Usable Security: Oxymoron or Challenge?

D. K. Smetters
Computing Science Laboratory
Palo Alto Research Center

Security is rapidly emerging as one of the greatest challenges of our modern, computer-centric society. One of the least well-addressed factors key to achieving effective computer security is that of *usability* – too often users and their pesky focus on the tasks they are actually trying to accomplish are considered primarily as an impediment to systems security, rather than the reason for building those systems in the first place.

Over the last several years, there has been a rapid expansion of research into making systems that are both usable and secure. Beginning with studies that simply characterize the overwhelming flaws of current technologies designed without any thought to the user, it has expanded into interest from the Human-Computer Interaction (HCI) community in the form of work on designing improved user interfaces to existing security technologies, in the hopes of increasing their usability. While both of these approaches play important roles, we have argued (Smetters & Grinter, Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications, 2002) (Balfanz, Durfee, Grinter, & Smetters, In Search of Usable Security – Five Lessons from the Field, 2004) that only through designing (or re-designing) security technologies and secure systems from the ground up with usability in mind will result in systems sufficiently usable and sufficiently secure to meet the demands of modern computing environments.

In this paper I will briefly review recent work both on improving the usability of security technologies and in designing systems to be simultaneously usable and secure, with an eye towards the challenges still faced in marrying these two seemingly opposing goals.

Passwords: The Old Standby

There is no more common "security technology" in use today than that of the ubiquitous user name and password. Easy to understand and implement, passwords play a critical functional role in online society as the secrets that bind real people to their digital information and personas. They require no more of their users than to simply remember them. Unfortunately, they are flawed in both nature and execution – as the simplest form of secret, they are easily given away and reused[1] by attackers. As such, they and other similar forms of "secret" information such as social security numbers are subject to increasingly sophisticated "phishing" attacks, which attempt to trick users into revealing them. They are also deployed in a fashion that overtaxes users' abilities to manage them securely and effectively. A number of studies have demonstrated clearly what most password users (i.e., all of us) know by experience – people don't know how to pick good passwords, and are asked to remember far too many of them and use them, or change them, far too often, resulting in poor password choice, and passwords being written down or shared (Adams & Sasse, Users are Not the Enemy: Why Users Compromise Computer Security Mechanisms and How To Take Remedial Measures, 1999).

Standard password policies are striking in how effectively they minimize usability. Even simple changes in typical password policies can demonstrably increase usability without decreasing security, for example not requiring password change, or increasing the number of password input errors allowed without requiring administrator intervention (e.g. from 3 to 10) (Brostoff & Sasse, Ten strikes and you're out: Increasing the number of login attempts can improve password usability, 2003).

[1] In contrast to more effective approaches utilizing cryptography, which allows secrets such as keys or passwords to be used without being revealed in a form that allows an attacker to intercept and re-use them.

Unfortunately, the increasing body of research on both traditional text passwords and various forms of new graphical passwords, appears to be moving inexorably to the conclusion that passwords, in any form, cannot be used securely in their “naked” form -- i.e. based only on what a user is able to remember on their own, without technological support. Most interestingly, it seems the very universality of passwords is also their downfall. For example, studies suggested that a simple technique for constructing passwords through mnemonic phrases could result in passwords that were as difficult to guess as randomly-generated passwords, while at the same time being easy to remember (Yan, Blackwell, Anderson, & Grant, 2005). Unfortunately, it turns out that in practice, most of us pick the *same* mnemonic phrases from which to generate our passwords, making these “hard” passwords vulnerable to simple dictionary attacks (Kuo, Romanosky, & Cranor, 2006).

Because of their strong appeal, increasing amounts of design activity is going into attempts to improve the security and usability of passwords, with mixed results. *Mutual Authentication* systems, such as SiteKey™, attempt to reduce the risk that users will reveal passwords to other than the web sites they intend by providing a supposedly user-friendly image-based method for users to authenticate the web site they are communicating with prior to entering their password. Unfortunately, recent research shows that users do not notice the absence of the correct SiteKey™ security indicators, still entering their passwords into possibly malicious web sites (Schechter, Dhamija, Ozment, & Fischer, 2007). *Two-factor Authentication Systems* require users to present something other than just a password for access, and attempt to ensure that the other “factor” required is one that is difficult for an attacker to steal. However, perhaps in order to minimize changes required to infrastructure, such systems have been deployed most commonly in a form that gives users one-time passwords to be entered in addition to their standard credentials, either from a list or generated automatically by an electronic token, rather than giving them cryptographic keys to use to authenticate themselves. Unfortunately these one-time password systems have shown themselves vulnerable in practice to attackers who interpose themselves between the user and the resource as a so-called “man in the middle”, capturing the password from the user and then handing it to the resource provider to gain access, while returning an innocuous-looking error message to the original user (e.g. “You typed your password incorrectly, please try again”). *Password Managers* range from simple software assistants that help users keep straight their bewildering number of passwords, to complex pieces of software that provide a number of defenses. The best of these, for example PassPet (Yee & Sitaker, 2006) and Web Wallet (Wu, Miller, & Little, 2006), act among other things to significantly strengthen passwords as a security measure, by asking users to remember a single, master password, from which they generate unique passwords for every site a user logs into. By enabling users to have distinct passwords for each site they access, these tools can not only minimize the damage caused by a single stolen password (which is usually significant, as passwords are frequently reused in practice), but can detect or prevent a user from entering a particular password into the “wrong” site (e.g. a phishing site). While promising, the most widely deployed of these tools (e.g. the simple password managers built into web browsers) lack the significant protective features required to make a significant impact on password theft. Ideally new tools, combining the best features from all existing systems, will begin to achieve widespread deployment in the near future.

Starting from (mostly) scratch: Usable Wireless Security

In the parlance of the introduction, passwords are an existing security technology that have been subject to research both attempting to characterize their flaws, and to improve their usability through the design of better interfaces and support tools. In this section, we turn to the question of combining new and existing security tools to make qualitative changes in users' experience of secure computing systems – making systems so attractive for their usability that the fact that they are also secure is simply a nice bonus.

Take the simple problem of securing a wireless network, or WLAN. The security options available to a home user to protect such a network involve providing every device on that network with the same secret key. Removing an unwanted device from the network requires changing that key on every other device. And until sometime in 2006, the protection afforded to users that performed all of these steps correctly was in name only – the security mechanisms (WEP, or "Wired Equivalent Privacy") built into IEEE 802.11, the most common wireless LAN standard were completely ineffective (Walker J. , 2000) (Stubblefield, Ioannidis, & Rubin, 2002). In 2006, an improved version of the security standards for wireless LANs brought protection to end users, but still at a cost – vulnerabilities in the shared-key based security intended for home users allowed offline guessing of secret keys, and require users to enter very long keys (e.g. 26 hex digits). While recent work by manufacturers to simplify the key distribution process for home users (e.g. WiFi Protected Setup) has improved the usability situation somewhat, it has not made available to those users the more sophisticated forms of WLAN security geared towards enterprises.

Enterprise WLAN security, in contrast, offers a number of alternatives with much higher security guarantees than that provided to home users. At the limit, enterprise WLAN users can be individually authenticated using digital certificates, and provided with separate keys for encrypting data; this provides strong authentication and network access control, the ability to revoke individual users' access easily, and protects network users from each other. However, availing oneself of such high security requires deploying a Public Key Infrastructure and issuing digital certificates, something considered so difficult that even most professionally-managed enterprises do not attempt it.

Our fundamental approach to building systems that are both usable and secure is to focus on the user and their application task (which is all they really care about anyway), and see if we can arrive at system designs that allow them to accomplish those tasks both effectively and securely, without adding additional requirements or burdens. In (Balfanz, Durfee, Grinter, Smetters, & Stewart, Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute, 2004), we turned that approach to the problem of deploying secure WLANs.

One of the most significant usability problems in deploying Public Key Infrastructures, even in enterprise environments, is the idea that such PKIs must be designed in the grand-scale, global infrastructure form in which such tools were first envisioned. If instead, one changes one's focus to building small-scale PKIs, requiring no interoperability or trust with any other infrastructure, and targeted only to the task at hand – say the small group of devices allowed onto one wireless LAN, they turn out to be simple tools that are easy to deploy and manage (Balfanz, Durfee, & Smetters, Making the Impossible Easy: Usable PKI, 2005). By building certification authority software (the sort of software that issues digital certificates) into wireless access point, as well as an enterprise-style authentication server configured to allow only devices certified by that authority onto the wireless LAN, we created a standalone, "home use"-style AP that automatically configured itself to provide a highly secure WLAN as soon as it was turned on for the first time, with no user intervention. (In a later enterprise form of the same system, we wired together our system for enrolling new WLAN clients (see below) into a standard enterprise certification authority and authentication server to achieve the same effect while allowing enterprise-level management of digital certificates and network access policy.)

The other, more critical, challenge in designing a user-friendly approach to securing WLANs is in how the user accomplishes the one task they must do – namely specifying the WLAN they wish their device to join, and indicating to their access point which devices should be allowed to do so. In the simplest usage of wireless LANs, networks are totally unsecured, and users simply “fall onto” any available network. However, it is not possible to do that and achieve any level of security, particularly the high levels of security we are attempting to provide here. Instead, we would like to ask the user to do the minimum work possible – namely to indicate to the network that they would like to join it, and from that simple indication, achieve fine-grained security and control.

We achieve this through the use of a technique we refer to as *demonstrative identification.* It is not possible for two devices who share no *a priori* trust information to authenticate each other over an unsecured medium such as a WLAN without risking a so-called “Man-in-the-Middle Attack”. So instead of having our candidate wireless device and AP find each other over the airwaves, we ask the user to “point out” the AP hosting the network they desire to join over what we refer to as a *location-limited channel*. Such a channel is one where the nature of the medium itself makes it difficult for an attacker to **transmit** information in the channel without being detected – for example, infrared (as with a remote control), physical contact (as with cables or USB tokens), sound, etc. Over this channel, we have the user’s device and the access point exchange **public** information – the cryptographic digests of their public keys

– which can later be used to allow the devices to authenticate each other over any channel they happen to use to communicate (e.g. the WLAN itself).

From the user's point of view, they are merely indicating their desired WLAN by pointing out the AP (or an enrollment device acting as a proxy for the AP) using infrared, as if they were using a remote control. From the AP's point of view, it implements a simple access control policy that says if a user is able to walk up to it and communicate with it over infrared (or audio, physical connection, etc.), that user/device is allowed on the WLAN. Having exchanged public key authenticators in this manner, along with a certain amount of configuration information, the user's device can set up a secure, authenticated connection to the AP over the existing wireless network. The AP can then use that connection to download to the device a digital certificate sufficient to allow it to authenticate as a user of the WLAN using standard protocols, and a small amount of software is sufficient to automatically configure the device to use that certificate in this way in the future. So from the point of view of the user, a small demonstrative act, which is in experimental tests perceived as simpler than the amount of manual configuration required to get a device onto a network providing lower levels of security, is all that is required to set their device up on a highly secure WLAN, and after that initial setup phase, use of the secure WLAN is no more complicated than using any other WLAN. From the point of view of the WLAN owner, they have established a highly intuitive security perimeter – to get on their WLAN, someone must have physical access (e.g. sufficient to communicate over infrared) to the access point. Without such access, they cannot get onto the network. Therefore securing your WLAN becomes equivalent to locking your front door, or if that isn't enough, locking the AP in a closet. And at the same time, the resulting network provides best-available enterpriseclass WLAN security, with per-user encryption keys and the ability to revoke access by any device at any time without requiring reconfiguration of any other device.

This simple system serves as just one example of how taking a slightly different, user-focused approach to the design of secure systems can result in systems which are easier to use than their insecure counterparts, while providing very high degrees of security. We have used this approach over the last several years to construct a number of such proof-of-concept systems, as well as components and tools that make building such systems easier in general.

References

Adams, A., & Sasse, M. A. (1999). Users are Not the Enemy: Why Users Compromise Computer Security Mechanisms and How To Take Remedial Measures. *Communications of the ACM , 42*, 40-46.

Balfanz, D., Durfee, G., & Smetters, D. K. (2005). *Making the Impossible Easy: Usable PKI.* (L. F. Cranor, & S. Garfinkel, Eds.) O'Reilly Media, Inc.

Balfanz, D., Durfee, G., Grinter, R. E., & Smetters, D. K. (2004). *In Search of Usable Security – Five Lessons from the Field.*

Balfanz, D., Durfee, G., Grinter, R. E., Smetters, D. K., & Stewart, P. (2004). Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute. *Proceedings of the 13th USENIX Security Symposium.* San Diego, CA.

Brostoff, S., & Sasse, M. A. (2003). Ten strikes and you're out: Increasing the number of login attempts can improve password usability. *Workshop on Human-Computer Interaction and Security Syste ms, part of CHI2003.*

Kuo, C., Romanosky, S., & Cranor, L. F. (2006). Human selection of mnemonic phrase-based passwords. *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* (pp. 67-78). New York, NY, USA: ACM Press.

Schechter, S. E., Dhamija, R., Ozment, A., & Fischer, I. (2007). The Emperor's New Security Indicators. *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy* (pp. 51-65). Washington, DC, USA: IEEE Computer Society.

Smetters, D. K., & Grinter, R. E. (2002). Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications. *New Security Paradigms Workshop '02.* ACM.

Stubblefield, A., Ioannidis, J., & Rubin, A. D. (2002). Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. *Proceedings of the 2002 Network and Distributed Systems Security Symposium (NDSS'02).* San Diego, CA.

Walker, J. (2000). *Unsafe at any key size: An analysis of the WEP encapsulation.*

Wu, M., Miller, R. C., & Little, G. (2006). Web wallet: preventing phishing attacks by revealing user intentions. *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* (pp. 102-113). New York, NY, USA: ACM Press.

Yan, J., Blackwell, A., Anderson, R., & Grant, A. (2005). The Memorability and Security of Passwords. In L. F. Cranor, & S. Garfinkel, *Security and Usability: Designing Secure Systems that People Can Use.* O'Reilly & Associates.

Yee, K.-P., & Sitaker, K. (2006). Passpet: convenient password management and phishing protection. *SOUPS '06: Proceedings of the second symposium on Usable privacy and security* (pp. 32-43). New York, NY, USA: ACM Press.