Usable Privacy and Security
Introduction to Security

January 26, 2006
Michael Reiter
Notes By: Sasha Romanosky

What is Security?

One way to help define computer security is to classify the types of misuse against computer systems (Neumann and Parker 1989). Another way is to discuss the controls that are used to protect them from loss to any of the following three properties: Confidentiality, Integrity and Availability. Table 1 presents some examples of common security controls and how they apply to various components in a networked computing environment.

	Physical	**Network**	**Host**	**Application**	**Policy**
Preventative	Big scary dogs, data center cages, security guards, fences, door locks, etc	Packet filtering firewall, router with access control lists	Host firewall, authentication (ssh, console, etc), patch management, change control, vulnerability scanning, host hardening	Authentication, patch management, change control, vulnerability scanning, secure coding,	Corporate security and acceptable use policies, government regulation, documentation
Detective	Surveillance cameras, alarm systems, motion sensors	Network intrusion detection system, passive vulnerability scanner	Host intrusion detection system, file integrity checking, system logging	Application logging and monitoring, anti-virus	Auditing procedures, paper trails
Corrective	Mantraps, automatic gates, alarm systems	Network intrusion prevention system, any inline security device	Host intrusion prevention system, reboot, shutdown,	Account lockout, failing safely, real-time anti-virus	Performance evaluation, reprimand, evacuation,
Recovery	Fire suppression, water sprinkler	High availability routing protocols (VSRP, HSRP)	Windows secure file recovery, Re-imaging	File and data backup	Incident response procedures

Table 1: Security Controls

Preventative security controls are those that prevent attacks or loss from occurring and help ensure a "desired state"[1]. Detective controls help identify attempted attacks or unauthorized activity but do nothing to stop them. Corrective controls serve to neutralize an existing attack and help reduce any potential losses from it. Recovery controls attempt to return the system to a known good state and "ensure continuity of the business."[1]

These concepts help address many types of abuses such as spam and surfing while on business time. For instance, surfing the internet during business hours would be accounted for by a preventative control enforcing a corporate acceptable use policy. Since spam is an abuse against the availability of email systems, a preventative application control would help reduce the loss.

Cryptography

Computer security often brings to mind thoughts of cryptography and encrypted messages. And no one has sent and received more encrypted messages across an untrusted channel than "Alice" and "Bob". These names were probably chosen to be humorous yet illustrative and help better

describe cryptographic protocols. Wikipedia also provides examples of other similar names used. See http://en.wikipedia.org/wiki/Characters_in_cryptography.

Symmetric vs. Public Key Cryptography

Because symmetric cryptography is computationally less expensive, it is typically used to encrypt bulk data like the full text of messages or data files. Let's consider that Alice were to use a symmetrical encryption algorithm to encrypt a message for Bob. After performing the operation, she would end up with a chunk of encrypted data and a symmetric key.

However, the trouble now becomes: how does Alice securely transmit both her encrypted message and the symmetric key to Bob across a potentially untrusted channel? She would use public key cryptography, of course! Alice would encrypt the symmetric key with Bob's public key and send both the encrypted message and the encrypted symmetric key to Bob. Bob would then use his private key to decrypt the message to reveal the symmetric key, then use the symmetric key to decrypt the actual message.[1] This is, in fact, a popular method of communication and is used by protocols such as SSL/TLS and PGP.

Getting Back to Usability

When users interact with these kinds of security systems, they are often presented with a suite of new terminology such as encryption, ciphertext, hash function, MAC, certificate, certificate authority, digital thumbprint, etc. The question for usability designers is: should the user be forced to know what these terms mean? They just want to communicate privately, right?

Just as with user interfaces in other disciplines, metaphors are often employed to help users relate and understand the purpose of the underlying action or device. For example, the SSL padlock, key, key ring, digital certificate, digital signature, digital thumbprint, key hygiene, etc, are examples of cryptographic metaphors. Some of them do, indeed, work but sometimes the metaphor breaks down and creates confusion. For example, what does it mean to sign a message with a private key? Moreover, the consequences are often not properly communicated to the user in a form they can understand and appreciate.

Consider Figure 1. What does it mean when the name on the certificate does not match the name of the site? Is the site still safe to visit? Viewing the certificate information (Figure 2) confirms the problem. The link we visited was https://mail.yahoo.com, but this certificate is for http://login.yahoo.com. But should the user care? How can one be sure?

[1] Crystal clear, right?!

Figure 1: Invalid Certificate

Figure 2: Yahoo! Certificate Information

What are the usability issues around all this? Well, for starters, users probably don't want to have to deal with any of this. All they really want to do is use their computer safely and not have to worry about any of this.

One other issue that users face when dealing with public key cryptography is that public keys are only useful when other people have them. (Remember how Alice used Bob's public key to encrypt a message for him? How did she get his key in the first place?) The problem is: how does Alice, or anyone else, for that matter, communicate their public key to anyone else? Moreover, if she changes her key, how does Bob know to get a new one? Key management (or more specifically, certificate management) for both SSL/TLS and PGP is a major problem and they each have different solutions[2], none of which are easy, or make for very usable systems.

Access Control

Access control has become more and more important because it is one of the foundations for the financial auditing that corporations must address in order to comply with Sarbanes Oxley, HIPAA and other regulations. Access control generally consists of authentication and authorization. Authentication is the act of confirming that some one[3] is who they claim to be. Authorization is the act of providing the user with access to only the resources they are entitled to. From a usability point of view, the user generally only sees the authentication component and while this may seem like a simple issue, the problem of how to easily, quickly and securely authenticate a user is still unresolved.

Consider for a moment, a user on one computer requesting a file from another computer over a network. The request will pass through many types of software and hardware layers, with each necessarily interfacing with the layer above and below it. There may be great ways to design the

[2] SSL/TLS certificates commonly use a dedicated protocol for verifying the status of a certificate and PGP assumes a "web of trust" where everyone simply shares their certificates with everyone else.
[3] This "some one" could be either a human or an application

components individually, but they become susceptible to abuse and accidents when combined together. So while a system may operate securely on its own, it may not be as secure when integrated with other components and may fail because of weaknesses, or exposures in the applications, or logic of the system as a whole. With more and more components that interact, the more risk there will be for loss (of confidentiality, integrity or availability) through deliberate or accidental incidents. This is one example of where security experts describe complexity as the bane of secure systems.

Authentication

Two-Factor Authentication

Two-factor authentication is frequently presented as a solution to password-only systems. While may be true, there will be costs. For example, consider the cost to the user of having to learn and use a new authentication device. Consider the cost to the organization that has to purchase, implement and support this new mechanism. Nevertheless, large enterprises frequently deliver SecurID tokens to their employees for remote (VPN) access. Ecommerce websites such as ETrade are also delivering these to their customers for added protection of their online account.

These tokens are considered more secure because they require "something you know" (the password) and "something you have" (the hardware token, or key fob[4]).

Graphical Authentication

While most password systems are text (character) based, people are researching alternate means of authenticating users. Systems such as graphical passwords exploit the recognition ability of humans by having a user remember set of images. While they have been shown to assist with memorability, these schemes introduce their own issues (such as the time to log into the site) which may present a usability issue.

Does a reliable password system necessarily make it more usable?

Biometrics

A biometric identifier used for authentication relies on the "something you are" paradigm. As shown in Table 2, and Table 3, however, biometrics have many other uses and issues.

Description of Use	Examples of Use
User Authentication and Access Control	Iris, fingerprint, voice, hand, face used for authentication and access to secret areas
Recognition	- Voice recognition systems used in customer service (IVR) systems.
Identification	- Fingerprints and DNA used to identify criminals - Biometrics used in passports - Facial identification used in airports to identify terrorists - Iris biometric used to identify the "Afghan girl[5]"
Fraud prevention	Fingerprints used to dispense pension[6], social grant payments and food stamps[7]
Point of sale	Expedited checkout and loyalty programs (Kroger[8], thriftway[9])

[4] A "fob" is defined as a "A short chain or ribbon attached to a pocket watch and worn hanging in front of the vest or waist" (http://dictionary.reference.com/search?q=fob). And from this the term was adapted to mean anything attached to a key ring. Ironically, used as a verb, this definition means to deceive some one.
[5] http://www.npr.org/programs/morning/features/2002/mar/girl/
[6] http://www.businesssolutionsmag.com/Articles/1998_02/980207.htm
[7] http://www.newswithviews.com/Ryter/jon46.htm

Table 2: Biometric Uses

Problem / Issue / Concern	Description
False Positive aka False Acceptance, False Match, Type I errors	When some one is wrongly identified; when they have been identified incorrectly. For example, an imposter.
False negatives aka False Rejection, False non-Match, Type II errors	When some one is not identified when they should have been. I.e. incorrectly rejected. For example, some one who provides a fingerprint with a glove or dirty finger
Invasiveness	While a face or hand biometric may not be personally invasive, an iris or retinal scan can be
Trust	- When a biometric is lost or stolen you can't change or revoke it like a password or certificate - Can you really trust the organization that stores your biometric (or even the template of your biometric)? - Also consider functional creep: will a biometric become the next SSN?
Privacy	- Some biometric identification can be done without a person's knowledge. For example, fingerprint, voice, face, hand - Retinal scan can detect medical conditions of the user - Even though only a template of the biometric is stored in a system (vs the actual biometric image), it is still a unique identifier for that individual
"Liveness" Test	Each biometric system needs to be able to differential between the actual biometric and a facsimile of one. For example a color photo copy of a face, or a gummy bear replica of a fingerprint[10]
Usability	- A usable system is one where all users are capable of providing the biometric you require (finger, hand, etc) - There can be the stigma of criminality when requiring a fingerprint from users

Table 3: Biometric Issues

Trust vs. Trustworthiness

Trust, like privacy, can be considered a personal state. Trust has been defined as "a psychological state comprising the intention accept vulnerability…"[11] Trustworthiness, on the other hand, "asserts that the system does what is required – despite environmental disruption, human user and operator errors, and attacks by hostile parties – and that it does not do other things."[12]

There have been studies that have shown that people are more inclined to trust as website if looks or seems more professional or legitimate. Indeed, this is the attack vector by which much social engineering has been accomplished.

Consider also a system that is not usable, can be considered trustworthy? One approach is to view the trust, trustworthiness and usability as shown in Figure 3.

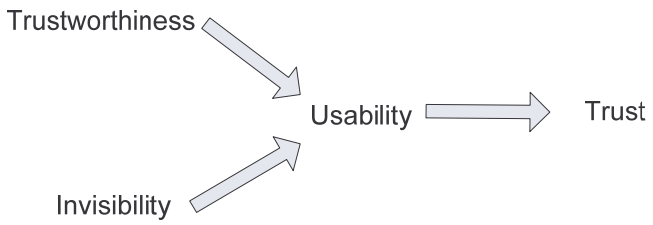

Figure 3: Trust Relationship

---

[8] http://www.biometricaccess.com/company/n_041102.htm
[9] http://news.com.com/Supermarket+Let+your+fingers+do+the+paying/2100-1029_3-5559074.html
[10] http://www.theregister.co.uk/2002/05/16/gummi_bears_defeat_fingerprint_sensors/
[11] Rousseau et al. Not so different after all: A cross-discipline view of trust. Academy of Management Review 32(3):393-404, 1998
[12] Schneider, ed. Trust in Cyberspace. Committee on Information Systems Trustworthiness, National Research Council, 1999.

Final Thoughts on Usability

Sometimes you might want to make a security system less usable in order to save the user from performing an action that would cause them harm. For example, consider Alice and Bob exchanging (private) encrypted emails using the PGP plug-in to Microsoft Outlook. Within this interface, it can be very easy to reply to an encrypted email thinking that your response will also be encrypted. In such a case, a "good" design may be one where the Alice has to confirm sending a cleartext message to Bob when his first message was sent encrypted.

Many of the issues surrounding computer security relate to cost and benefit. For example, how much will it "cost" me to use a particular security system, and how much will it benefit me? Is it really worth Alice's time to learn public key cryptography in order to send her message to Bob? Perhaps it is, but what is the cost to her of accidentally sending a private message in cleartext? Are there other, more convenient or usable methods for accomplishing this?

References

[1] Tim Keanini, "Top Ten Ways to Use Controls as the Language Between Business and Technology", SANS, 2003, http://www.sans.org/rr/audittech/Tim_Keanini_WP.pdf