplish their desired task. Most research on security in ubiquitous computing has taken the technical approach [15]. From the technical perspective, one approach has explored ''transparent'' security infrastructures systems that are secure without even needing the user to be aware [16]. While this offers the potential advantage of freeing the end user from the need to understand the mathematical concepts that form the basis of the security solutions they will use, it comes with drawbacks also.

The migration to electronic health records and the passage of the Health Information Technology for Economic and Clinical Health legislation spotlight the importance of usable security in health information technology. As per [17], in many areas of health care, workarounds are epidemic. Struggling to cost-effectively meet patients' needs while balancing regulatory demands and ever-changing technology, nurses might improvise to circumvent failed processes. Confidential data hemorrhaging from health-care providers pose financial risks to firms and medical risks to patients. In [18], the researchers present evidence of the threat by examining user-issued searches. Their analysis demonstrates both the substantial threat and vulnerability for the health-care sector and the unique complexity exhibited by the US health-care system. Better compliance with both the security and privacy rules is certainly needed. Of course, HIPAA can do little to stop patients from disclosing their medical identities voluntarily to individuals posing as health care providers, or poorly managing their own computerized documents. According to [19] firms across all business sectors struggle with data security problems and it is unlikely that there is a prescribable out-of-box solution that will work for all parties handling EHR. Any platform that does become widely adopted will become a larger and larger target for parties seeking to exploit EHRs for personal and financial gains. In the end, the hope is that health professionals will respond to the offered incentives and that the HITECH Act will make health care faster, less expensive, and higher quality. The focus of this paper is security during the early transition as HITECH security rules became effective.

There are both security and usability experts that offer guidelines for designing usable security systems. In addition they outline reasons usable security can be so challenging.

Ka-Ping Yee proposes ten guidelines for design (Appendix A)[20]. Whitten and Tygar state that security is a difficult domain for usable design. They outline five properties of security that make it problematic:

- the unmotivated user
- abstraction,
- lack of feedback
- barn door (once a secret is revealed)
- weakest link (the attacker only needs to find a single weakness).

Security software is usable if people are aware of the task they need to perform, understand how perform those tasks, do not make errors, and are comfortable with the interface [21].

Garfinkle suggests that there are no set of guidelines that would guarantee to produce a usable system. It may not be possible to design a system to be usable for all scenarios. Part of usability is to focus on the target audience, whether it be the home user or an administrator in a corporate environment [22].

There are other sources for security guidelines for software development. Build Security In, sponsored by the Department of Homeland Security, provides resources and guidelines for software developers and security practitioners to build secure systems [23]. The SANS Institute is a research and educational organization that provides resources to security practitioners, including the Top 20 Programming Errors [24].

Grady Booch talks about the different forces that affect software development and design. Ultimately business needs, system complexity, its environment, and operability all affect the outcome of the product. A larger system with many stakeholders is more difficult to build functionality and usability than a smaller system with a small group of users [25]. Verdon discusses the importance of developers understanding and following corporate security policies [26]. Doing so reduces a company's exposure to lawsuits. He states that best practices in secure software development are easily obtained, and are now becoming the norm. It's no longer acceptable to claim no knowledge of secure development practices or have security policies that support such practices.

According to Lampson, users do not have an understandable security model. They know if they click OK on a dialog box they can continue their current task [27]. Sasse and Adams outline the reasons users cannot follow recommended password practices – multiple accounts, password content, compatibility with work practices, and user perception of organizational security and information sensitivity [28].

What is clear to developers may not be to the end user. Furnell evaluated both a browser and a word processing program. He offers suggestions to improve usability, such as using consistent labeling with easy to understand terms. Grouping similar features together under the same sections and menus would also helpful [29]. In "Security Beliefs and Barriers for the Novice Internet Users" Furnell and his team interviewed several users who did not have a good understanding of the risks and where to go to for information. They reported having problems with security software and ended up disabling features to avoid interruptions and pop-up dialog boxes. Another suggestion he proposes is to take reliance and decision making out of the hands of the novice user and enable security features by default with self auditing systems [30].

Encryption is another example of a complex security task for many users. In "In Search of Usable Security", computing experts struggle to set up a PKI system for their wireless security. Advanced users struggled to obtain certs and configure wireless on their end devices; there were thirty-eight steps to go through and the average time took 140 minutes to complete. To improve the situation, they developed a system to automate many of the steps. The enrollment time went down, however, when they tested it on end users, they had difficulty that was not anticipated. The study supports conventional wisdom that encryption is difficult to use [31].

4. Improvements and Challenges in Usable Security

This is not to say that there haven't been improvements in usable security over the past several years. OS and antivirus updates are now automated; people no longer have to remember to update them. On the other hand, frequent downloads and reboots are seen as intrusive; users might be tempted to disable or ignore the features that are supposed to provide protection. Passwords are problematic; multifactor authentication is recommended - a combination of something the user has, knows, and/or is stronger than a simple password.

Other improvements to usability are ultimately not secure. Out of the box systems, such as wireless routers, come "plug and play" ready. Default settings enable easy set up and installation, but it also enables hackers to easily break in. Attackers are able to access systems using default administrative passwords, many of which can be found online [32].

Today many large corporations are incorporating security and usability into their software development lifecycle. They recognize the need for vulnerability and pen testing to find weaknesses in the software and to correct them. Building security and usability early into the project's lifecycle is a best practice, and correcting problems is easier and less costly earlier in the project. Retrofitting usability and security after release is difficult to do, and ends up costing more money in the long run [33].

4.1 Usability Testing

A common challenge cited in creating usable systems is the amount of time and resources to do usability testing in the first place. Steve Krug and Jacob Nielsen, both experts in the field of usability, propose low cost methods to address schedule and budget concerns. In "Usability Testing on Ten Cents a Day", Krug stresses that some form of testing is better than none at all. One only needs a conference room, a user, and a computer. Low fidelity prototypes, heuristic testing, among other methods can uncover many usability problems early in the design phase. A small group of testers can uncover about 80 percent of problems with an interface. Elaborate testing can be a waste of resources; the rate of return levels out after the number of testers increase [34].

4.2 Changing Target

Another challenge is that security is a changing target. New technology and systems are developed every year, and with that, new vulnerabilities and threat vectors (smartphones, for example). No sooner does an administrator close a threat vector, hackers and APT find new and unique ways to exploit a system. It has been compared to an arms race; as defenses get better so do the skills of hackers [35]. Social engineering is a reliable method that has stood up over time.

Peter Mitnik, a well known hacker, did most of his work through social engineering tactics. The point is, security is not a static state, it must be maintained.

To meet the most significant challenges in the healthcare, technology infrastructure requires secure and effective systems for improving the quality of care and controlling costs. Usability is an issue for Electronic Medical Record (EMR), so the fact is that most of the healthcare industry is now experiencing another security issue with regard to mobility, as we know many medical staffs want to access patient data on the go via their mobile devices. If the information technology department in healthcare has not properly planned for this contingency, serious security problems will definitely present themselves.

4.3 Awareness of security

Our findings indicate that users are aware of security threats and vulnerabilities but they believe that it's their IT department job to make sure they are working on a secured network. The research study by Chan et al. [36] investigates the power of security and self-efficacy on end user compliance to security policies and procedures. This finding suggests that engaging the management and staffs of the organization can create an information security environment.

4.4 Responsibility

It is evident from literature review and interviews that the users themselves have considerable influence on their perception of the security environment. This clearly states that management should take necessary steps for conducting security awareness programs in addition to implementing policies and procedures. The lessons learned in these programs should be applied in their work, which can help to create a strong information security environment at the work place.

5. Healthcare Problem Scenarios

Many healthcare security researchers' point to government regulations likes the HIPAA and the HITECH regulatory and compliance. Generally they do raise the bar to a minimum security issues but generally only for the organizations, which see these requirements merely as bothersome necessity.

5.1 HIPPA Whom and What does it protect

HIPAA is a multifaceted document that covers a large number of health care situations, many of which have nothing to do with electronic PHI. It has changed dramatically since its inception, as changes in the way that medical information is communicated and acted upon have changed the objectives that make up the overall goal of the Act. With the passage of the HITECH act in 2009, HIPAA was modified to encourage the adoption of electronic health records, and to move the question of electronic PHI privacy to the forefront of HIPAA compliance.

5.2 HITECH legislation

The Health Information Technology for Economic and Clinical Health Act (HITECH Act) legislation created to stimulate the adoption of electronic health records (EHR) and supporting technology in the United States. President Obama signed HITECH into law on February 17, 2009 as part of the American Recovery and Reinvestment Act of 2009 (ARRA), an economic stimulus bill.

The HITECH act stipulates that, beginning in 2011, healthcare providers will be offered financial incentives for demonstrating meaningful use of electronic health records (EHR). Incentives will be offered until 2015, after which time penalties may be levied for failing to demonstrate such use. The Act also establishes grants for training centers for the personnel required to support a health IT infrastructure [37].

5.3 Healthcare Application – Usability & Security issues

Research shows that deploying new features and technologies in healthcare applications without considering information assurance and security makes patient privacy vulnerable. Additionally, patient identifiable and/or healthcare data of an individual are highly sensitive. Hence, security is a vital requirement of healthcare applications, especially in the case of patient privacy. This paper aims to initiate discussion on these critical issues since the success of healthcare application depends directly on patient security and privacy as well as usability of healthcare providers. In addition, we discuss the issues with existing security tools, and outline the important security requirements for such applications.

An application that is highly secured may not be usable and an application that is highly user friendly may not be secured. After all it's not the responsibility of healthcare provider to make sure an application is secured. But they have to make sure that they are accessing the application from a secured network. The platform on which a provider run the healthcare application presents security vulnerabilities that can be exploited by wireless attackers, and therefore must be considered has a threat not only to the provider but to the entire organization where the provider is working. Currently most healthcare providers are using handheld devices and mobile devices/PDA, so that they can have global access to patient information and clinical information. These mobile devices have global connectivity and are easily integrated into the healthcare providers workflow. But are they really secured? We know from research and security experts that wireless networks have rapidly become a popular area for hackers and games like war walking is a popular pastimes activity for the hacker community.

5.4 Workarounds

It's a human nature to try workarounds when something doesn't work according to their need or expectation. This can create problems not only to the user but to the entire organization. In a healthcare environment timely reports or information is very critical. So obviously the user is required to send information as soon as possible. In some cases the healthcare application may not work as expected or a feature that was previously working may be disabled in the new version for security reasons. In those scenarios the user may be forced to try workarounds since the expectations are very high and they are suppose to send information sooner than later. In our research we found that when trying workarounds the users were not aware of the security risks because for them timely reporting was more important. This doesn't mean that workarounds always creates security risk. What we are trying to emphasis here is that when a feature is disabled; make sure it won't create risk before trying workarounds. This is clearly a usability issue since it's the developer's responsibility to make sure when a feature is disabled there should be an alternate way to do what the user needed.

5. Cloud Computing

The future of information technology in healthcare is unclear. With the advent of smart phones and technologies, increasing amount of patient information, on-demand data services are becoming a necessity for healthcare system users. As noted above, health-care organizations need to adhere to various stringent IT regulations like HIPAA and HITECH. Therefore, to comply with these regulations and deploy IT controls, the tasks become more challenging for IT Security professionals to evaluate security controls over the cloud. More over nobody is sure about how the scalability, Usability and Security issues will be addressed in cloud environment. Anyway as per [38], it's important for health professionals to determine if cloud computing can provide them a secure, reliable, scalable, and inexpensive computing platform that can be used to facilitate health care customers' HIPAA-compliant applications and data.

6. Interview Research

We interviewed IT professionals as well as observed health care users to better learn the state of usable, scalable security (Appendix B). Many of those we interviewed had first hand experience with the problem of using and administering systems that were secure, usable, and scalable.

Generally, all participants agreed there is a link between usability, security, and scalability within a system. One interviewee commented that while these features are interrelated, each must be addressed separately from an architectural point of view. It is important to also note that many participants acknowledged that the software industry is already addressing usability and security issues, but more needs to be done.

To many, a scalable system means it systems must be able to support the expected use. For example, an e-commerce site should be able to accommodate millions of visitors. Ease and ability to manage systems and support changes is also a factor in both usability and scalability.

Similar to what we found in the literature, many interview participants stated that that relying on users to implement security poses a risk. For example, in one scenario users had physical access to a firewall; when their system was blocked after an OS upgrade, they rerouted the cabling to bypass the firewall in order to get their work done.

Automatic patch updates may have improved the state of security, but many have observed both experienced and inexperienced users disabling this feature. Some participants acknowledge that many antivirus systems are too resource intensive. One example cited a product that slowed pc performance to a crawl, so they had to remove it. In another example, one user had to flatten and reload a system after downloading a free antivirus application.

During the course of the interviews, some participant suggested improvements. One suggested that pop up notifications and dialog boxes could both educate users about security and issue warnings when security features are disabled. Another suggestion was to present users with questionnaires during software installation. The answers could be used to evaluate a user's level of comfort or technical ability, and make some configurations automatically.

Others said pop-up dialog boxes could be considered unusable, since many users are now conditioned to click through windows in order to get where they want to go. Scare ware and malware takes advantage of this, many duplicate the same look and feel and trick users into downloading a virus or disabling a security patch. One participant called it the psychology of presentation – a polished, professional looking user interface is attributed to being more secure. The presentation can mask the efficacy of the software.

A system is trustworthy if the vendor is trustworthy and there is confidence that security is configured properly. Some have observed COTS application security settings are difficult to use due to unclear labels, inconsistent terminology, and poor documentation. This is similar to Furnell's evaluation of browser and word processing applications.

Power users prefer to be in control of configuring security settings, versus letting the vendor decide. One participant suggested that single enterprise security settings might be too restrictive for all users. There should be two tiers of security settings; one for most of the user community, then another customizable set for those in development or support roles (the specifics of how restrictive their current security policies were not discussed to know if this would be recommended from a security or usability perspective).

We asked was about obstacles to creating usable secure systems. Time, budget, and poorly defined requirements are common problems in system development. Security and vulnerability testing is considered too expensive and time consuming. Some larger companies may have the resources in place, but this is not always the case with smaller organizations or contractors. Additionally, many do not understand the planning involved in creating usable systems.

A few participants commented that the user interface alone is not the only aspect of usability in security. Training, technical support, and online help are other components. It is not just about a great user interface with people able to intuitively figure out what to do, or automatic settings that run in the background. While its true users do not want to spend and evening reading about firewall configurations, they do want a way to find help that easily accessible and digestible, something that can help them figure out what to do so they can go on their way. People have time limitations, so while reading a manual is not considered "usable", easy to reach context help or technical support is.

Despite careful planning and testing, people find ways to use systems in ways it was not intended. User research is key to understanding the different use case scenarios; user needs, and paint points. Research does not need to be time consuming or expensive. As stated earlier, information can be obtained through phone interviews, surveys, or focus groups. One participant suggested adding documentation on what the system was *not* intended to do.

Legacy systems are another challenge to usable security. Often a critical system does not get regular upgrades because it must be up 7x24. It ends up costing more in the long run and brings a greater risk in operations if an upgrade does not work as intended. There is not usually a spare system or lab in which upgrades can be tested, and many components that are part of the legacy system may be end of life, or too expensive to re-create in a lab setting. Some systems run on older OS that is no longer supported by the vendor. To upgrade would require both a hardware and software replacement, taking the system out of commission and introducing a large capital cost.

Other important points to usable security were:

Trust is a factor in usable security. People must have confidence in configuration settings, that security will work as expected, and not have to reconfigure or recheck settings after upgrades. From a layman's perspective it can be difficult to really know if a system is working, or how it's working. One participant commented that you can't really know. So they trust systems until they read about company breaches where credit card and password information has been stolen.

Well-defined use cases are important. Scenarios and intended uses should be identified and documented, as well as supported environments. How does the system scale under large installations or heavy traffic use? Document what is supported, as well as what is *not* supported.

When developing systems, don't just talk to administrators, talk to power users as well. Interview people or do focus groups.

Systems should be generic enough that they can share common backend and front end operations. Systems should also have low false positives. Too many false positives are intrusive and the customer ends up not taking the alerts seriously.

One participant identified cloud based security as the next movement towards usable security, such as endpoint protections and data center infrastructure protection. Many enterprises find cloud based solutions usable from a management perspective.

7. Findings

From both the literature review and professionals interviewed Table [1], more work needs to be done to improve usability, security, and scalability.

Many participants have observed end users disabling security features; some for functionality and performance, others for temporary troubleshooting purposes.

There are best practices for designing usable secure and scalable systems but they are not widely adopted or recognized for various reasons. Best practices in security and usability are still looked at as separate items.

For training there are conflicting findings. Some say a truly usable and secure system should not require any training, while some observe that security training can at least make users aware of threats and the implications of their actions.

Description	Literature Review	SME Interview	End User Interview
Security, usability, and scalability are not the primary features of most commercial of the shelf vendor applications	Several researchers pointed out that it's not an easy task for the developers to realistically achieve them	Experts agree that the major challenge was getting end users to use them correctly.	We came to know from interviews that it's not their priority to think about how important usability, security and scalability are.
A balance is needed for the best combination of all three features to work interdependently	From literature review we can see that currently there is no specific guidelines or rules	Most agree that these features are interrelated. Issues with scalability and security affects usability and vice versa. One feature should not diminish the others. Some suggest that each feature needs to be examined separately then integrated as a whole.	End users agree that a balance is needed
Security, usability, and scalability issues are not unique to software	There is increasing agreement in designing secure systems that are usable and scalable, but less agreement about how to reach this goal.	Usability and security issues also apply to online games, e-commerce sites, and integrated systems. There are also issues pertaining to support and maintenance of systems as well.	In healthcare, usability and security issues apply to many clinical devices that transfer data between applications.
Many developers and users are not aware of security threats	As per researchers developers are not spending time on usability issues or they don't pay attention to usability problems.	Poorly defined user requirements, budget and time are common problems in system development	By observing end users it was evident that they are not aware of all security threats or vulnerabilities associated with usability issues.
Disabling features to continue primary tasks is still common practice today	Numerous literature reviews cite users will disable security settings that interrupt work or misconfigure settings they do not understand	End users disable security settings some do it out of necessity when a security setting such as anti-malware is too intrusive and slows down performance. Administrators will disable settings for implementation and testing purposes only, though human error results in settings remaining disabled.	In commercial of the shelf application, end users are given the least privileges depending on their job requirement. However from observation they have a tendency to try workarounds if something doesn't work according to their expectation.
There are many different possible use case scenarios and complex environments that affect usability and security	Although some literature discussed how users operated systems in ways that was not intended, none specifically addressed how the wide variety of different use cases and complex environments effect usability, security, and scalability	It is impossible to test for every scenario and environment out there. What would help is to test for the target customers. Also consider documenting what scenarios are not supported.	People have used systems and applications in ways that were never intended by the manufacturer.
Time, budget, organizational culture, and lack of awareness are the top reasons why systems are created with usability, scalability, and security issues	As per many researchers, building security and usability early into the project's lifecycle is a best practice, and correcting problems is easier and less costly earlier in the project.	Most SME's agree that organization culture plays a major role in implementing security controls as well as adopting to new security technologies.	Majority of end users are not aware of the reasons or are not concerned about knowing them.

Table [1]: Summary of Research findings

By observing healthcare end users over a period of time it was evident that usability failures sometimes lead to user workarounds and security failures. In some cases healthcare users create excel spreadsheets or local access database because the enterprise system's user interface is not user friendly or unusable. Such data in spreadsheets might later be manually entered into a healthcare system. Such workarounds can create information risks and data loss. From our research we found that healthcare segment undergo such data loss with multiple consequences like privacy violations, mortification and medical identity theft.

Another common theme is to automate security settings. These should work in the background, rather than rely on users. At the same time users want the assurance that they're protected, that security features are working.

Ideally security features should interact with all systems, with little reconfiguration. This is not an easy task, given the number of applications, platforms, and operating systems available, the variants of versions available, not to mention the different possible use cases.

Clearly identifying the way the system is supported, including use cases, can help users and administrators understand what will work and what will not.

Table 1 outlines additional findings from interviews and observation:

8. Recommendations

Moving healthcare information out of access database; excel spreadsheets and word documents into a secure healthcare system will definitely reduce various types of data loss that we observed in our research. Healthcare providers should consider different measures to protect against data loss that arises due to usability issues.

Training

While systems should be intuitive to use, a well designed user interface is only part of the equation. Some training in security systems can help educate users on what is expected of them as well as the proper way to use the system. Users will better understand the risks and impact of their actions (or inaction) when it comes to securing their systems. Online accessible help should also be available to users to help them quickly find the answers they need to get them on with their work.

Understand Principles Behind Usable Security

Jacob Nielsen advocates the following qualities for system and application usability:

- Understandability
- Visibility
- Feedback
- Error prevention
- Recovery
- Flexibility
- Efficiency

Design Appropriately for both System and User

Know both the audience and use case. For example, many people may appreciate the extra security steps involved to access their credit card and bank account, but for an online game or social media it would become unusable. Always test the product with users during the development lifecycle. What may seem intuitive to developers is not to the target users. Take notes on difficulty of performing tasks, and get user feedback through interviews, focus groups, and observation. Document likes and dislikes, what works and what does not. User testing need not be expensive or time consuming. In addition, heuristic testing can be used to evaluate a system as well.

Scalability

Plan and build for security and usability early in the system development lifecycle. Forrester and others state that it is more expensive to retrofit problems post production.

Understand Vulnerabilities and Weaknesses in Systems:

Usability should not take precedence over security. There are several resources and best practices available to developers and designers to understand and prevent what SANS refers to as dangerous programming errors. Applications and supporting systems should be evaluated for dependencies and potential threats. These should be documented as well as create plans to mitigate. Security testing can also be conducted to reveal vulnerabilities.

The following are just a few questions that could help evaluate security, usability, or scalability of any system, consider the following questions:

- Are security settings enabled by default, or will the user need to enable or disable settings?
- Are unused services enabled or disabled by default?
- Does the system enforce strong password choices?
- Are there visible indications that security is enabled?
- Do alerts and pop-up windows provide sufficient information? Or are users conditioned to click through to return to the task at hand?
- Are security setting easy to find with clear labels?
- Does the security features work in the background or do they interrupt or slow down performance?
- Are terms and labels clear and consistent? Do they provide meaningful definitions?
- Can users find answers to their questions easily through online help? Or will the user need to spend considerable time to learn to configure the system properly?
- Do patches and updates maintain configuration settings? Or do settings have to be reconfigured after updates?
- How well does the system interact with other systems, especially other vendor's products?
- Can the system maintain performance under heavy use?
- What are the requirements to maintain the system?

9. Future Directions

More research would be useful in investigating how issues of organizational cultures can be influenced from the perspective of improving usability and security. Although the first step was taken by identifying end users issues in healthcare application, further studies and testing should be done on a real-time application by observing end users closely on usability issues on a regular basis. This will help application developers and security experts to understand some of the common usability issues and security flaws in commercial of the shelf application packages. While this would not address scalability in particular, but will address issues of new risks and security flaws in application packages, also it would provide a useful support for assessing the majority of commonly know usability issues.

Another important factor to consider for future research is the user's behavior. This may be determined from how they connect to a particular situation and their continued use may depend on how they narrate their understanding with usability. If errors happen while dealing with usability issues, it may be due to how the users anticipate the erroneous experience, and recovering may depend on how the users reflect on their understanding. Subject matter experts and end users should evaluate application and security together before bringing the application to live. This may bring up the know-how approach of both end users and developers which can help to understand issues more thoroughly.

10. Conclusion

Though there may not be widely adopted guidelines for creating usable, scalable, secure systems, the guidelines do exist. There is not a single approach to all systems and use cases. The security and use case scenarios are unique to each application. Professionals can combine best practices from both disciplines without sacrificing the efficacy of the other. Awareness is a key factor, as organizations pay more attention to the issues facing both security and usability, we hope that these practices become common sense in design and development.

11. Acknowledgements

We would like to thank Ajith Abraham [39] and Barbara Endicott-Popovsky [40] for the guidance and support, which helped us to finish the research in a timely manner. We also extend thanks to those (anonymous) professionals who agreed to participate in the interviews; their experience and observations have been invaluable.

References

[1] J. Saltzer, M. Schroder. “The Protection of Information in Computer Systems”. In *Proceedings of the IEEE 63:9*. 1975.

[2] K. Yee. “Guidelines and Strategies for Secure Interaction Design”, *Security and Usability*, O’ Reilly Media, Sebastopol. 2005.

[3] J. D. Tygar, A. Whitten. “Why Johhny Can’t Encrypt”, *8th USENIX Security Symposium,* Washington, D.C. Aug. 23–36, 1999.

[4] P. Jaferian, D. Botta, K. Hawkey, K. Beznosov. “A Case Study of Enterprise Identity Management System Adoption in an Insurance Organization”, *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, pp.46-55, November 07-08, 2009, Baltimore, Maryland.

[5] A. Whitten, J. D. Tygar. "Usability of Security: A Case Study", (CMU-CS-98-155, Pittsburgh, Pennsylvania: Carnegie Mellon University Computer Science Department, 18 December 1998).

[6] S.L. Garfinkel. *Design Principles and Patterns for Computer Systems that Are Simultaneously Secure and Usable*, PhD thesis, Mass. Inst. of Technology, 2005.

[7] B. D. Payne, W. K. Edwards. "A Brief Introduction to Usable Security," *IEEE Internet Computing*, vol. 12, pp. 13-21, 2008.

[8] S. Parkin, A. van Moorsel, P. Inglesant, M. Sasse. “A Stealth Approach to Usable Security: Helping IT Security Managers to Identify Workable Security Solutions”. In *Proceedings of the 2010 Workshop on New Security Paradigms*, pp. 33–50. ACM, 2010.

[9] P. Dourish, D. Redmiles. 2002. “An Approach to Usable Security Based on Event Monitoring and Visualization”. In *Proceedings of the New Security Paradigms Workshop* 2002 (Virginia Beach, VA).

[10] I. Flechais, M. A. Sasse, S. M. V. Hailes. “Bringing Security Home: A Process for Developing Secure and Usable Systems”. In *ACM/SIGSAC New Security Paradigms Workshop*, 2003.

[11] S. Chiasson, R. Biddle, A. Somayaji. “Even Experts Deserve Usable Security: Design Guidelines for Security Management Systems. In *Proceedings of USM2007: Usable IT Security Management, Pittsburgh, Pennsylvania*, 2007.

[12] J. R. C. Nurse, S. Creese, M. Goldsmith, K. Lamberts. “Guidelines for Usable Cybersecurity: Past and Present”, In *The 3rd International Workshop on Cyberspace Safety and Security (CSS 2011) at The 5th International Conference on Network and System Security (NSS 2011), Milan, Italy*, 6-8 September, 2011.

[13] S. Furnell, “Security Usability Challenges for End-Users,” in *Social and Human Elements of Information Security: Emerging Trends and Countermeasures*. IGI Global, pp. 196–219, 2009.

[14] N. Mathiasen, S. Bodker. “Threats or Threads: From Usable Security to Secure Experience?”, *Proc. NordiCHI,* pp. 283-290, 2008.

[15] T. Kindberg, K. Zhang. “Secure Spontaneous Device Association”, In *Proceedings of the 5th International Conference on Ubiquitous Computing (Ubicomp 2003)*, Seattle, Washington. Lecture notes in computer science LNCS 2864, Springer, Berlin Heidelberg New York, October 2003.

[16] M. Blaze. “A Cryptographic File System for UNIX”, In *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS’93)*, Fairfax, Virginia. ACM Press, New York, pp. 9–16, November 1993.

[17] M. E. Johnson, N. D. Willey. *Usability Failures and Healthcare Data Hemorrhages*, (9) No. 2, pp. 35-42, March/April 2011.

[18] M.E. Johnson, “Data Hemorrhages in the Health‐Care Sector,” *Lecture Notes in Computer Science*, R. Dingle- dine and P. Golle, eds., LNCS 5628, Springer-Verlag,□pp. 71–89, 2009.

[19] M. Sturdevant, “1.5 Million Medical Records at Risk in Data Breach,” *The Hartford Courant*, 19 November 2009, www.courant.com/health/hc-healthbreach1119. artnov19,0,1798384.story.

[20] K. Yee. “Guidelines and Strategies for Secure Interaction Design”, *Security and Usability*, O’ Reilly Media, Sebastopol. 2005.

[21] J. D. Tygar, A. Whitten. “Why Johhny Can’t Encrypt”, *8th USENIX Security Symposium,* Washington, D.C. Aug. 23–36, 1999.

[22] S.L. Garfinkel. “Design Principles and Patterns for Computer Systems that Are Simultaneously Secure and Usable”, PhD thesis, Mass. Inst. of Technology, 2005.

[23] Retrieved from http://buildsecurityin.us-cert.gov/bsi/home.html

[24] Retrieved from http://www.sans.org/top25-software-errors/2009/

[25] G. Booch. “From Small to Gargantuan”, *IEEE*

Software, pp. 14-15, July/August 2006.

[26] D. Verdon. “Security Policies and the Software Developer”, *IEEE Security and Privacy. IEEE Computer Society*, 2006.

[27] B. Lampson. “Usable Security: How to Get It”, *Communications of the ACM*, (52), November 2009.

[28] A. Adams, A. Sasse. “Users Are Not the Enemy”, *Communications of the ACM*, December 1999.

[29] Furnell, S. Making Security Usable: Are Things Improving? *Computers and Security* (26), pp. 434-443, 2007.

[30] S. Furnell. “Security Beliefs and Barriers for Novice Internet Users”, *Computers and Security*, (27), pp. 234-240, 2008.

[31] D. Balfanz, G. Durfee, D. Smetters, R. Brinter. “In Search of Usable Security: Five lessons From the Field”, *IEEE Security and Privacy*, pp. 19-24. September/October 2004.

[32] Retrieved from http://www.phenoelit.org/dpl/dpl.html and http://www.default-password.info/.

[33] D. West. “Best Practices: Software Development Processes”, *Forrester*, Retrieved from http://www.forrester.com/rb/biz_process, 2009.

[34] S. Krug. *Don’t Make Me Think! A Common Sense Approach to Web Usability*, pp. 140-153. New Riders, Berkley, 2006.

[35] B. Popovsky, D. Frincke. “Adding the Fourth “R”: A Systems Approach to Solving the Hacker’s Arms Race”, *Proceedings of the 2006 Symposium 39th Hawaii International Conference on System Science*, 2006.

[36] M. Chan, I. Woon, A. Kankanhalli. “Perceptions of Information Security in the Workplace: Linking Information Security Climate to Compliant Behavior”, *Journal of Information Privacy and Security* 1 (3), 2005.

[37] An Introduction to the HITECH Act, Meaningful Use and Nebraska’s Regional Extension Center: NAHQRS April1, 2011.

[38] D., McCament, J. Kane. Retrieved 11/27/2012, from considerthesourceblog: http://www.considerthesourceblog.com/consider_the_source/2010/06/preparing-your-organization-for-hipaa-compliant-cloud-computing.html

[39] A. Abraham, C. Grosan, C. Martin-Vide. “Evolutionary Design of Intrusion Detection Programs”, *International Journal of Network Security*, (4), No.3, pp. 328-339, 2007.

[40] B. Endicott-Popovsky, D. A. Frinke, C. A. Taylor. “A Theoretical Framework for Organizational Network Forensic Readiness.” *Journal of Computers*, (2), No 3, May 2007.

Author Biographies

Tanya Ann Baklanoff received a BA from the University of Alabama (1990), and has earned various network and security certifications from the University of Washington. She received a Masters Degree in Information Management from the University of Washington (2012).

Ms. Baklanoff has several years working in IT, specifically working in network support, engineering, and security for manufacturing and business environments.

Anish Abraham Padath received his BS from Mahatma Gandhi University, India (1992), Masters Degree in Computer Application from Bharathidasan University, India (1996) and Masters Degree in Information Management (Specialized in Information Assurance and Security) from University of Washington (2011).

Mr. Padath is a senior member of Association of Computing Machinery (ACM). He has over 17 years experience in System Analysis and Management, Risk Assessment, Security, Healthcare application support, and Implementation of various healthcare projects.

Appendix A

Ka-Ping Yee Ten Guidelines for Secure Design

- Path of least resistance - match the most comfortable way to do tasks with the least granting of authority. Users prefer the least effort, most familiar, and most obvious methods.
- Active authorization - grant authority to others in accordance with user actions indicating consent. Users should know what systems or users are accessing their systems.
- Revocability - offer the user ways to reduce others’ authority to access the user’s resources.
- Visibility - maintain accurate awareness of others’ authority as relevant to user decisions.
- Self-awareness - maintain accurate awareness of the user’s own authority to access resources.
- Trusted path - protect the user’s channels to agents that manipulate authority on the user’s behalf.
- Expressiveness - enable the user to express safe security policies in terms that fit the user’s task. Security policies should be expressed in familiar, consistent language and concepts.
- Relevant boundaries - draw distinctions among objects and actions along boundaries relevant to the task. Decide which underlying actions to show or hide (for example, all the background activity from clicking on a single web link or uninstalling an application). Hide activities that does not matter to the task at hand.
- Identifiability - present objects and actions using distinguishable, truthful appearances. Applications with similar names or appearances can result in choosing the wrong action (phishing for example).
- Foresight - -indicate clearly the consequences of decisions that the user is expected to make. Information needed to make a decision should be apparent before action is taken.

Appendix B

A list of questions was prepared to get input on what the current challenges are as well as ideas for improvement:

1. What role do you have in the following activities? How long have you been in this role?
   a. Project management
   b. Developer
   c. Security Specialist

   d. Application subject matter expert
   e. Other

2. Describe what usable, scalable security means to you?
3. Do you consider usability, scalability, and security to be independent application/system features or interrelated? Why or why not?
4. In your experience, have you or have you observed users disabling a system's security setting that was interrupting a task? Can you describe a specific example? Was it temporarily or permanently disabled?
5. Considering COTS software systems, how easy is it for end users to find and configure security settings (encryption, host firewall, etc.)?
6. How confident are you that a security feature is working to your expectation? What indicators would you need to feel a security feature is working?
7. Have you ever had to reconfigure system settings after a patch or OS update? Have you had to rollback a patch or update to a previous version? If yes, what were the circumstances?
8. How would you assess usability or scalability of security software? Security features within an application?
9. Some might argue that some systems should be more difficult to use. In what cases do you see that security should take precedence over usability and scalability?
10. Do you believe there is sufficient security awareness amongst end users? Why or why not? If not, what would you suggest to improve awareness?
11. Do you believe there is sufficient security and usability awareness amongst developer? Why or why not? If not, what would you suggest to improve awareness?
12. Can you think of an example of a system that does an effective job of incorporating all three features together? A system that does not incorporate all three features together?
13. What are the challenges to creating systems and applications that are secure, usable, or scalable? What suggestions would you have to address these challenges?