Human Computer Interface Security (HCISEC)

References:

Whitten, Alma; Tygar, J. D. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*

Yee, Ka-Ping. *User Interaction Design for Secure Systems*

Presentation Overview

- This presentation introduces the notion of Human Computer Interface Security (HCISEC).
- It does so by covering a seminal HCISEC paper:

  Why Johnny Can't Encrypt:

  A Usability Evaluation of PGP 5.0
- And, as the above is somewhat dated, this presentation also includes a brief overview of the more recent paper:

  User Interface Design for Secure Systems

HCISEC

- The study of Human Computer Interface Security exists at the intersection of HCI and security research.
- HCISEC is an emerging discipline, generally regarded to be in its early stages.
- "Conventional wisdom holds that security and usability are two antagonistic goals in system design. There is an alternative view that holds that the expanded use of computers by the general public has turned the traditional tradeoff of security-for-usability on its head: unless designers create systems that are both secure *and* usable, they will build systems that are neither."

HCISEC Resources

- HCISEC Yahoo Group
- IEEE Security & Privacy Sept/Oct 2004 Special Issue
- HCISEC Bibliography:
  http://www.gaudior.net/alma/biblio.html
- HCISEC Workshops/Conferences:
  - CHI 2003 Workshop on HCI & Security Systems
    http://www.andrewpatrick.ca/CHI2003/HCISEC/
  - DIMACS Workshop on Usable Privacy and Security Software
    http://dimacs.rutgers.edu/Workshops/Tools/
  - **Symposium On Usable Privacy and Security (July 2005)**
    http://cups.cs.cmu.edu/soups/

Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0

References:

Whitten, Alma; Tygar, J. D. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*

WJCE Introduction

- Security mechanisms are only effective when used correctly.
- In *UNIX Security: Threats and Solutions* Matt Bishop has claimed that configuration errors are the probable cause of more than 90% of all computer security failures.
- Since average citizens are now increasingly encouraged to make use of networked computers for private transactions, the need to make security manageable for even untrained users has become critical.
- Why is there such a lack of good user interface design for security? Are general HCI principles adequate for security?

WJCE Introduction (Cont.)

- The paper offers a specific definition of usability for security and identifies several significant properties of security as a problem domain for HCI design.
- The paper proposes that the design priorities require to achieve usable security are significantly different from those of general HCI priorities.
- The paper presents the cognitive walkthrough technique and laboratory user tests as complementary methodologies well-suited for the analysis of HCISEC properties of a system.

Choice of PGP 5.0

- The authors chose PGP 5.0 for their case study because it served as:
  - A representative example of the best current user interface design for security,
  - An example of general HCI principles as applied to security software,
  - And as an example of security software designed with the expectation that it would be operated by a novice user.
- Although the paper uses PGP 5.0, the screenshots included in this presentation are from PGP 8.1. Though there are some differences, the more modern screenshots are not divergent enough as to constitute a problem.

Defining Usability for Security

Security software is usable if the people who are expected to use it:

1. are reliably made aware of the security tasks they need to perform;
2. are able to figure out how to successfully perform those tasks;
3. don't make dangerous errors; and
4. are sufficiently comfortable with the interface to continue using it.

Problematic Properties of Security

- **The Unmotivated User Property**

  Security is usually a secondary goal. People do not generally sit at their computers wanting to manage security; rather, they want to complete a task that may tangentially involve security.

- **The Abstraction Property**

  Computer security management often involves security policies which may be alien and unintuitive to many members of the general user population. This makes achieving simple abstractions difficult.

- **The Lack of Feedback Property**

  Providing good feedback for security management is a difficult problem. The state of a security configuration is usually complex, and attempts to summarize it are often inadequate.

Problematic Properties of Security (Cont.)

- **The Barn Door Property**

  Once a security secret has been left accidentally unprotected, even for a short time, there is no way to be sure that it has not already been read by an attacker. Because of this, a very high priority must be placed on making sure the user does not make potentially high-cost mistakes.

- **The Weakest Link Property**

  It is well known that the security of a networked computer is only as strong as its weakest component. This means that users need to be guided to attend to all aspects of their security, not left to proceed through random exploration as they might with a word processor or spreadsheet.

A Usability Standard for PGP

The authors derived the following question by applying the general definition of usability for security to PGP:

> *If an average user of email feels the need for privacy and authentication, and acquires PGP with that purpose in mind, will PGP's current design allow that person to realize what needs to be done, figure out how to do it, and avoid dangerous errors, without becoming so frustrated that he or she decides to give up on using PGP after all?*

A Usability Standard for PGP (Cont.)

In more detail, will the user, at a minimum:

- Understand that private is achieved by encryption, and figure out how to encrypt email and how to decrypt email received from other people;
- Understand that authentication is achieved through digital signatures, and figure out how to sign email and how to verify signatures on email from other people;
- Understand that in order to sign email and allow other people to send them encrypted email a key pair must be generated, and figure out how to do so;

A Usability Standard for PGP (Cont.)

- Understand that in order to allow other people to verify their signature and to send them encrypted email, they must publish their public key, and figure out some way to do so;
- Understand that in order to verify signatures on email from other people and send encrypted email to other people, they must acquire those people's public keys, and figure out some way to do so;
- Manage to avoid such dangerous errors as accidentally failing to encrypt, trusting the wrong public keys, failing to back up their private keys, and forgetting their pass phrases; and
- Be able to succeed at all of the above within a few hours of reasonably motivated effort.

Cognitive Walkthrough

- Cognitive walkthrough is a usability evaluation technique modeled after the software engineering practice of code walkthroughs. To perform a cognitive walkthrough the evaluators step though the use of the software as if they were novice users, looking for probably errors and areas of confusion.
- Although the analysis in the paper is best categorized as a cognitive walkthrough, it also incorporates aspects of heuristic evaluation. In heuristic evaluation the UI is evaluated against a specific list of high-priority usability principles.

Visual Metaphors

- The metaphor of *keys* is built into cryptographic terminology, and PGP's user interface relies heavily on graphical depictions of keys and locks.
- PGPTools*, shown above, provides a button for invoking the PGPKeys application, plus four buttons representing its four basic operations:
  - Encrypt
  - Sign
  - Encrypt & Sign
  - Decrypt/Verify

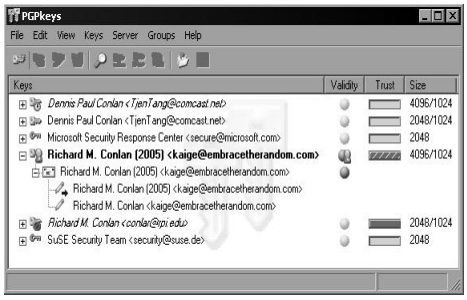
* Named PGPmail in PGP 8.1

Visual Metaphors (Continued)

- For a novice user these appear to be relatively straightforward visual metaphors for the operations to be performed.
- However:
  - the metaphors employed make no distinction between public and private keys;
  - the metaphor for signing in no way indicates that the signing operation involves the user's private key; and
  - The metaphor for decrypt/verify does not include the notion of verification as a step separate from decryption and therefore could easily lead to the user overlooking the meaningfulness of verification.

* Named PGPmail in PGP 8.1

PGPKeys



Different Keys Types

- Originally, PGP used the popular RSA algorithm for encryption and signing. As of PGP 5.0, it uses the DH/DSS algorithms instead. The RSA and DH/DSS algorithms use different types of keys.
- This leads to a few problems:
  - Old versions of PGP cannot decrypt using DH/DSS keys;
  - If a file is encrypted to both DH/DSS keys and RSA keys, recipients with old versions of PGP cannot decrypt it even if they have the corresponding RSA keys.

Different Keys Types (Cont.)

- PGP 5.0 alerts users to this compatibility issue in two ways.
  - It uses different icons to depict the different key types. It uses a gray key with an old fashioned shape for RSA keys, and a brass key with a more modern shape for DH/DSS keys.
  - When users attempt to encrypt documents using mixed key types a warning message is displayed.
- Unfortunately, information about the meanings of the icons is difficult to find.
- Explanation of why the different key types are significant, aside from the warning message, is given only in the 132-page manual.
- Finally, PGP breaks its metaphor when the user is selecting recipients for encryption – the stage at which the key difference arguably matters most.

Selecting Recipients for Encryption

Key Server

- Key servers are publicly accessible databases in which anyone can publish a public key joined to a name. PGP offers three key server operations under the Keys pull-down menu.
- This is problematic, because nothing about the top-level menus calls attention to the existence of the key server.
- PGP 5.0 also keeps no record of key server accesses. There is nothing to show whether a key has been sent to a key server, or when a key was fetched or last updated.
- The PGP key revocation operation does not automatically send the revocation certificate to the server. While this might be good in-and-of-itself, it also does not notify the user this must be done to publicly revoke the key.

Irreversible Actions

- Accidentally Deleting the Private Key
  - There is no warning about the potential consequences.
- Accidentally Publicizing a Key
  - Users may not realize that keys added to the database are added permanently. This is especially bothersome since notions of key revocation and expiration are relatively sophisticated.
- Accidentally Revoking a Key
  - Even if the user does not distribute the revocation, they cannot unrevoke the key in their key ring.

Irreversible Actions (Cont.)

- Forgetting the Pass Phrase
  - PGP suggests that the user make a backup revocation certificate, but aside from the concept being somewhat advanced, it is hard to do even for those that understand it.
- Failing to Back Up the Key Rings
  - The user is not prompted to backup their key ring until the exit PGP. Also, the default backup location is in the user's PGP folder.

Information Overload

- There is a lot of information displayed in the PGPKeys interface.
- A notice user is most likely interested in encrypting their e-mail to maintain confidentiality and is not overly concerned with the authentication aspects of PGP.
- The authors propose that more security would be realized, because the product would be simpler to understand, if the interface was pruned down even though this would result in the loss of some security critical information.
- They also note that the information should still be available for access by experts, who are the most likely to understand the information anyways.

User Test

- The test was conducted using the PGP 5.0 plug-in for Eudora.
- Test participants were tutored on the use of Eudora, but not PGP or the PGP plug-in.
- The testers were allowed to answer any Eudora related questions during the test.
- The testers could send minimally supportive answers when participants seemed stuck and asked "team members" within the test for help.

Test Design

- Participants were told that they were assisting in a political campaign and that they had to send their candidate's itinerary to five fictitious campaign members in a signed and encrypted e-mail.
- In order to complete this a participant had to:
  - Generate a key-pair
  - Get the team members' public keys
  - Make their generated public key available
  - Sign the e-mail using their private key
  - Encrypt the e-mail under each of the team members' public keys
  - Send the result

Test Design (Cont.)

- The test was further complicated by the fact that one of the team members had an RSA key-pair while the others all had Diffie-Hellman/DSS keys. This caused a problem if they just send the e-mail encrypted under all five keys.
- If a participant completed the initial task they were sent an encrypted e-mail with further instructions. These instructions suggested they back up their key ring and make sure to generate a revocation certificate.
- The test lasted 90 minutes.

Participants

- The test was run with twelve different participants, all of whom were experienced users of email, and none of whom could describe the difference between public and private key cryptography prior to the test session.
- The participants all had attended at least some college, and some had graduate degrees.
- Their ages ranged from 20 to 49, and their professions were diversely distributed.
- More detailed demographics are available.

Results

- Only one-third of the participants were able to use PGP 5.0 to correctly sign and encrypt an e-mail message.
- One-quarter of the participants accidentally exposed the secret they were meant to protect.
- Nobody successfully created a revocation certificate (though only 3 participants even made it this far).
- The paper offers a lot more detail than this about the specific steps that users failed at and the types of confusion they encountered.

Conclusions

- The results of the case study supported the hypothesis that standard HCI design is not sufficient to make computer security usable for people who are not already knowledgeable in that area.
- PGP's UI fails to enable effective security where it is not designed in accordance with the definition of usability for security.
- Standard usability evaluation methods, simplistically applied, may treat security functions as if they were the primary rather than secondary goals for the user, leading to faulty conclusions.

User Interaction Design for Secure Systems

References:

Yee, Ka-Ping. *User Interaction Design for Secure Systems*

Security Design Principles

- **Path of Least Resistance**
  The most natural way to do any task should also be the most secure way.
- **Appropriate Boundaries**
  The interface should expose, and the system should enforce, distinctions between objects and between actions along boundaries that matter to the user.
- **Explicit Authorization**
  A user's authorities must only be provided to other actors as a result of an explicit user action that is understood to imply granting.

Security Design Principles (Cont.)

- **Visibility**
  The interface should allow the user to easily review any active actors and authority relationships that would affect security-relevant decisions.
- **Revocability**
  The interface should allow the user to easily revoke authorities that the user has granted, wherever revocation is possible.
- **Expected Ability**
  The interface must not give the user the impression that it is possible to do something that cannot actually be done.

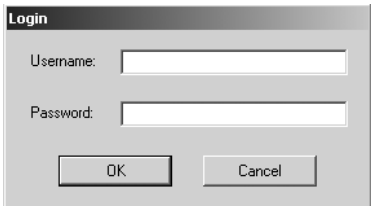
Security Design Principles (Cont.)

- **Trusted Path**
  The interface must provide an unspoofable and faithful communication channel between the user and any entity trusted to manipulate authorities on the user's behalf.
- **Identifiability**
  The interface should enforce that distinct objects and distinct actions have unspoofably identifiable and distinguishable representations.

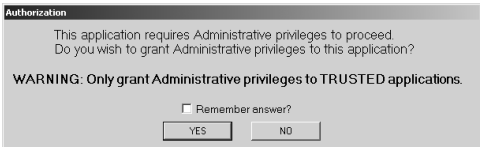
Security Design Principles (Cont.)

- **Expressiveness**
  The interface should provide enough expressive power (a) to describe a safe security policy without undue difficulty; and (b) to allow users to express security policies in terms that fit their goals.
- **Clarity**
  The effect of any security-relevant action must be clearly apparent to the user before the action is taken.

Login Screen

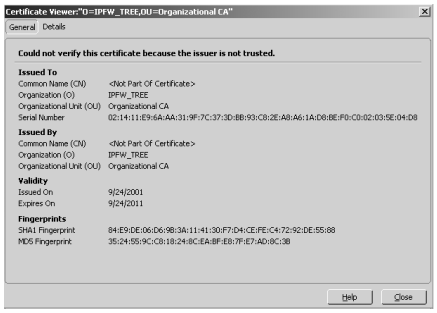


Privilege Authorization



Firefox Install Certificate Prompt

Firefox Certificate Details



IE Install Certificate Prompt

IE Certificate Details

Questions?