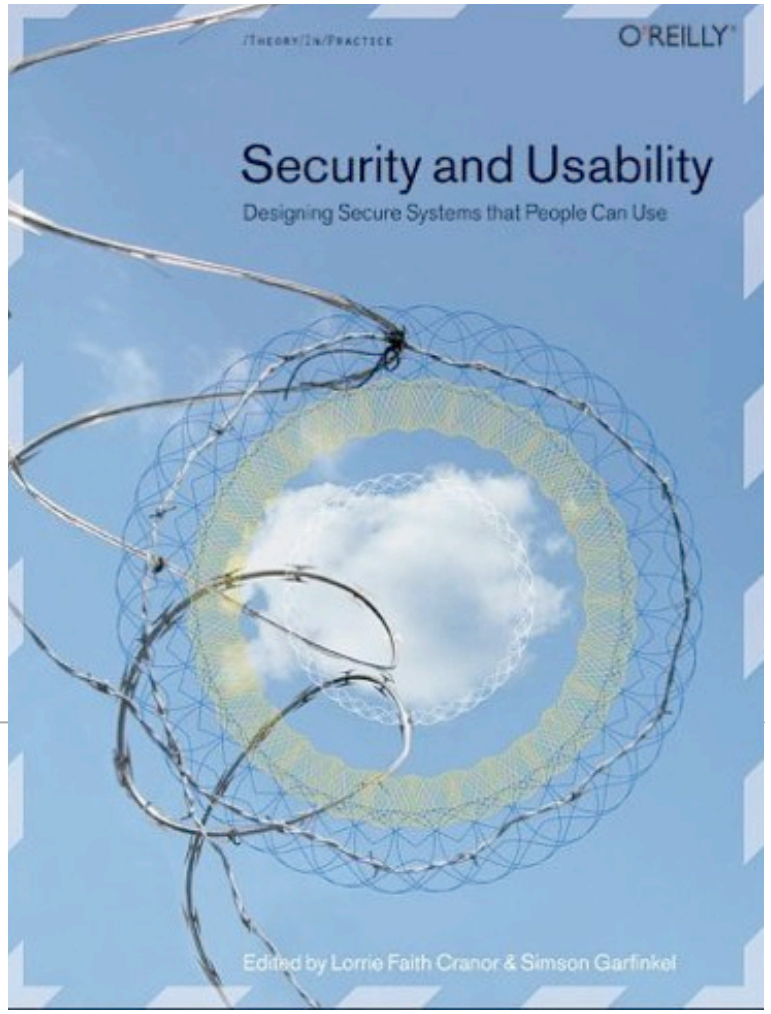
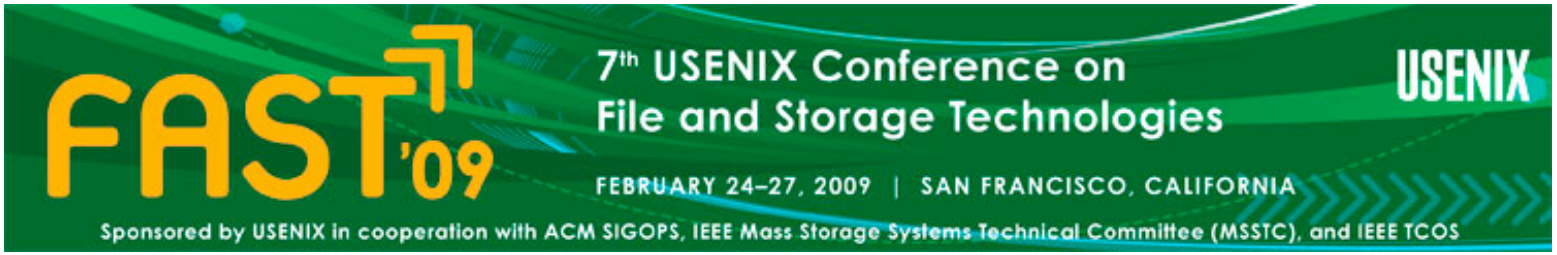
Security and Usability: What do we know?



Simson L. Garfinkel, Ph.D.
http://www.simson.net/

Tuesday, February 24, 2009



A bit about me

Tech Journalist: 1985—2002

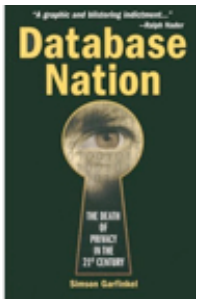
Entrepreneur: 1988—2002

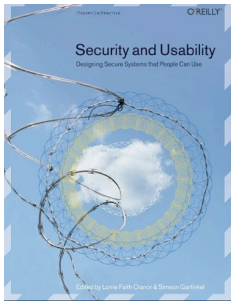
MIT EECS 2002—2005

Harvard SEAS, 2005—2008

Associate Professor, 2006—
Naval Postgraduate School,

Associate, 2008—
School of Engineering and Applied Sciences,
Harvard University





Harvard
School of Engineering
and Applied Sciences

NPS
PRAESTANTIA PER SCIENTIAM

"The views expressed in this presentation do not necessarily reflect those of the Department of Defense or the US Government."

A bit about NPS.

Located in: Monterey, CA

Campus Size: 627 acres

Student: 1500

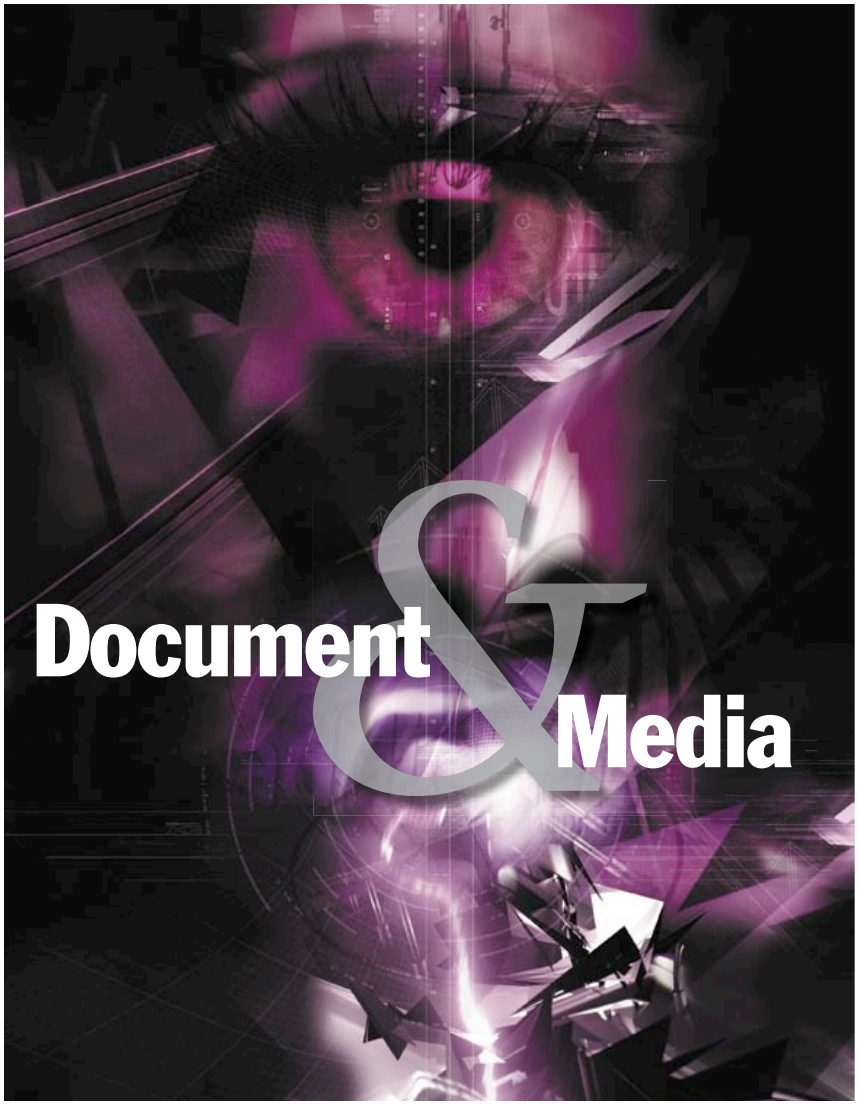
- US Military (All 5 services)
- US Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)

Schools:

- Business & Public Policy
- Engineering & Applied Sciences
- Operational & Information Sciences
- International Graduate Studies

ASK ME ABOUT OUR MS AND PHD PROGRAM!

My current research:
Automated Document & Media Exploitation



A computer used by Al Qaeda ends up in the hands of a *Wall Street Journal* reporter. A laptop from Iran is discovered that contains details of that country's nuclear weapons program. Photographs and videos are downloaded from terrorist Web sites.

As evidenced by these and countless other cases, digital documents and storage devices hold the key to many ongoing military and criminal investigations. The most straightforward approach to using these media and documents is to explore them with ordinary tools—open the word files with Microsoft Word, view the Web pages with Internet Explorer, and so on.

Although this straightforward approach is easy to understand, it can miss a lot. Deleted and invisible files can be made visible using basic forensic tools. Programs called *carvers* can locate information that isn't even a complete file and turn it into a form that can be readily processed. Detailed examination of e-mail headers and log files can reveal where a computer was used and other computers with which it came into contact. Linguistic tools can discover multiple documents that refer to the same individuals, even though names in the different documents have different spellings and are in different human languages. Data-mining techniques such as cross-drive analysis can reconstruct social networks—automatically determining, for example, if the computer's previous user was in contact with known terrorists. This sort of advanced analysis is the stuff of DOMEX, the little-known intelligence practice of document and media exploitation.

The U.S. intelligence community defines DOMEX as "the processing, translation, analysis, and dissemination

Exploitation

SIMSON L. GARFINKEL, PH.D.

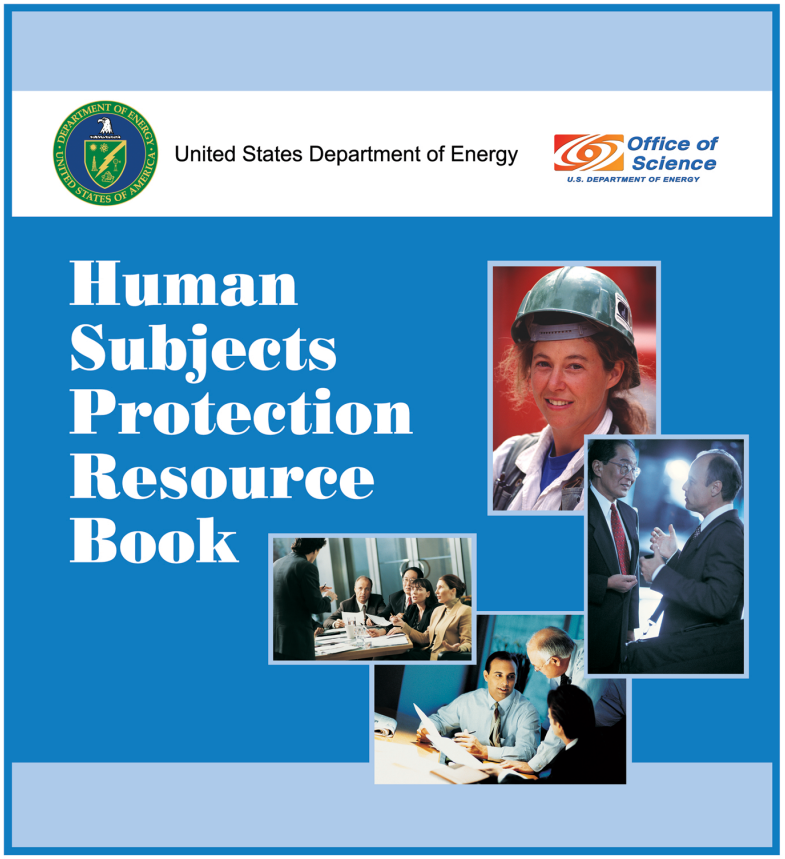
The DOMEX challenge is to turn

digital bits into actionable intelligence.

I am also active in HCI-SEC and IRB issues.

Symposium On Usable Privacy and Security
July 15-17, 2009
Mountain View, CA

http://cups.cs.cmu.edu/soups/2009/



Goals for this tutorial.

This tutorial assumes you know more about security than usability.

- Understand how usability is relevant to security.
- Basics of usability design.
- Understand why HCI-SEC is hard.
- Review research on HCI-SEC
- Understand what works.

Outline for this morning's tutorial.

9:00 — 10:30 **What is HCI-SEC and why is it so hard?**

- Understanding Security, Usability, and HCI-SEC
- What makes HCI-SEC different?
- 10 years of HCI-SEC research

10:30 — 11:00 ***Coffee!***

11:00 — 12:30 **HCI-SEC in practice**

- Principles for aligning security and usability
- Specific approaches
- Where to go for more information.

What is Usable Security...
...and why is it so hard?

Security
Usability
Usable Security

What is Usable Security...
...and why is it so hard?

Security
Usability
Usable Security

What is computer security?

Traditionally, computer security was:

- Confidentiality
- Integrity
- Availability

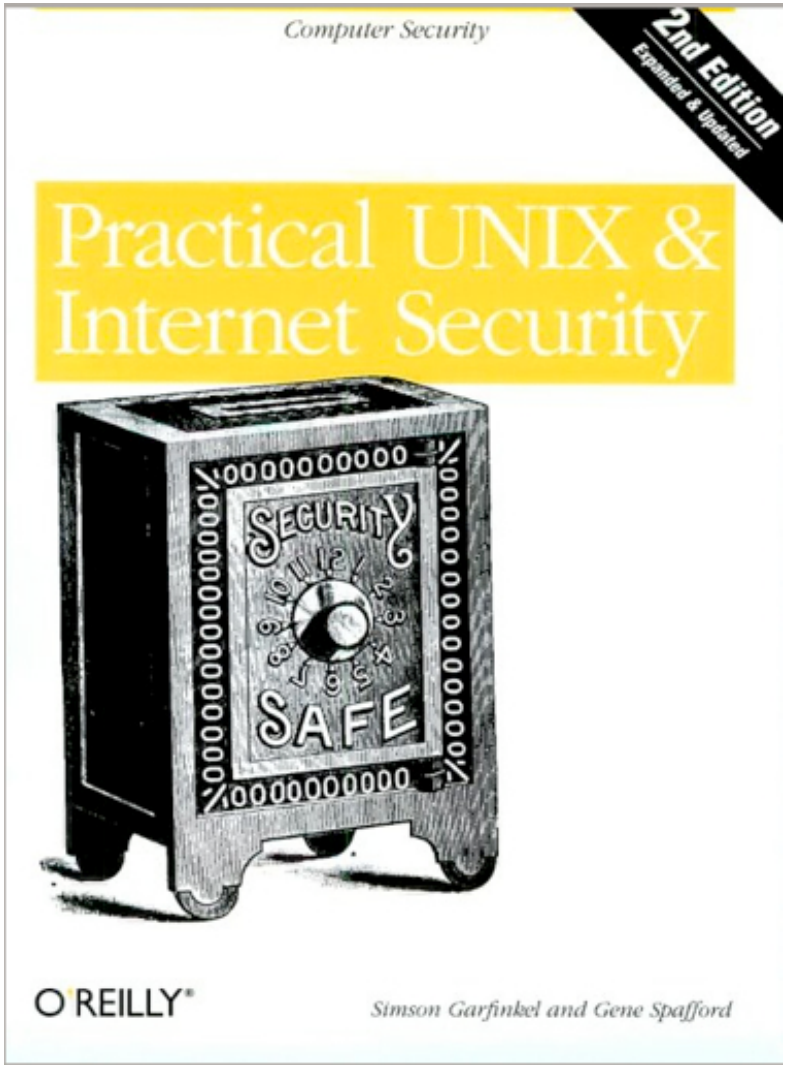
Then we added:

- Non-repudiation
- Risk management
- Incident Response
- Disaster planning

What is security?

"A computer is secure if you can depend on it and its software to behave as you expect. (PUS, '91)

Systems are secure if they are *trustworthy.*



What is computer security?

Traditionally, computer security was:

- Confidentiality
- Integrity
- Availability

Then we added:

- Non-repudiation
- Risk management
- Incident Response
- Disaster planning
- ***Usability***

Usability & Security is typically called HCI-SEC.

Human
Computer
Interaction

&

SECurity

But in fact,

- "*usability*" *has always been a security requirement.*
- "*security*" *has always been a usability requirement.*

Why the recent interest in HCI-SEC?

Security is getting worse, not better.

- Increased number of security incidents
- Increased *variety* of security problems

Why?

- Demilitarization of cyberspace.
- End users are system managers.
- Increased connectivity.
- More financial incentives (home banking, phishing, etc.)
- Proliferation of passwords, pins, challenge questions, etc.

How do you build a trustworthy system?

In 1975 Jerome Saltzer and Michael Schroeder formulated three design principles for building secure systems:

1. Economy of mechanism.
2. Fail-safe defaults.
3. Complete mediation.
4. Open design.
5. Separation of privilege.
6. Least privilege.
7. Least common mechanism.
8. Psychological acceptability.

Two of these principles involve usability.

Fail-safe defaults

- “Base access decisions on permission rather than exclusion.”
- Make the system secure by default.
- [Implies control of user-initiated configuration changes.]

Psychological acceptability

- “It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.”
- If the security system is not easy to use, people will circumvent it.

So what is usability?

Usability: "I know it when I see it."

Jakob Nielsen identified five elements in a usable interface:

1. **satisfaction**: Interfaces we enjoy using
2. **efficiency**: Interfaces we are fast at using
3. **learnability**: Interfaces we can use without asking for help
4. **memorability**: Interfaces we can use after time
5. **few and non-catastrophic errors**:
   mistakes are rare and easily recovered

(Notice that "usability" is different than from "accessibility.")

"Catastrophic errors" are about security.

Many security problems seem difficult because ***there is no way to recover*** from the error.

Information Disclosure

- You can't **make the adversary forget** a secret once it has been revealed.
- You can't **stop somebody from spending $$** if your credit card is revealed.

Information Destruction

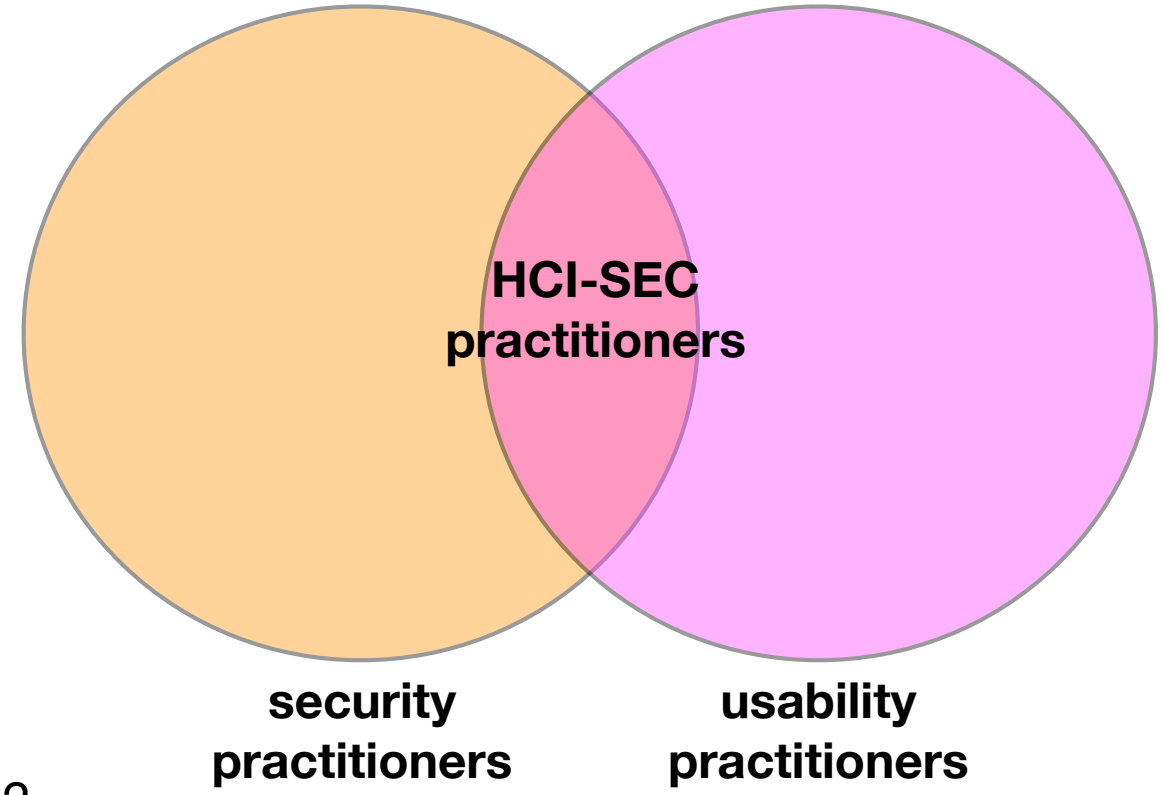
- You can't get data back once it has been erased.

Why is HCI-SEC hard?

User interfaces are hard to design.

Security is hard to understand.

There just aren't that many people with both skills.



Is something else going on as well?

HCI-SEC: Usability in the face of an adversary

The adversary can exploit usability features.

The adversary can exploit usability problems.

The adversary can masquerade as tech support.

Security features make systems harder to use—so people disable them.

Adversaries adapt faster than legitimate users.

People have a hard time distinguishing attacks from system errors.

Security professionals traditionally blamed users (and administrators!)

Users were expected to:

- Use passwords that were too difficult to guess, but could be remembered without writing them down.

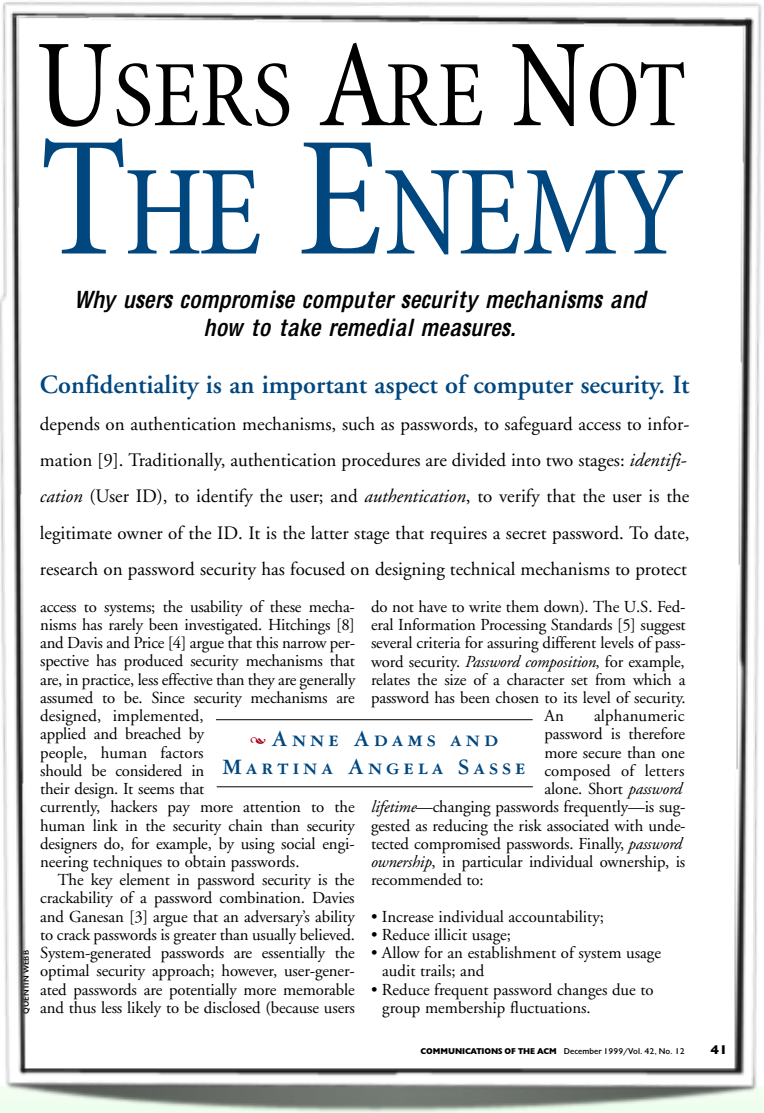
Administrators were expected to:

- Maintain system & apply patches

Developers were expected to:

- Securely code.
- Understand crypto.

Like blaming plane crashes on "pilot error."



USERS ARE NOT THE ENEMY

Why users compromise computer security mechanisms and how to take remedial measures.

Confidentiality is an important aspect of computer security. It depends on authentication mechanisms, such as passwords, to safeguard access to information [9]. Traditionally, authentication procedures are divided into two stages: *identification* (User ID), to identify the user; and *authentication*, to verify that the user is the legitimate owner of the ID. It is the latter stage that requires a secret password. To date, research on password security has focused on designing technical mechanisms to protect access to systems; the usability of these mechanisms has rarely been investigated. Hitchings [8] and Davis and Price [4] argue that this narrow perspective has produced security mechanisms that are, in practice, less effective than they are generally assumed to be. Since security mechanisms are designed, implemented, applied and breached by people, human factors should be considered in their design. It seems that currently, hackers pay more attention to the human link in the security chain than security designers do, for example, by using social engineering techniques to obtain passwords.

The key element in password security is the crackability of a password combination. Davies and Ganesan [3] argue that an adversary's ability to crack passwords is greater than usually believed. System-generated passwords are essentially the optimal security approach; however, user-generated passwords are potentially more memorable and thus less likely to be disclosed (because users do not have to write them down). The U.S. Federal Information Processing Standards [5] suggest several criteria for assuring different levels of password security. *Password composition*, for example, relates the size of a character set from which a password has been chosen to its level of security. An alphanumeric password is therefore more secure than one composed of letters alone. Short *password lifetime*—changing passwords frequently—is suggested as reducing the risk associated with undetected compromised passwords. Finally, *password ownership*, in particular individual ownership, is recommended to:

- Increase individual accountability;
- Reduce illicit usage;
- Allow for an establishment of system usage audit trails; and
- Reduce frequent password changes due to group membership fluctuations.

ANNE ADAMS AND MARTINA ANGELA SASSE

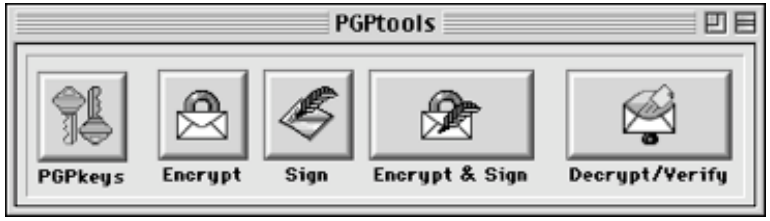
QUENTIN WEBB

COMMUNICATIONS OF THE ACM December 1999/Vol. 42, No. 12 41

A. Adams & M. A. Sasse (1999): Users Are Not The Enemy: Why users compromise security mechanisms and how to take remedial measures, in: Communications of the ACM, 42 (12), pp. 40-46 December 1999

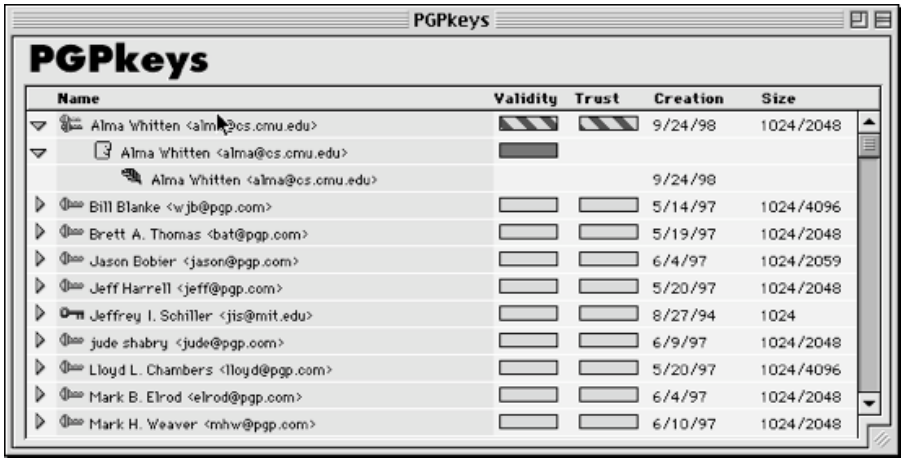
“Why can’t Johnny Encrypt?” (Whitten & Tygar, 1999 Usenix Security)

A usability study of PGP 5.0.



PGP 5.0 was a:

- Mature program.
- “Easy-to-use” GUI
- Integrated with Eudora on the Mac.



Whitten & Tygar performed:

- Cognitive walk-through of program.
- Laboratory study with 12 users.

Conclusion: Designing interfaces for “security software” is ***inherently more difficult*** than designing ordinary software.

Why is HCI-SEC so hard?
Whitten and Tygar proposed five reasons.

1. The Secondary Goal Property

- "People do not generally sit down at their computers wanting to manage their security; rather, they want to send mail, browse web pages, or download software."

2. The Abstraction Property

- "Security policies are usually phrased as abstract rules that are easily understood by programmers but "alien and unintuitive to many members of the wider user population."

Why is HCI-SEC so hard?
Whitten and Tygar proposed five reasons.

3. The Hidden Failure Property

- “It is difficult to provide good feedback for security management and configuration because configurations are complex and not easy to summarize”

4. The Barn Door Property

- “Once a 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.”

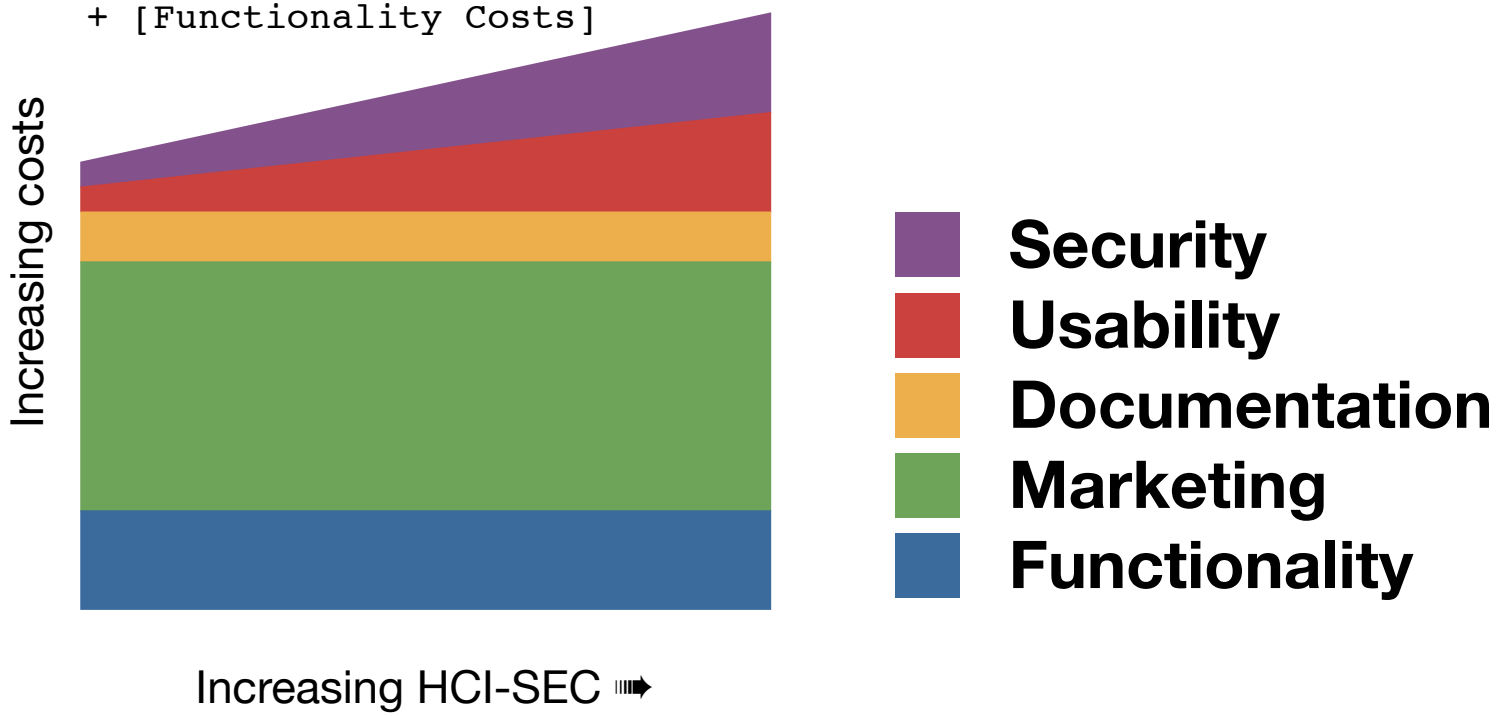
5. The Weakest Link Property

- “The security of a networked computer is like a chain: it is only as strong as the weakest component. If a cracker can exploit a single error, the game is up.”

Security and Usability are often presented as having conflicting goals.

HCI and SEC conflict when addressed separately:

```
Total cost = [Security Costs] + [Usability Costs]
           + [Documentation Costs] + [Marketing Costs]
           + [Functionality Costs]
```



But without usability, systems cannot be operated securely.

Usability is necessary for "effective security."

Security & Usability practitioners have adopted similar approaches.

"Design in from the beginning:"

- "We agreed that security was not optional, and that it needed to be designed in from the beginning."

  "Report of the IAB Security Architecture Workshop," RFC 2316, S. Bellovin, April '98

- "Usability must be designed in from the beginning."

  "The Importance of Designing Usable Systems," Susan M. Dray, Interactions Magazine*, January 1995 (Volume 2, issue 1), pp. 17-20*

"Iterative Design:"

- "Security is a process, not a product"

  Bruce Schneier, Crypto-Gram, May 15, 2000

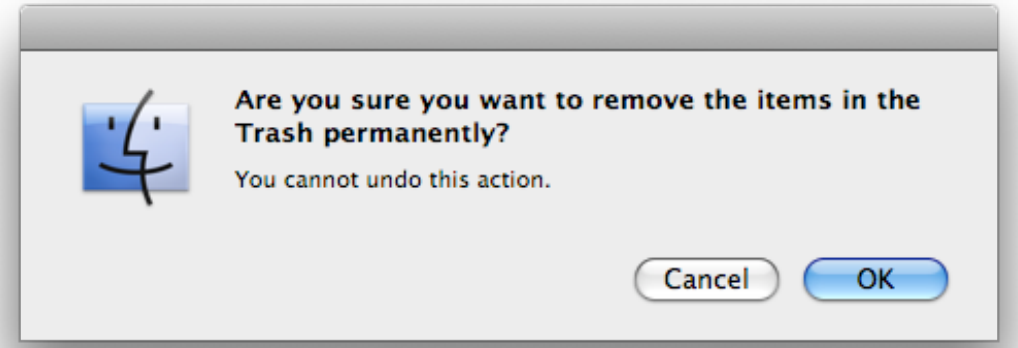
- "Use an iterative design process"

  John Gould and Clayton Lewis, 1983

  "25 Years in Usability," Jakob Nielsen, April 21, 2008

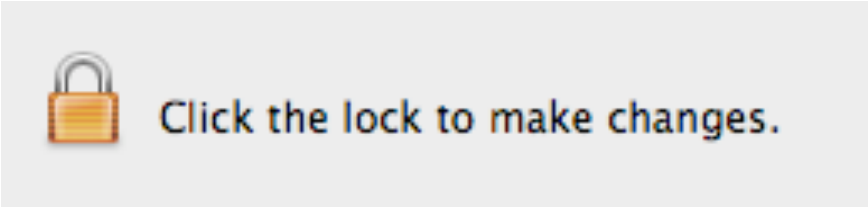
HCI-SEC approach #1:
Make catastrophic errors hard.

Confirmation boxes:



Simplify configuration and management:

- Remove user options



This is the traditional approach to HCI-SEC.

HCI-SEC approach #2:
Make catastrophic events reversible.

~~You can't **make the adversary forget** a secret once it has been revealed.~~

Make the secrets not matter:

- Anti-fraud systems
- Change keys; Get a new credit card
- DRM systems for document control

~~You can't get data back once it has been erased~~

- Backups
- Multiple Servers

Time Machine

This is the new, enlightened approach to HCI-SEC.

The big problem with much of this work

Which of these is “Security Software?”

- ❏ PGP 5.0 for Macintosh
- ❏ Microsoft Word 2007 for Windows
- ❏ Internet Explorer 7.0
- ❏ Ubuntu Linux 8.10
- ❏ Apache 2.0?

Answer: all software is security software.

HCI-SEC research areas

Most HCI-SEC research has been in one of these areas:

- Authentication
- Anti-phishing
- Secure pairing of wireless devices.

What's been missing:

- Redesigning underlying operating system
- Physical Authentication

Recent Publications:
“What Instills Trust? A Qualitative Study of Phishing”

Markus Jakobsson, Alex Tsow, Ankur Shah, Eli Blevis, and Youn-Kyung Lim
Usable Security 2007

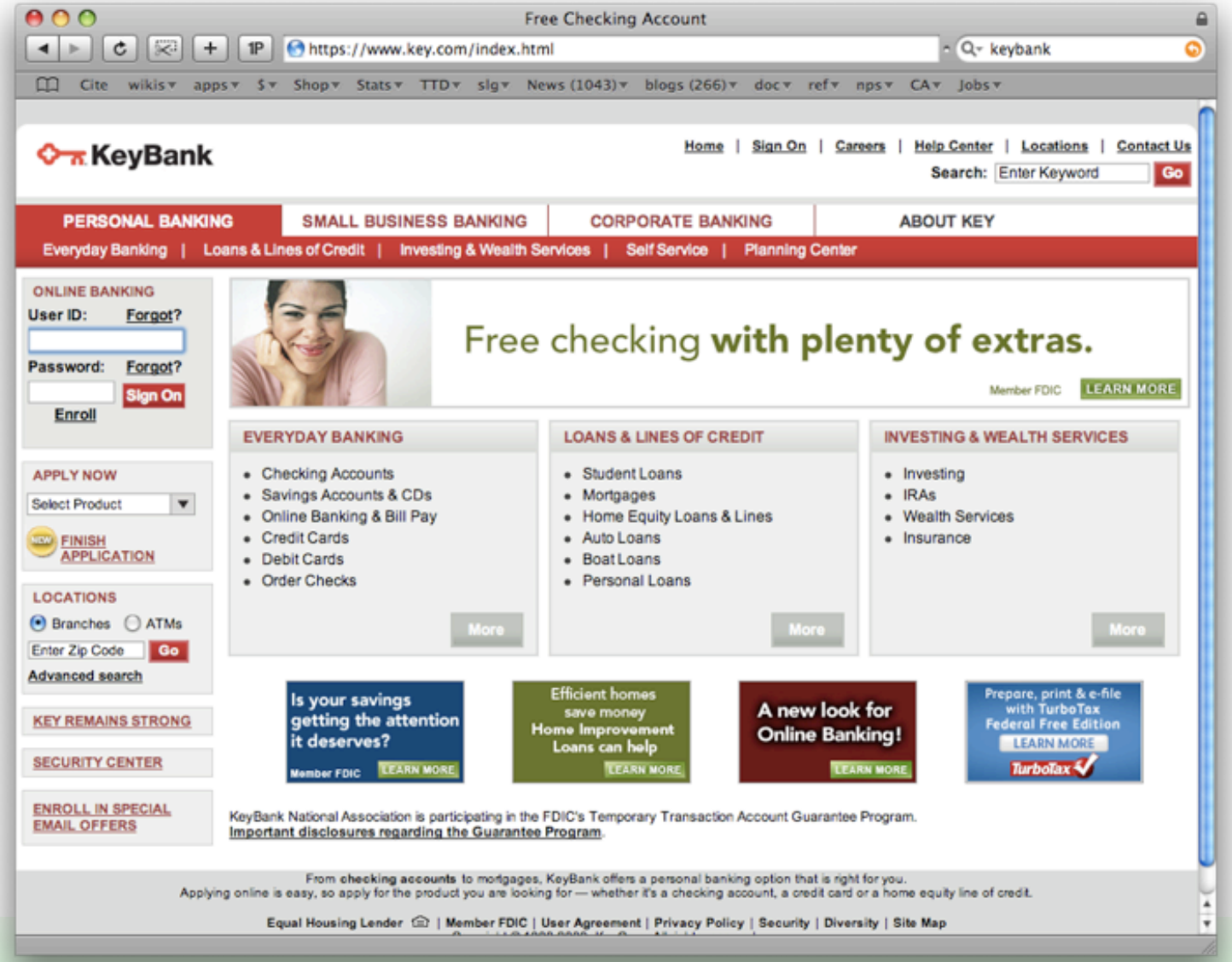
- 17 subjects from 18 to 60 participated in “think out loud” study.
- Shown legitimate and “phishy” emails, web pages, & phone calls.

Conclusions:

1. Spelling and design matter.

- phising email dismissed based on spelling.
- Emails signed by a person (Jim Smith) more credible than those signed by roles (e.g. “Account manager, Paypal.”)
- Security awareness more credible when confined to portion of web page.

Is this a Phishing Site?



Jakobsson et. al additional findings

2. Too much emphasis on security can backfire.

3. People look at URLs.

- Phishy: http://65.33.213/ http://www-chase.com/
- Not phishy: http://www.chase-alerts.com/

4. Third party endorsements depend on brand recognition.

5. People judge relevance before authenticity.

- Verisign good; TRUST-e bad

6. Pesonalization creates trust

- *Any personalization:* ***4246-XXXX-XXXX-XXXX; ZIP Code; '4432***

7. Phone calls are not phishy.

- Not even voice response systems
- Not even when phone number is in email.

8. Padlocks in chrome create trust—sometimes.

Other recent titles from literature:

"An Evaluation of Extended Validation and Picture-in Picture Phishing Attacks."

- EV certificates help if users are told what they mean.

"Empirical studies on software notices to inform policy makers and usability designers."

- Surprisingly, some people actually read EULAs. (but not many)
- Most EULAs are too long to be useful.

"Low-cost Manufacturing, Usability, and Security: An Analysis of Bluetooth Simple Pairing and Wi-Fi Protected Setup."

- In-band is too susceptible to attack.
- Industry should push for a single out-of-band channel for setup.
- Decoy devices or scanners to detect attackers can improve security.

Usability Design in 5 minutes

User Interfaces are hard to design

You are not the user!

- Most software engineering is about communicating with other programmers
- UI is about communicating with users

The user is almost always right

- Consistent problems are the system's fault

... but the user is not always right

- Users can change through training and experience.
- User's aren't designers

Who is the designer? Who is the user?

In most cases, programmers design for themselves...

- This can be a *good thing:*
  - ✓ *Developer tools*
  - ✓ *Interfaces for "power users."*
  - ✓ *Programmers that have an innate sense of design.*
- This can be a bad thing:
  - ✓ Small projects that can't afford a proper designer.
  - ✓ Legacy systems designed for experts, migrating to larger user base.

In most cases, security experts design for themselves...

- The security issues frequently aren't understood or known.
- This is *invariably a bad thing*.

Jeff Johnson's usability design principles are relevant to HCI-SEC.

#1 - Focus on the users and their tasks, not the technology

#2 - Consider function first, presentation later.

#3 - Conform to the users' view of the task

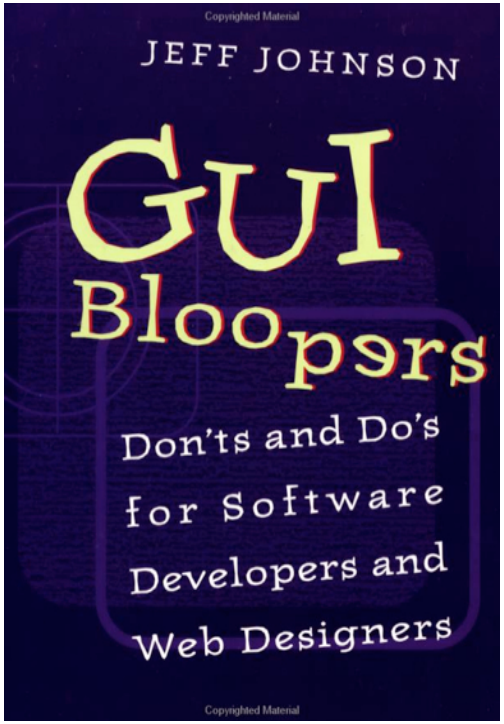
#4 - Don't complicate the user's task

#5 - Promote learning Inside the Interface

#6 - Deliver information, not just data

#7 - Design for responsiveness

#8 - Try it out on users, then fix it!



Principle #1:
Focus on the users and their tasks, not the technology

For whom is this product being designed?

What is the product for?

What problems do the users have now?

What are the skills and knowledge of the users?

How do users conceptualize and work with their data?

Principle #2:
Consider function first, presentation later.

Does not mean "worry about the user interface later!"

1. Develop a conceptual model
2. Keep the model as simple as possible, but no simpler
3. Develop a lexicon.

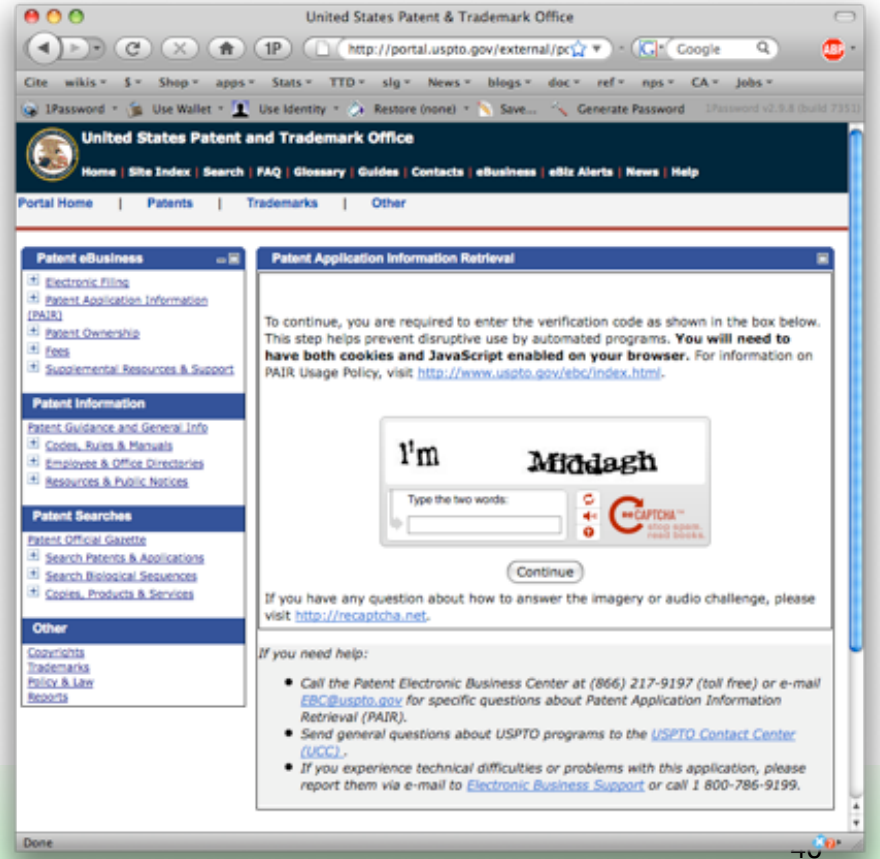
One of the most significant problems in HCI-SEC is the lack of a consistent lexicon!

Principle #4:
Don’t complicate the user’s task

Common tasks should be easy.

This is especially a problem with *passwords* and *captchas:*

- *The purpose of passwords is to complicate tasks!*
- CAPTCHAS exist because computer time is more valuable than human time!



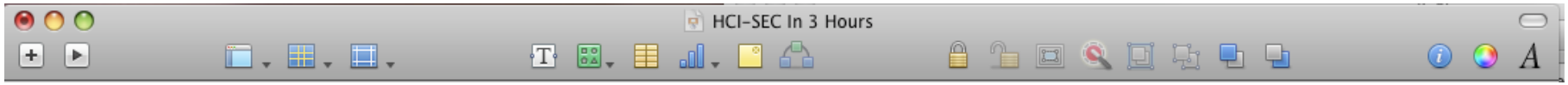
Principle #5:
Promote learning inside the interface

The user wants to solve a problem, not learn how to use your program!

- Think “outside-in,” not “inside-out”

Be careful of ambiguity

Icons do not naturally make sense...



Be consistent so there is something to learn!

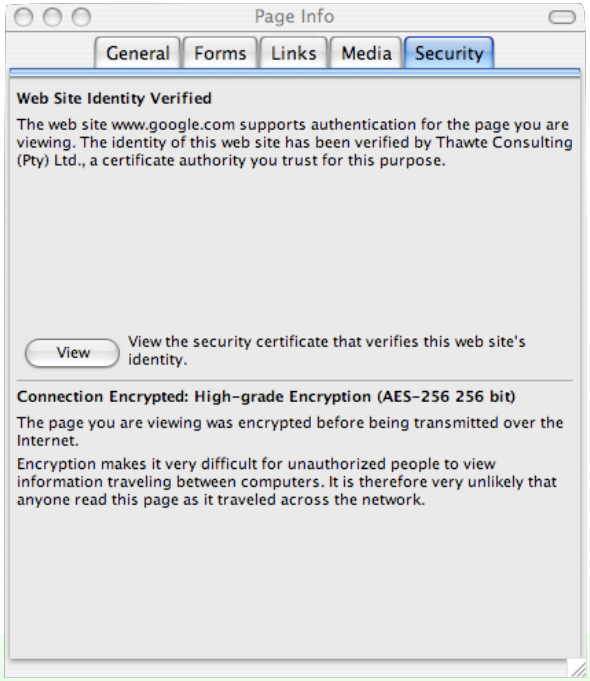
Principle #6: Deliver information, not just data

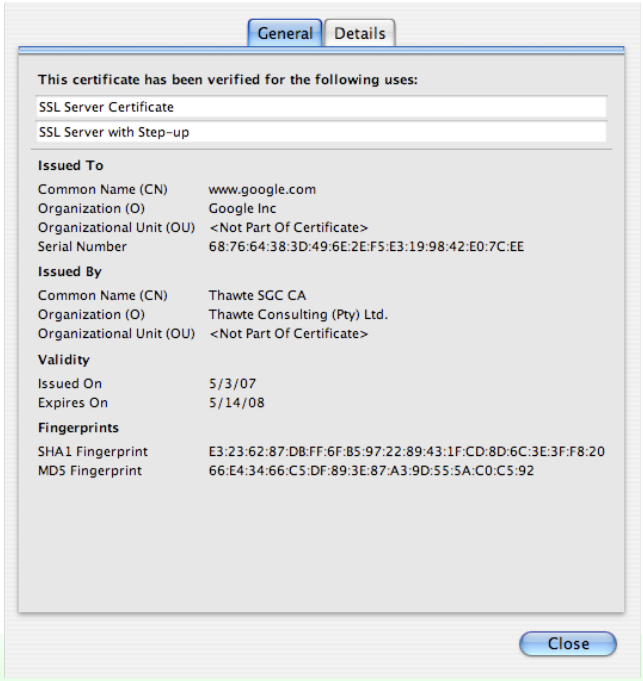
Design displays carefully.

The screen belongs to the user.

Preserve display inertia.

Certificate dialogues do not follow this principle at all...





HCI-SEC is about more than user interfaces.

Underlying system design:

- What functionality is provided?
- Where is the functionality provided?
  - ✓ Client? Server?

User expectations and education:

- Marketing materials
- Documentation

Economics and user incentives:

- What encourages the user to act in a secure manner?
- What is the price/penalty for unsecure behavior?

Questions or comments?

HCI-SEC:
10 years of research

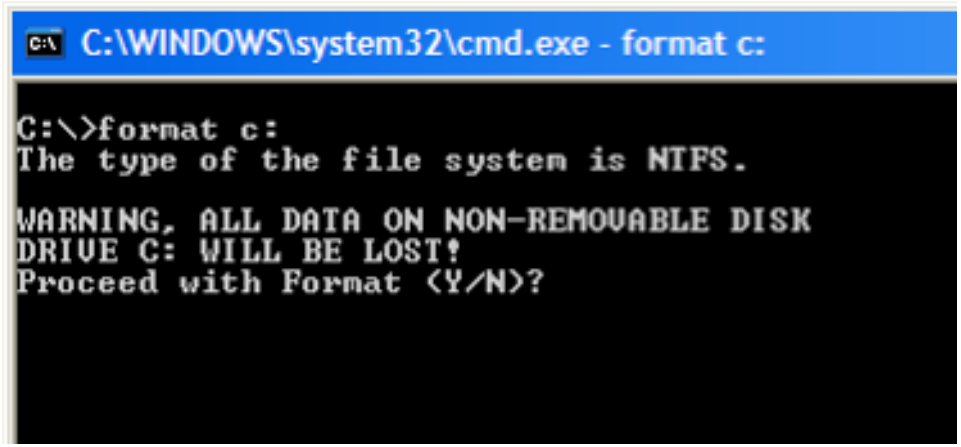
Many security systems fail invisibly.

Which of these is encrypted?

```
U2FsdGVkX1/XlRf6Gt1czLSd9VgyKQatH76f4VFoF5w=

VGhpcyBpcyBhIHRlc3QK==
```

Which of these actually erases data?

```
format c:      (Windows 95)

format c:      (Windows XP)

format c:      (Vista)
```



The lack of *transparency* makes it hard to understand online privacy & security.

Hidden Information at the Server:

- Log files
- Third-party Image Servers
- Web Bugs

Hidden Information at the Client:

- Cookies
- Browser History
- Browser Cache

DNS is opaque to most users:

- Many DNS names can map to one IP address
- Many IP addresses can map to one DNS name
- No relationship between a DNS name and a company

File-sharing programs encourage reckless sharing.

"A study of Kazaa P2P file Sharing," Good & Krekelberg, 2003

- Lab study of users.
- Searched Kazaa for potentially confidential files.
- Made "Credit Cards.xls" file available; people tried to download it!

Kazaa study main conclusions: Users can't believe programmers are so dumb.

Easier to share entire hard drive than a specific directory.

Users confused by specific program behavior:

- Sharing a directory automatically shares sub-directories.
- Sharing a directory with "movies" shared all file types in directory.

"You mean it shares all files?"

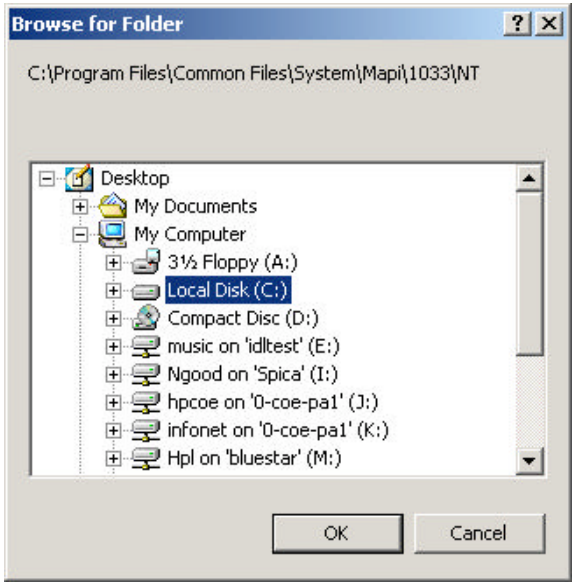


Figure 3 Browsing and selecting interface for the Shared/Download Folder. Note that the interface says browse for folder, and does not mention that the folders will be recursively searched for files.

"Why Phishing Works" (Dhamija, Tygar & Hearst, CHI 2006)

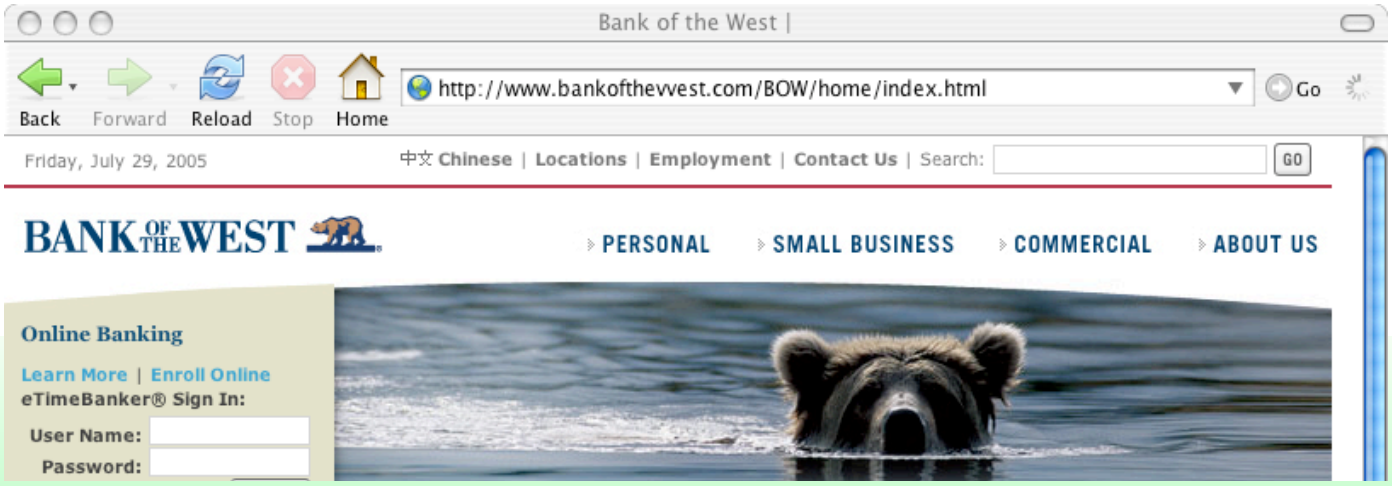
Hypothesis: specific strategies are effective at deceiving users.

Researchers:

- Analyzed a large number of phishing attacks.
- Performed laboratory usability study with 22 participants and 20 web sites.

Key Results:

- 23% of participants did not look at browser-based cues
- Users made incorrect choices 40% of the time.

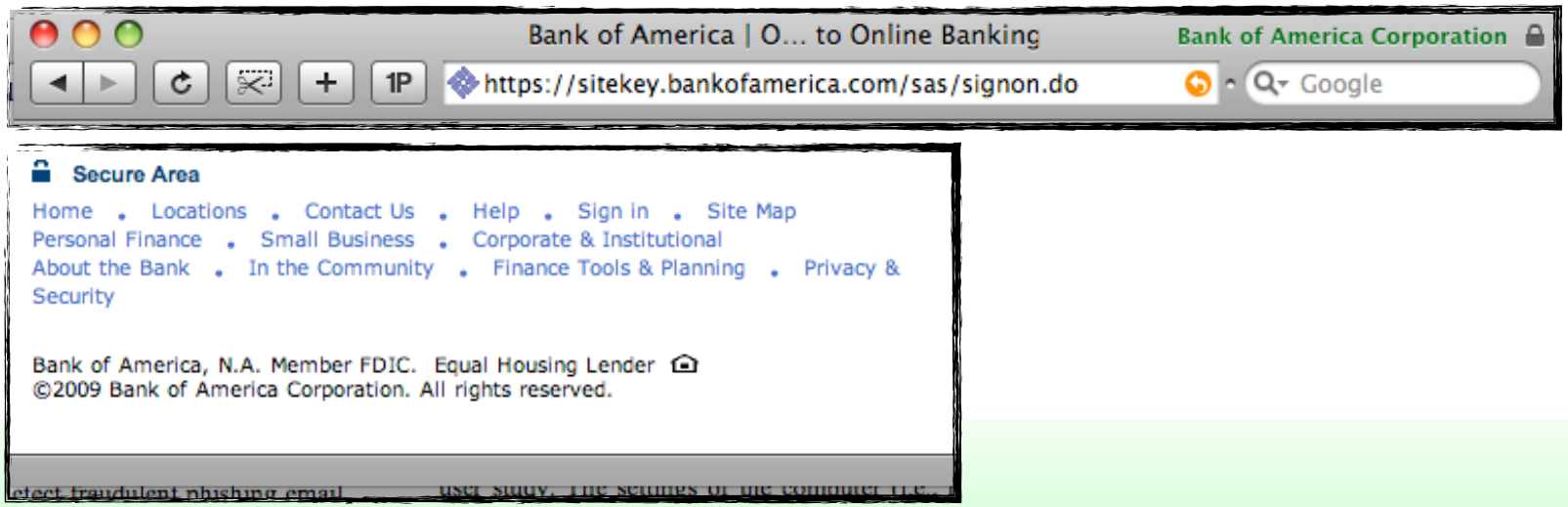


Why does phishing work? (Dhamija, Tygar & Hearst, CHI 2006)

1. Lack of knowledge

- Users do not understand how computer systems work.
- Users do not understand security.
- Users do not understand security indicators.

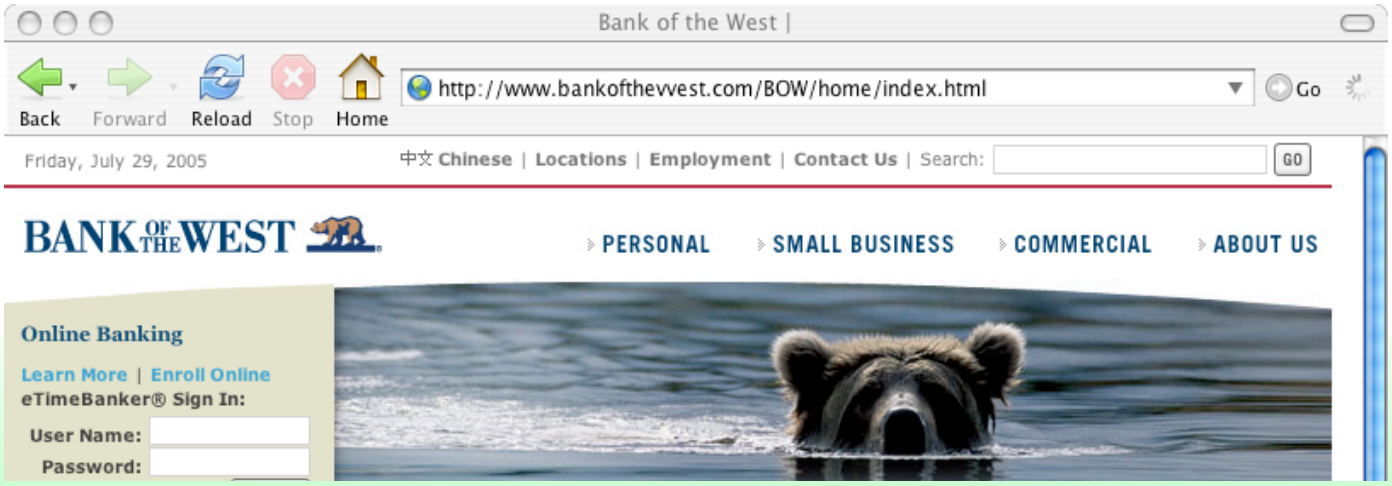
- For example:
  - ✓What does SSL do?
  - ✓What's browser "chrome?"



Why does phishing work? (Dhamija, Tygar & Hearst, CHI 2006)

2. Visual Deception

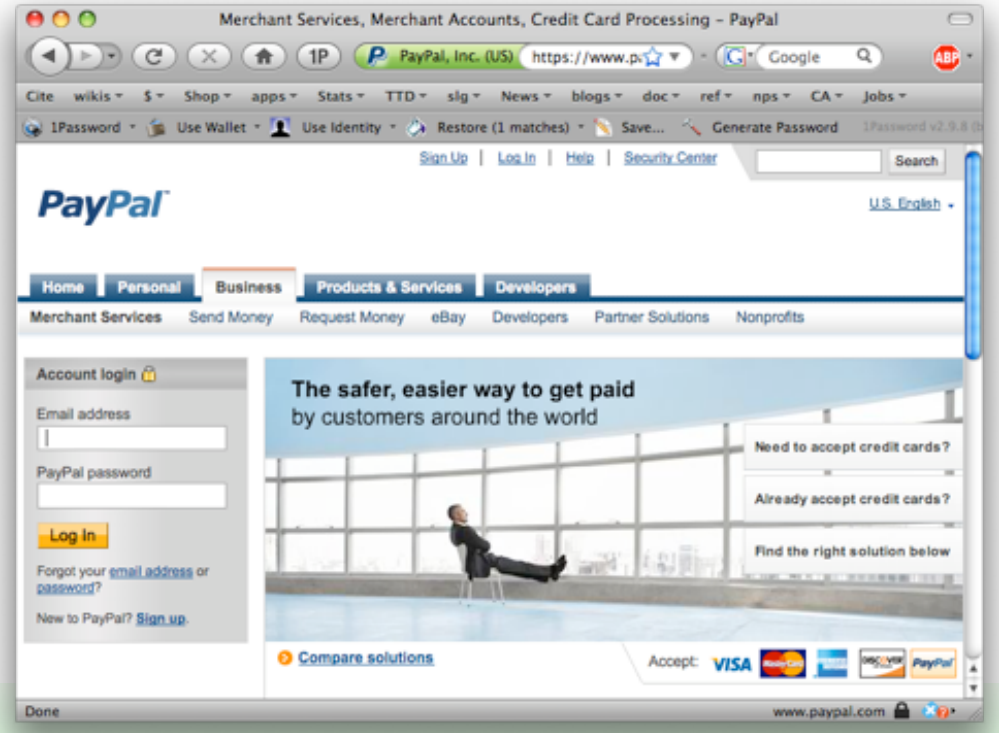
- Visually deceptive text (homographs)
- î vs. i vs. l
- http://www.bankofthevvest/
- images masking text
- images mimicking windows
- pop-up windows
- deceptive look and feel

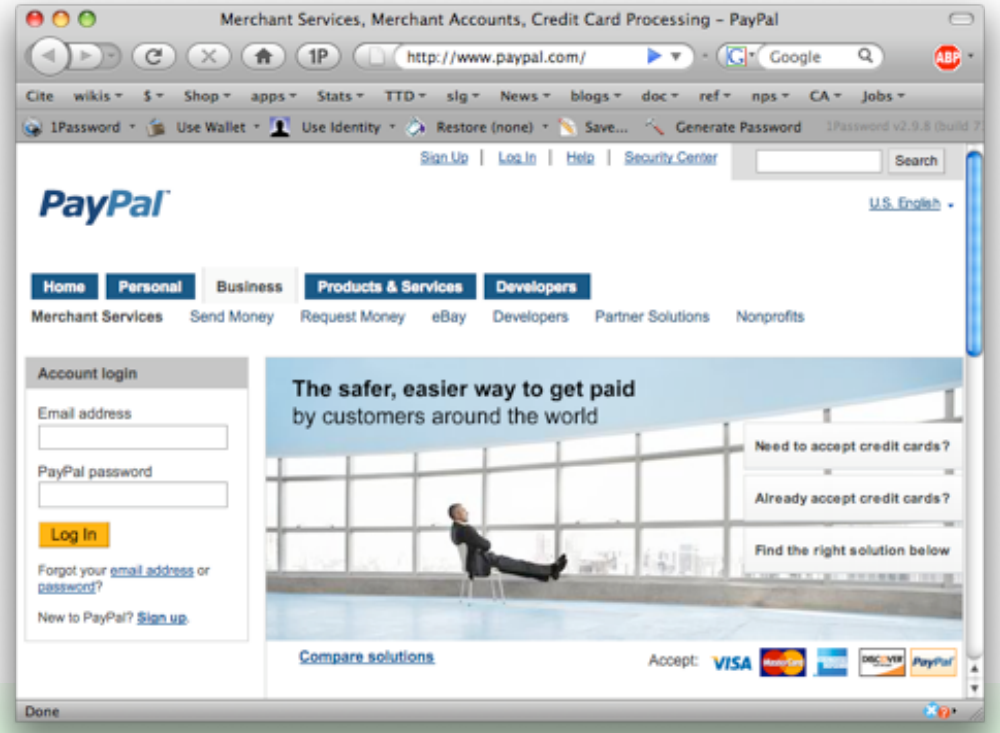


Why does phishing work? (Dhamija, Tygar & Hearst, CHI 2006)

3. Bounded Attention

- Lack of attention to security indicators
- Lack of attention to the absence of security indicators.





Why does phishing work? (Dhamija, Tygar & Hearst, CHI 2006)

Industry Practices

Website	URL	% Wrong
Bank of the West	http://bankofthevvest.com/	91%
PayPal	http://paypal-signin03.com/	59%
PayPal	http://63.4.241.49/	59%
Capital One	http://cib.ibanking-services.com/	50%
Ameritrade	http://ameritrading.net/	50%
Etrade	http://etrade.everypath.com/	77%

"Do security toolbars actually prevent phishing attacks?" (Wu, Miller & Garfinkel, CHI 2006)

Answer: NO

Reasons?

- Toolbars aren't perfect; users thought the toolbar was wrong.
- Users can rationalize the warnings if they want to do something.

User rationalizations are quite creative! (Wu, Miller & Garfinkel, CHI2006)

- 12 subjects (60%) used rationalizations to justify the indicators of the attacks that they experienced. Nine subjects explained away odd URLs with comments like:

  `www.ssl-yahoo.com` is a subdirectory of Yahoo!, like `mail.yahoo.com`.

  `sign.travelocity.com.zaga-zaga.us` must be an outsourcing site for `travelocity.com`.

  Sometimes the company [Target] has to register a different name [`www.mytargets.com`] from its brand. What if `target.com` has already been taken by another company?

  Sometimes I go to a website and the site directs me to another address which is different from the one that I have typed.

  I have been to other sites that used IP addresses [instead of domain names].

Users focus on the task, ignoring the indicators.

Nine subjects (45%) said they were spoofed *because they were focused on finishing the study task.*

Five subjects (25%) claimed *they did not notice the toolbar.*

Many subjects do not understand man-in-the-middle or keyboard sniffer attacks:

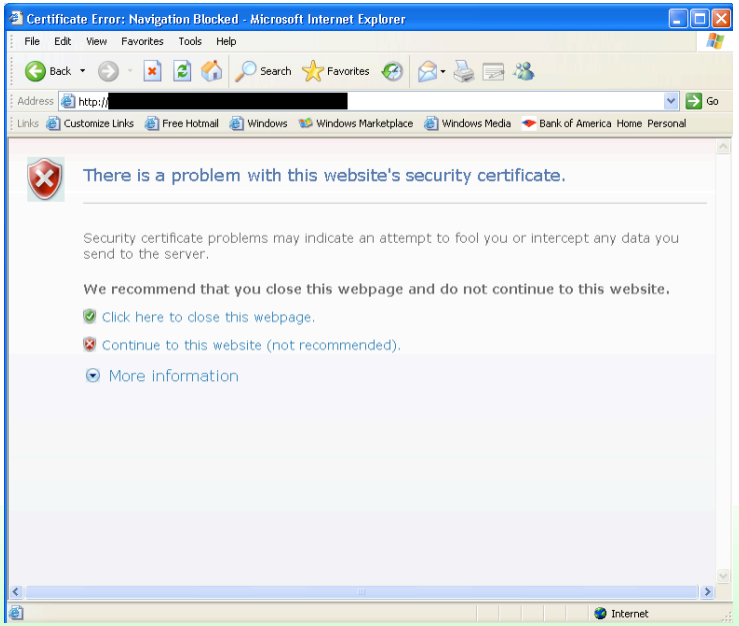
- Websites that have their data are assumed legitimate.
- Websites that "work" and don't have errors are assumed legitimate.

Users can't distinguish errors from attacks.

"The Emperor's New Security Indicators,"
Schechter, Dhamija, Ozment and Fischer, 2007

Study of Site Key.

- Study of 67 users under varying conditions.
- "https" removed
- site-authentication images remove ("site key not available at this time")
- SSL warning pages



23 of 25 participants (92%) provided username & password when Site Key authentication image was missing.

Coffee

Outline for this morning's tutorial.

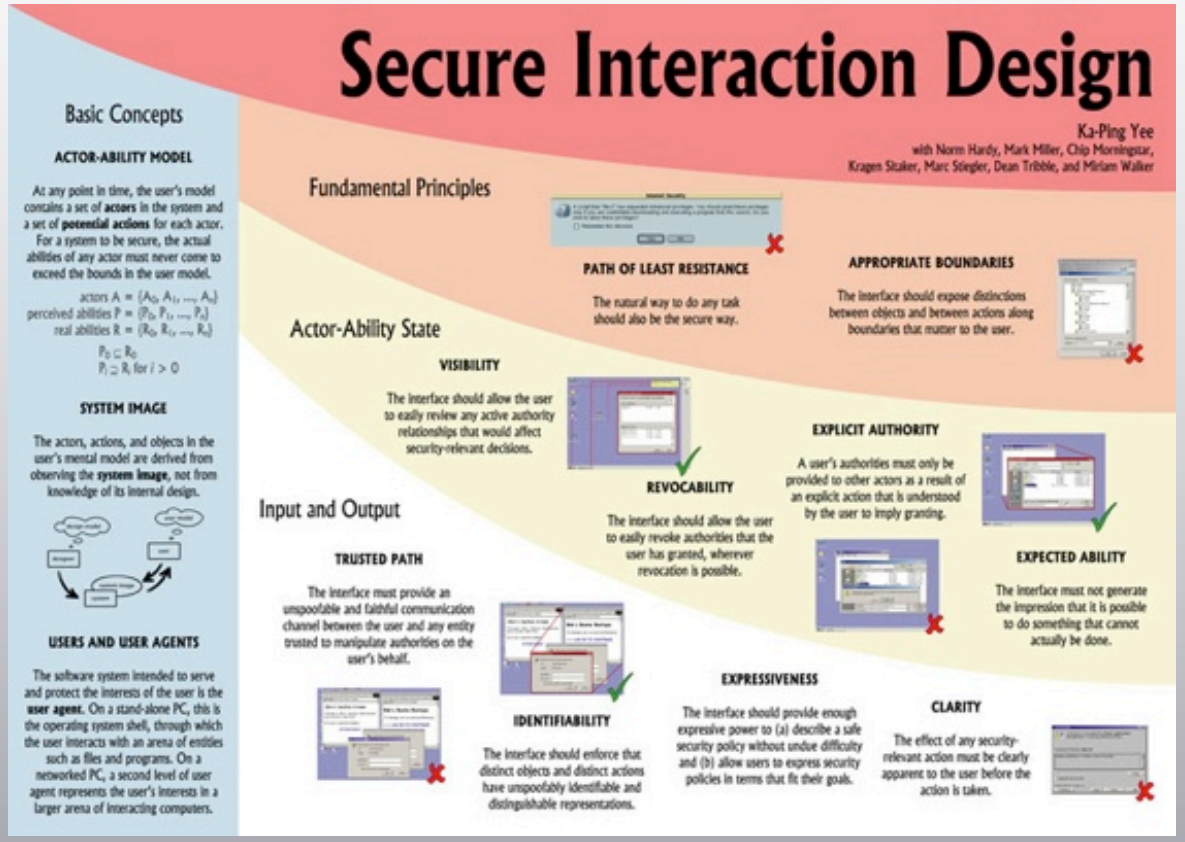
9:00 — 10:30 **What is HCI-SEC and why is it so hard?**

- Understanding Security, Usability, and HCI-SEC
- What makes HCI-SEC different?
- 10 years of HCI-SEC research

10:30 — 11:00 ***Coffee!***

11:00 — 12:30 **HCI-SEC in practice**

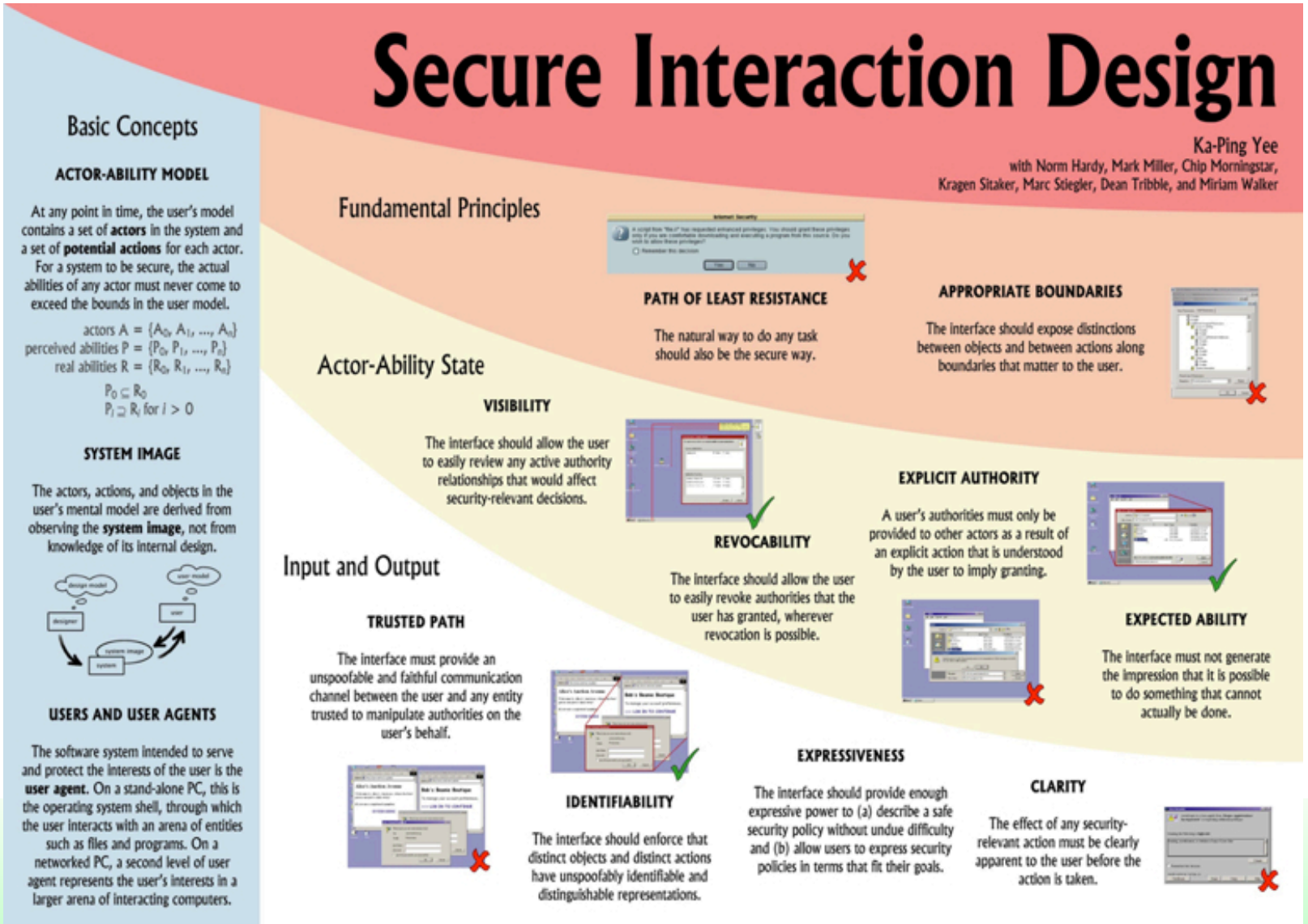
- Principles for aligning security and usability
- Specific approaches
- Where to go for more information.



Principles for aligning security and usability

"*User Interaction Design for Secure Systems*" Ka-Ping Yee, Dec. 2002

Yee proposed actor/agent models for building secure user interfaces.



Secure Interaction Design

Ka-Ping Yee
with Norm Hardy, Mark Miller, Chip Morningstar,
Kragen Sitaker, Marc Stiegler, Dean Tribble, and Miriam Walker

Basic Concepts

ACTOR-ABILITY MODEL

At any point in time, the user's model contains a set of **actors** in the system and a set of **potential actions** for each actor. For a system to be secure, the actual abilities of any actor must never come to exceed the bounds in the user model.

actors $A = \{A_0, A_1, ..., A_n\}$
perceived abilities $P = \{P_0, P_1, ..., P_n\}$
real abilities $R = \{R_0, R_1, ..., R_n\}$

$P_0 \subseteq R_0$
$P_i \supseteq R_i$ for $i > 0$

SYSTEM IMAGE

The actors, actions, and objects in the user's mental model are derived from observing the **system image**, not from knowledge of its internal design.

USERS AND USER AGENTS

The software system intended to serve and protect the interests of the user is the **user agent**. On a stand-alone PC, this is the operating system shell, through which the user interacts with an arena of entities such as files and programs. On a networked PC, a second level of user agent represents the user's interests in a larger arena of interacting computers.

Fundamental Principles

PATH OF LEAST RESISTANCE

The natural way to do any task should also be the secure way.

APPROPRIATE BOUNDARIES

The interface should expose distinctions between objects and between actions along boundaries that matter to the user.

Actor-Ability State

VISIBILITY

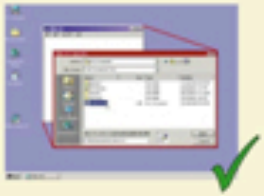
The interface should allow the user to easily review any active authority relationships that would affect security-relevant decisions.

REVOCABILITY

The interface should allow the user to easily revoke authorities that the user has granted, wherever revocation is possible.

EXPLICIT AUTHORITY

A user's authorities must only be provided to other actors as a result of an explicit action that is understood by the user to imply granting.



EXPECTED ABILITY

The interface must not generate the impression that it is possible to do something that cannot actually be done.

Input and Output

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.

EXPRESSIVENESS

The interface should provide enough expressive power to (a) describe a safe security policy without undue difficulty and (b) 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.

Garfinkel '05 proposed 6 principles for HCI-SEC.

Least Surprise/Least Astonishment

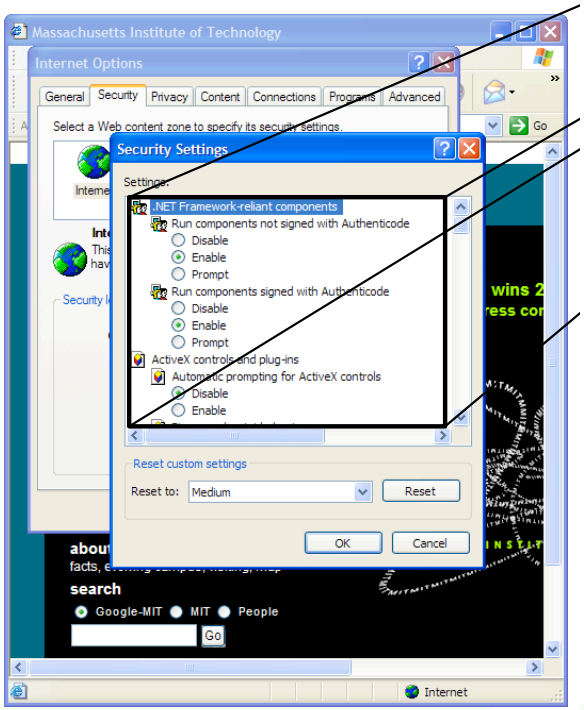
- Ensure that the system acts in accordance with user's expectations.

Good Security Now (Don't wait for perfect)

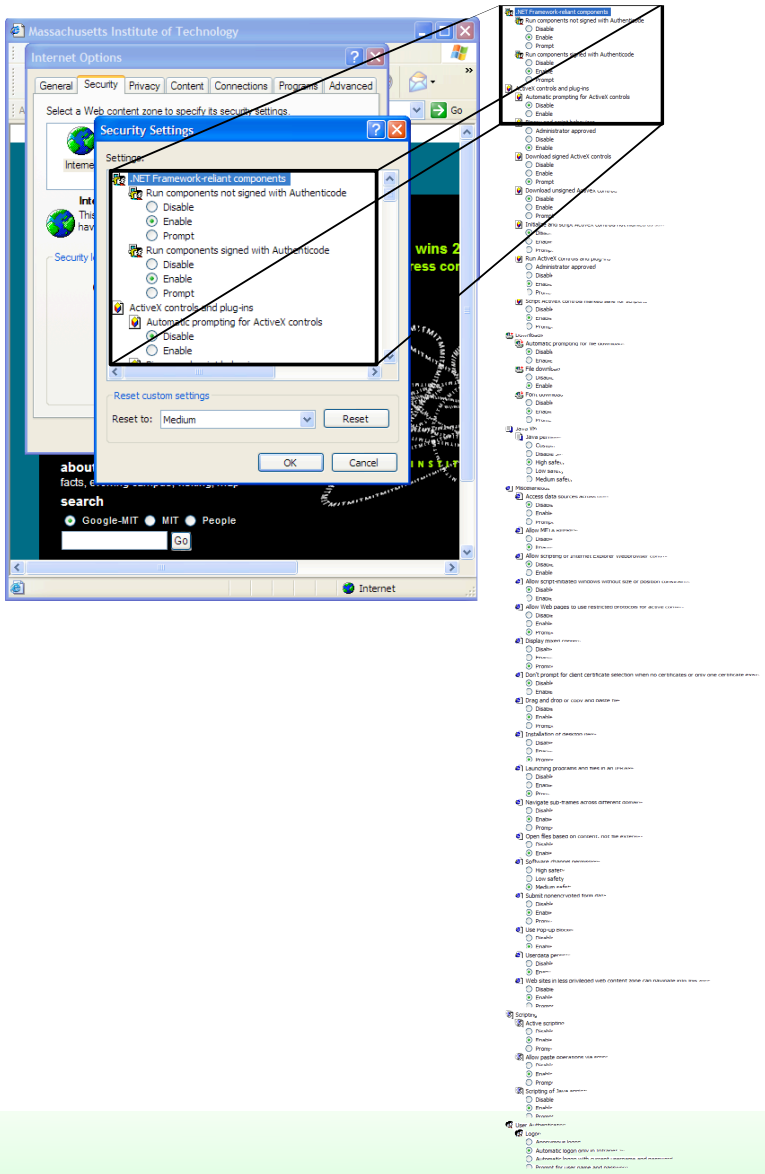
- Don't try to develop a perfect system.

Provide Standardized Security Policies (No Policy Kit)

- Avoid excessive configuration



Internet Explorer 6 had excessive security configuration



HCI-SEC Principles 4-6: Consistency & commaradery

Consistent Meaningful Vocabulary

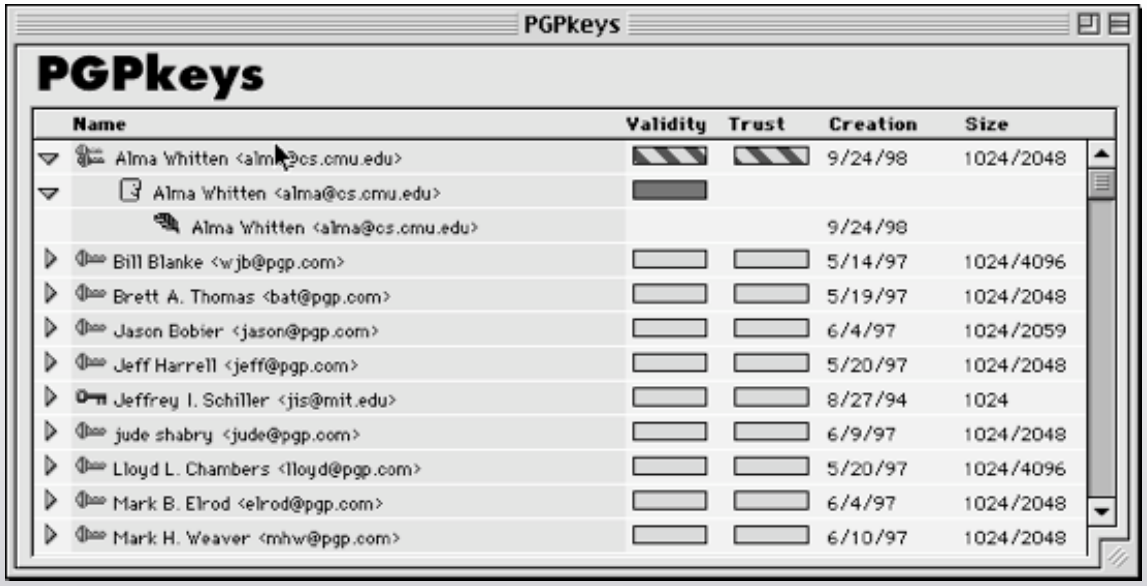
- Use words consistently to convey the same idea or concept.

Consistent Controls and Placements

- "Structure applications so that similar functionality is located in similar positions on different applications—especially if those applications are manufactured by competitors."

No External Burden

- "Design security systems to have minimal or no negative impact on the friends, associates and co-workers of those using the technology, so that they do not push back on the users of the tools and ask that the use be curtailed."



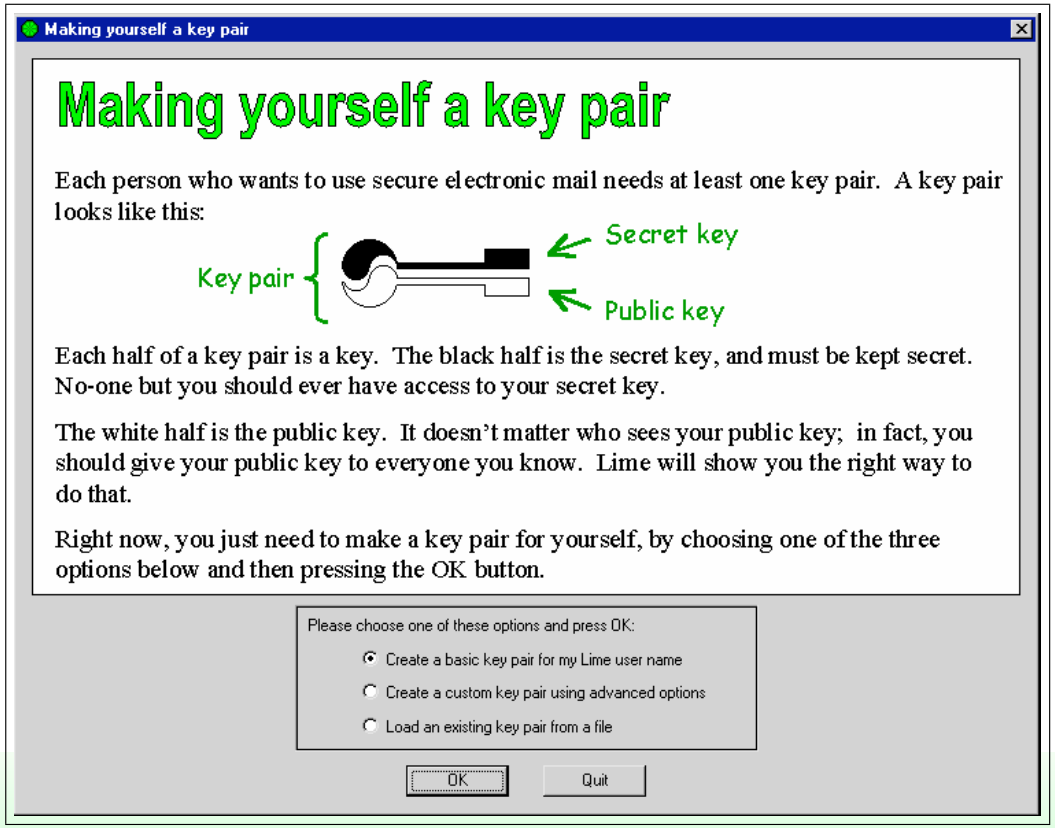
Specific ideas for aligning security & usability

Safe Staging
Metaphor Tailoring
Sanitization Patterns

"Safe Staging" and "Metaphor Tailoring."

Proposed by Whitten in 2003.

- Staging: a "safe" place to learn about security features.
- Metaphor Tailoring: "visual representations of security mechanisms."



Other examples of "metaphor tailoring:" key and hash visualization.

Which of these SHA1 hashes are the same?

1. `e5fa44f2b31c1fb553b6021e7360d07d5d91ff5e`
2. `7448d8798a438111d4b52229b45555f6f9e24e7a`
3. `a3db5c13ff90a36963278c6a39e4ee3c22e2a436`
4. `9c6b057a2b9d96a4067a749ee3b3b0158d390cf1`
5. `7448d813ff4381116327222939555 3c22e24e7a`

- ❑ 1 & 2
- ❑ 2 & 4
- ❑ 2 & 5
- ❑ 4 & 5

Other examples of "metaphor tailoring:" key and hash visualization.

Which of these SHA1 hashes are the same?

- `e5fa44f2b31c1fb553b6021e7360d07d5d91ff5e`
- `7448d8798a438111d4b52229b45555f6f9e24e7a`
- `a3db5c13ff90a36963278c6a39e4ee3c22e2a436`
- `9c6b057a2b9d96a4067a749ee3b3b0158d390cf1`
- `7448d813ff4381116327222939555 3c22e24e7a`

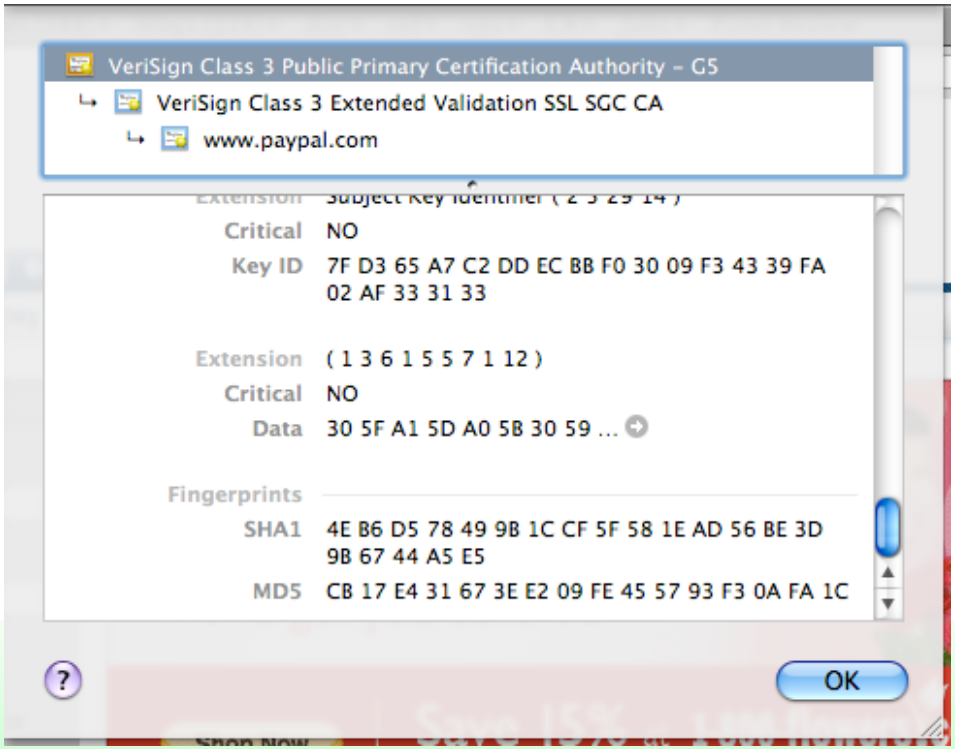
Answer: *NONE OF THEM!*

Reading and validating hashes is difficult. Yet this is a common security task.

SSH:

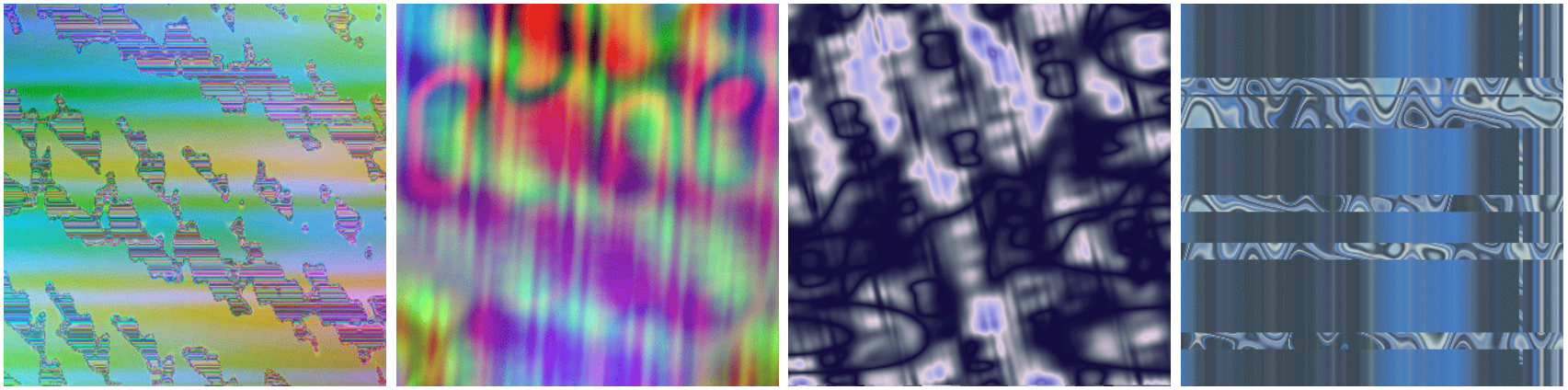
```
$ ssh www.m57.biz
The authenticity of host 'www.m57.biz (208.97.188.9)' can't be established.
RSA key fingerprint is 90:fe:0c:9c:0e:04:2f:12:a9:37:18:4c:ae:78:57:0f.
Are you sure you want to continue connecting (yes/no)?
```

Certificate fingerprints:



Perrig and Song [Cryptec 1999] proposed "Random Art" for visualizing hashes.

The random number generator is seeded by the hash:

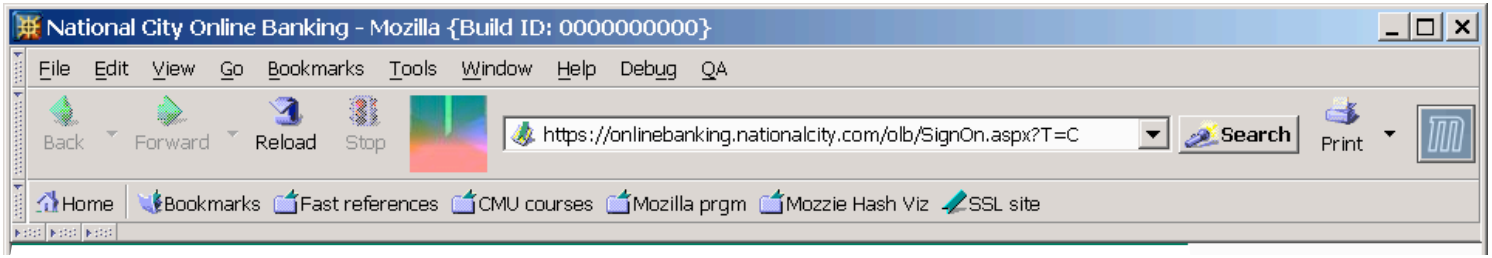


Which you will now find in SSH "visual fingerprints:"

```
$ cvs co ports
The authenticity of host 'anoncvs.usa.openbsd.org (204.152.184.203)' can't be established.
RSA key fingerprint is 49:67:9a:46:62:8a:3f:4e:b3:63:ca:d6:41:29:2a:2f.
+--[ RSA]---------+
|                 |
|                 |
|     .o o o      |
|  ..oo + *       |
| ..o.   S        |
|o  ..  .         |
|.. .=.           |
|E.oo++           |
| oooo.           |
+-----------------+
Are you sure you want to continue connecting (yes/no)?
```

Many people have experimented with "random art."

Tay '04 [CMU honors thesis]:



Jakobsson, Papadimitratos, Perrig, Wang and Wetzel [AAAS '05] proposed Random Art for Ad-Hoc authentication in wireless networks:

A big question with random art: how "random" is it?

How easy is it to make a "collision?"

Design Patterns: a “practical” HCI-SEC approach.

Design Patterns and Principles for Computer Systems that are Simultaneously Secure and Usable. PhD Thesis. MIT. 2007

Visibility and Sanitization Patterns: 5

Identification and Key Management Patterns: 9

Patterns for Promoting Secure Operation: 7

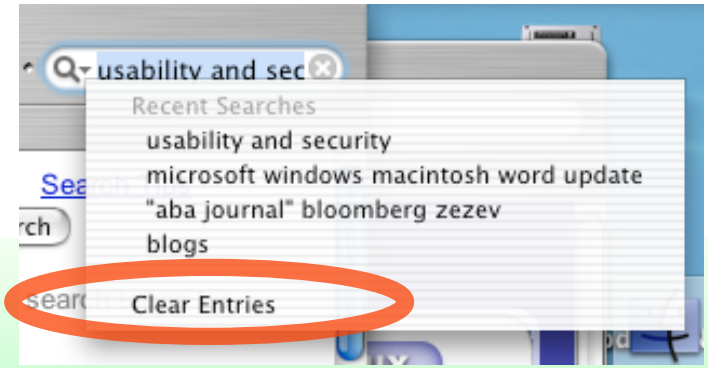
User Visibility and Sanitization Design Patterns

Explicit User Audit:

- Allow the user to inspect all user-generated information stored in the system to see if information is present and verify that it is accurate. There should be no hidden data."

Explicit Item Delete:

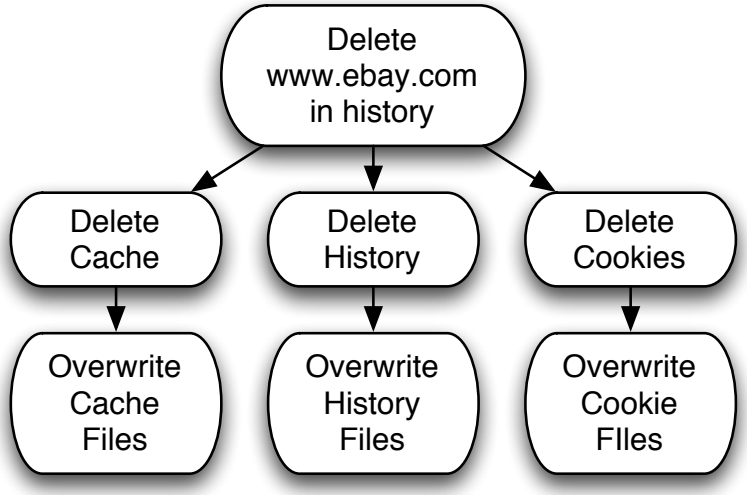
- Give the user a way to delete what is shown, where it is shown.



Complete Delete and Delayed Unrecoverable Action

Complete Delete:

- Ensure that whenthe user deletes the visible representation of something, the hidden representations are deleted as well.

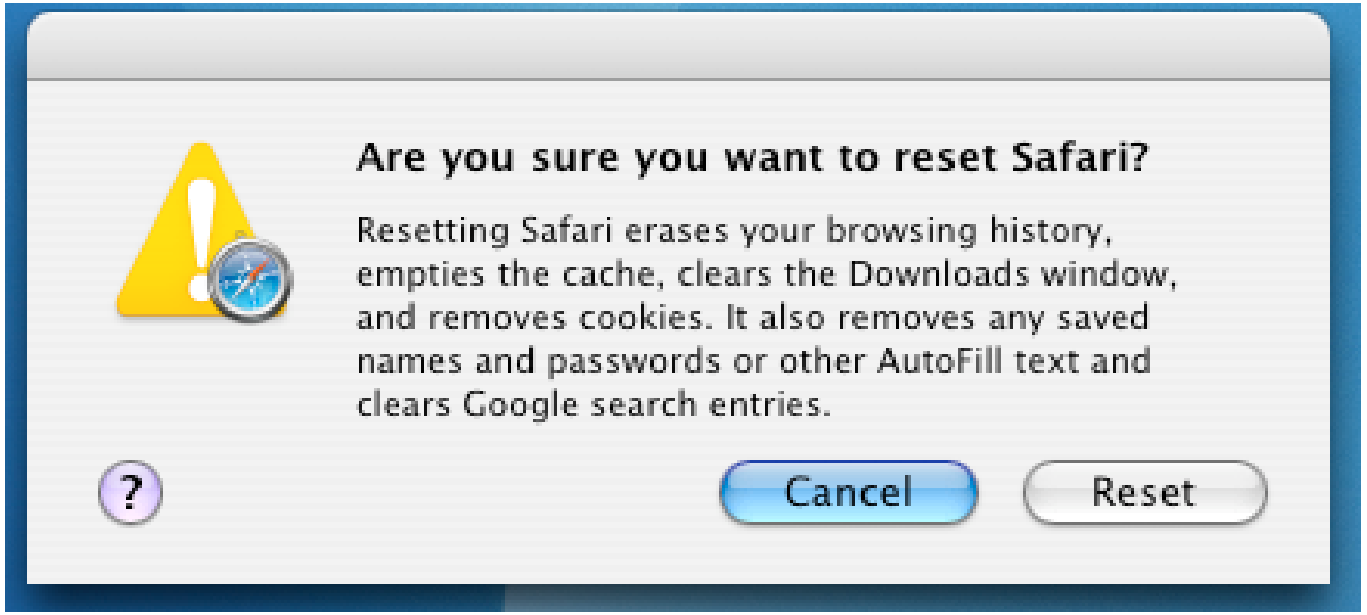


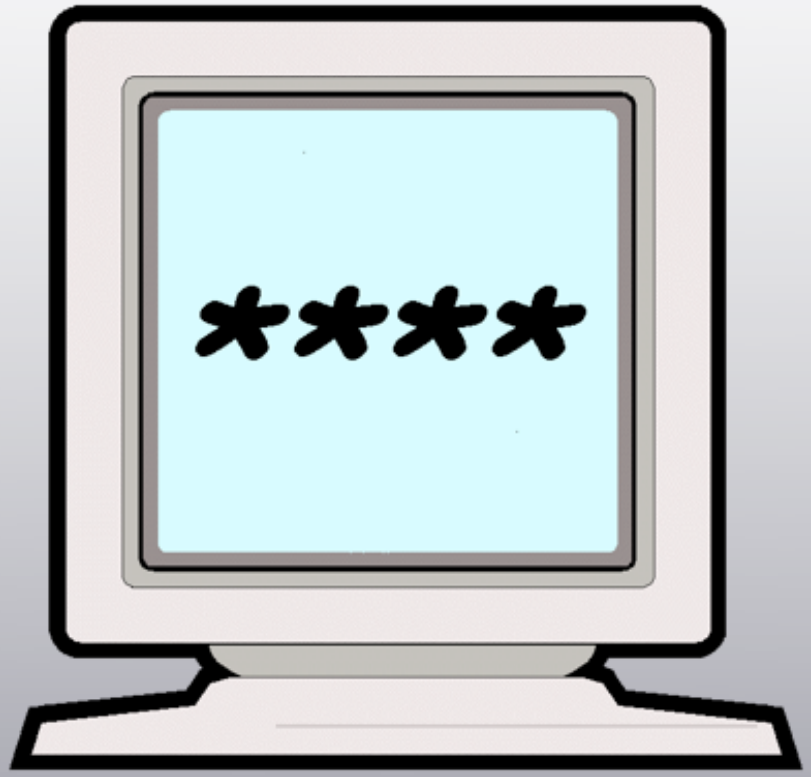
Delayed Unrecoverable Action

- Give users a chance to change their minds after executing an unrecoverable action.

Reset to Installation

- Provide a means for removing all personal or private information associated with an application or operating system in a single, confirmed, and ideally delayed operation.





Authentication

Easier Entry
Easier Authentication

Why do we use passwords?

Advantages of Passwords:

- Broad acceptance (computers & humans)
- Easy to share (promotes informal delegation)

Disadvantages:

- Same password can be used at multiple administrative domains.
- Hard to audit.
- Hard to know if a password has been compromised.

Usability problems:

- Inconsistent password restrictions (why is this a usability problem?)

Human memory does not support strong passwords.

Human Memory (Sasse):

- *Limited Capacity*
- *Items stored decay over time*
- *Cues make recall easier*
- *Non-meaningful items harder to recall than meaningful ones.*
- *"Humans cannot 'forget on demand.'"*

Passwords:

- *Unaided recall*
- *Strong passwords are hard to guess (not meaningful)*
- *Recall must be 100% correct; no feedback for failure.*

FIPS PUB 112 — Password Usage (1985)

Most password policy is based on FIPS PUB 112.

Key points:

- Passwords must be *variable length*.
- Passwords should be reset at least once a year.
  - ✓ Once a month for high-security passwords.
- Forgotten passwords must be reset (not re-issued)

Usability concerns:

- Warn people before passwords need to be changed.
- Users need to plan for good passwords.
- Do not change before weekend

Strategies for making (multiple) passwords manageable.

Single sign-on: Use one password everywhere.

Use the same password with multiple administrative domains.

- Dangerous if domains are not mutually trustworthy.

Use a password "algorithm."

- p4ssword-MIT, p4ssword-CMU, etc.

Write them down:

- On paper
- In an encrypted password keeper

Password reset systems

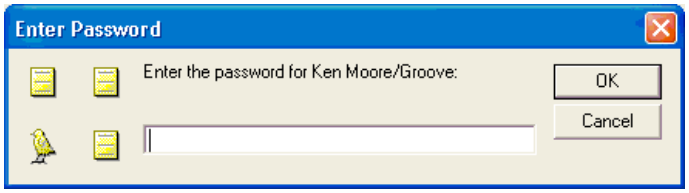
- Email recovery — effectively makes your email provider "you."

Make password entry easier. (where appropriate).

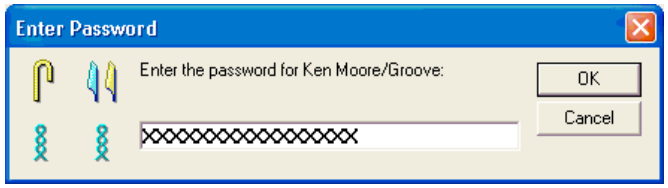
Problem: people make mistakes when all they see is:

```
PASSWORD:   •••••••
AGAIN:      •••••••
```

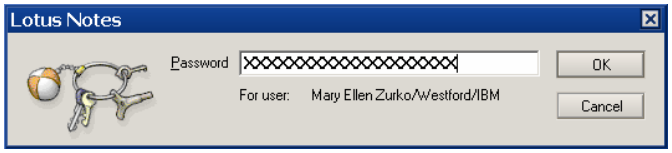
Lotus Notes solution: Password Hieroglyphics:



Bruce Tognazzini's solution: "Marching Dots:"



The iPhone does this!

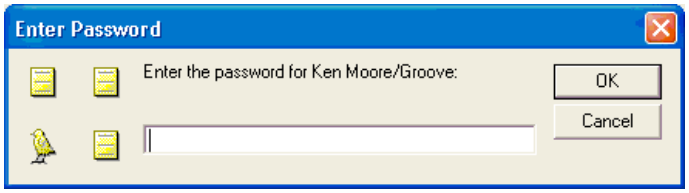


Make password entry easier. (where appropriate).

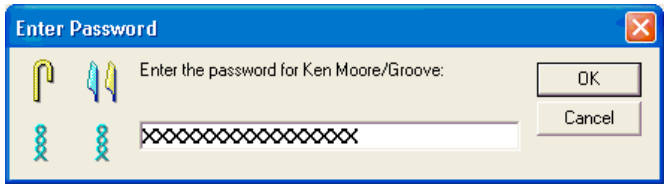
Problem: people make mistakes when all they see is:

```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

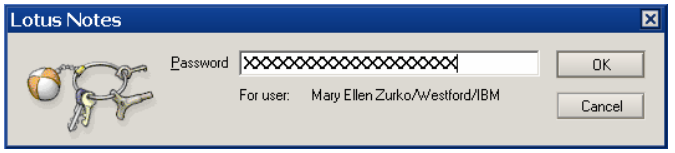


Bruce Tognazzini's solution: "Marching Dots:"




```
PASSWORD:   P
```

The iPhone does this!

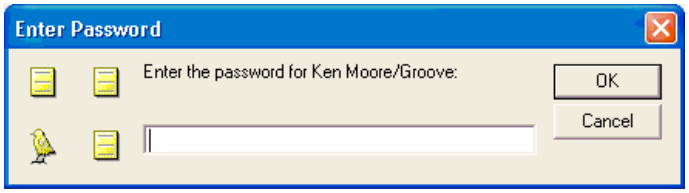


Make password entry easier. (where appropriate).

Problem: people make mistakes when all they see is:

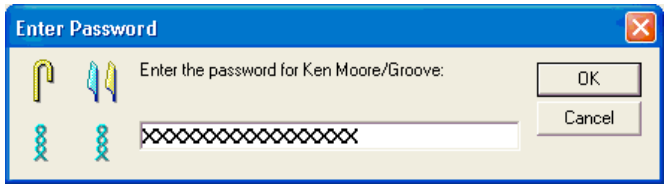
```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

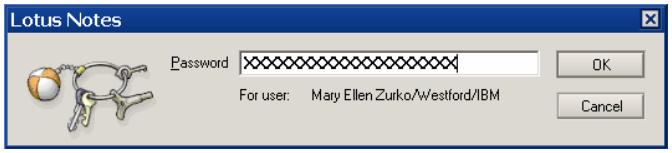


Bruce Tognazzini's solution: "Marching Dots:"

```
PASSWORD:   •A
```



The iPhone does this!

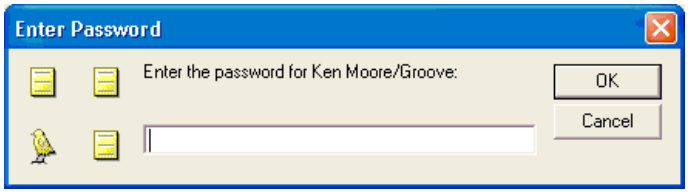


Make password entry easier. (where appropriate).

Problem: people make mistakes when all they see is:

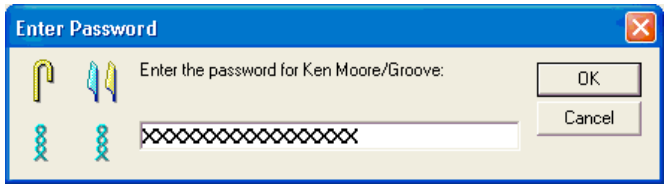
```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

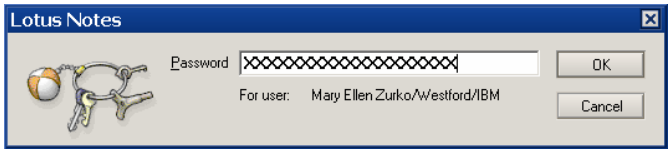


Bruce Tognazzini's solution: "Marching Dots:"

```
PASSWORD:   ••S
```



The iPhone does this!

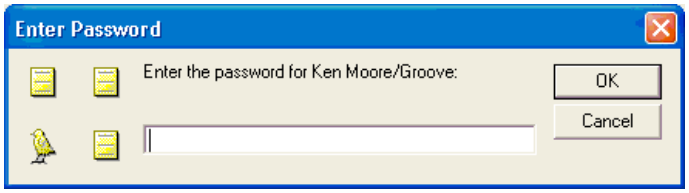


Make password entry easier. (where appropriate).

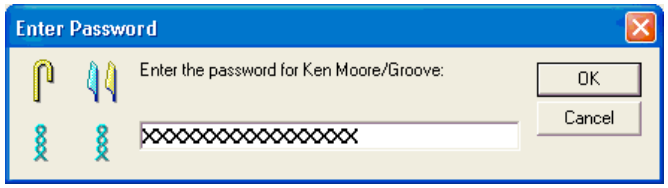
Problem: people make mistakes when all they see is:

```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

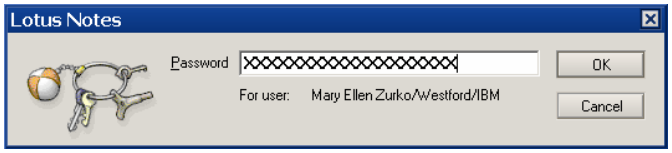


Bruce Tognazzini's solution: "Marching Dots:"




```
PASSWORD:   •••S
```

The iPhone does this!

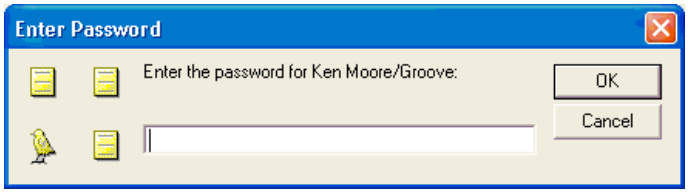


Make password entry easier. (where appropriate).

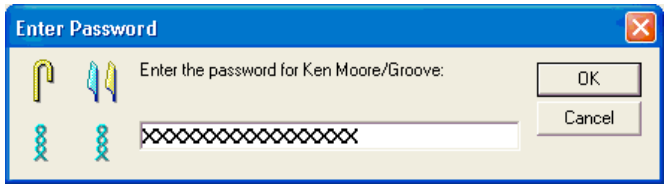
Problem: people make mistakes when all they see is:

```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

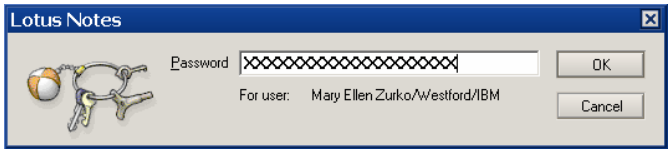


Bruce Tognazzini's solution: "Marching Dots:"




```
PASSWORD:   ••••W
```

The iPhone does this!

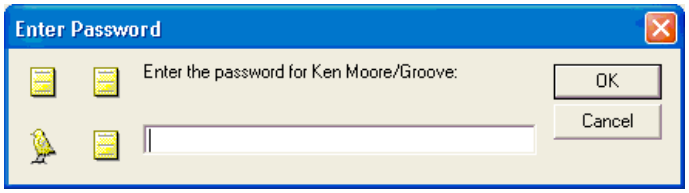


Make password entry easier. (where appropriate).

Problem: people make mistakes when all they see is:

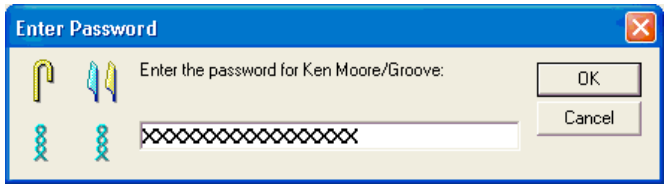
```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

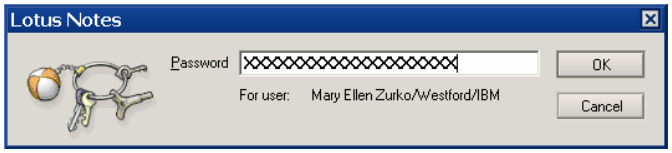


Bruce Tognazzini's solution: "Marching Dots:"

```
PASSWORD:   •••••O
```



The iPhone does this!

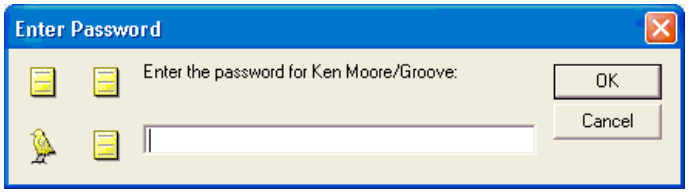


Make password entry easier. (where appropriate).

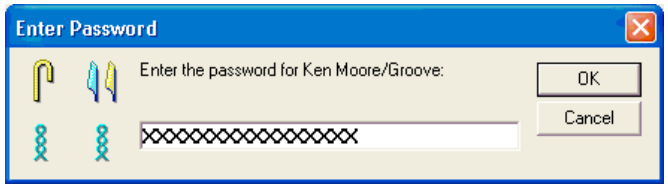
Problem: people make mistakes when all they see is:

```
PASSWORD:   •••••••
AGAIN:      •••••••
```

Lotus Notes solution: Password Hieroglyphics:

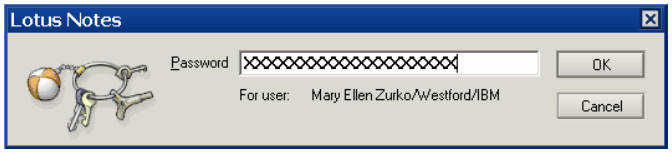


Bruce Tognazzini's solution: "Marching Dots:"




```
PASSWORD:   ••••••R
```

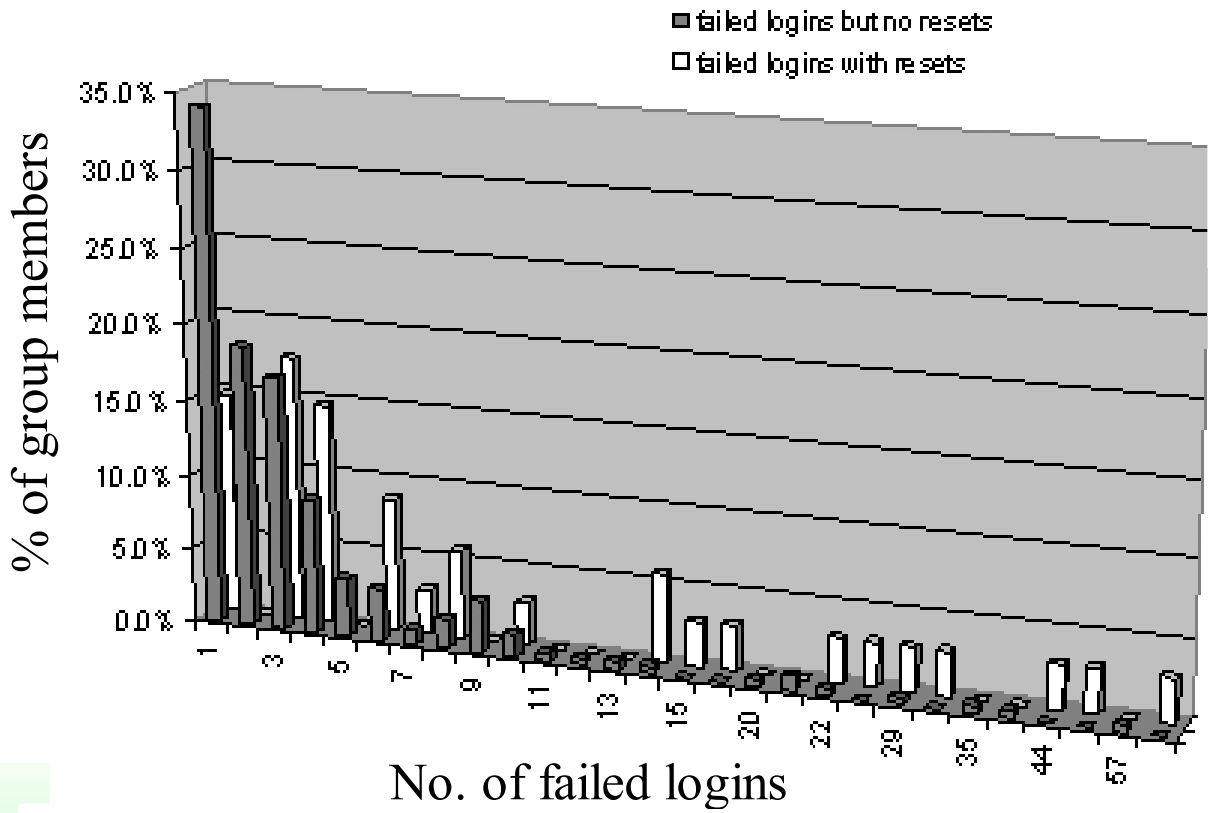
The iPhone does this!

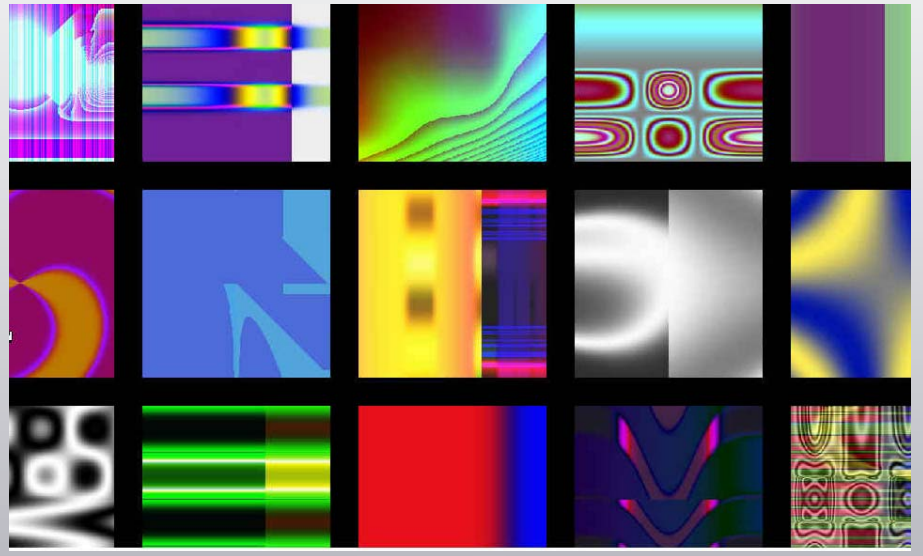


Be forgiving:
"Ten strikes and you're out."

"Ten strikes and you're out": Increasing the number of login attempts can improve password usability, *Brostoff & Sasse, CHI 2003 Workshop on HCI-SEC*

Radical idea: allow more than 3 bad passwords.





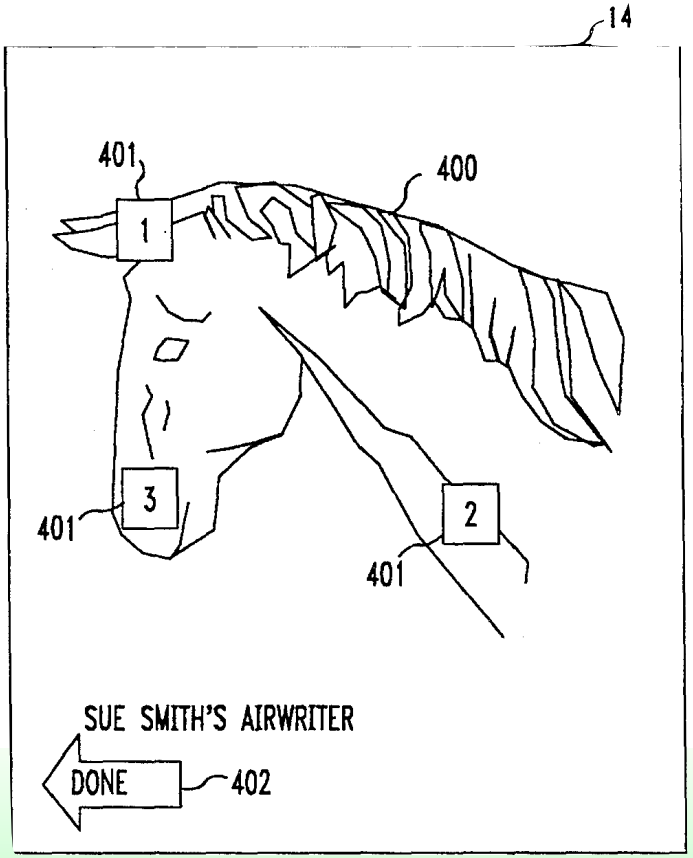
Graphical Passwords

Graphical Passwords: Something you point or draw

"Graphical Password," Greg E. Blonder, US Patent 5559961

- Click specific regions in order
- Typically 6-12 clicks

FIG. 4



Graphical passwords in practice

Passpoints (Wiedenbeck et al., 1995)

- User clicks as wishes
- Clicks aligned with invisible grid

Problems:

- Points are predictable
- People forget points (70% recall)

Graphical passwords in practice

Passpoints (Wiedenbeck et al., 1995)

- User clicks as wishes
- Clicks aligned with invisible grid

Problems:

- Points are predictable
- People forget points (70% recall)

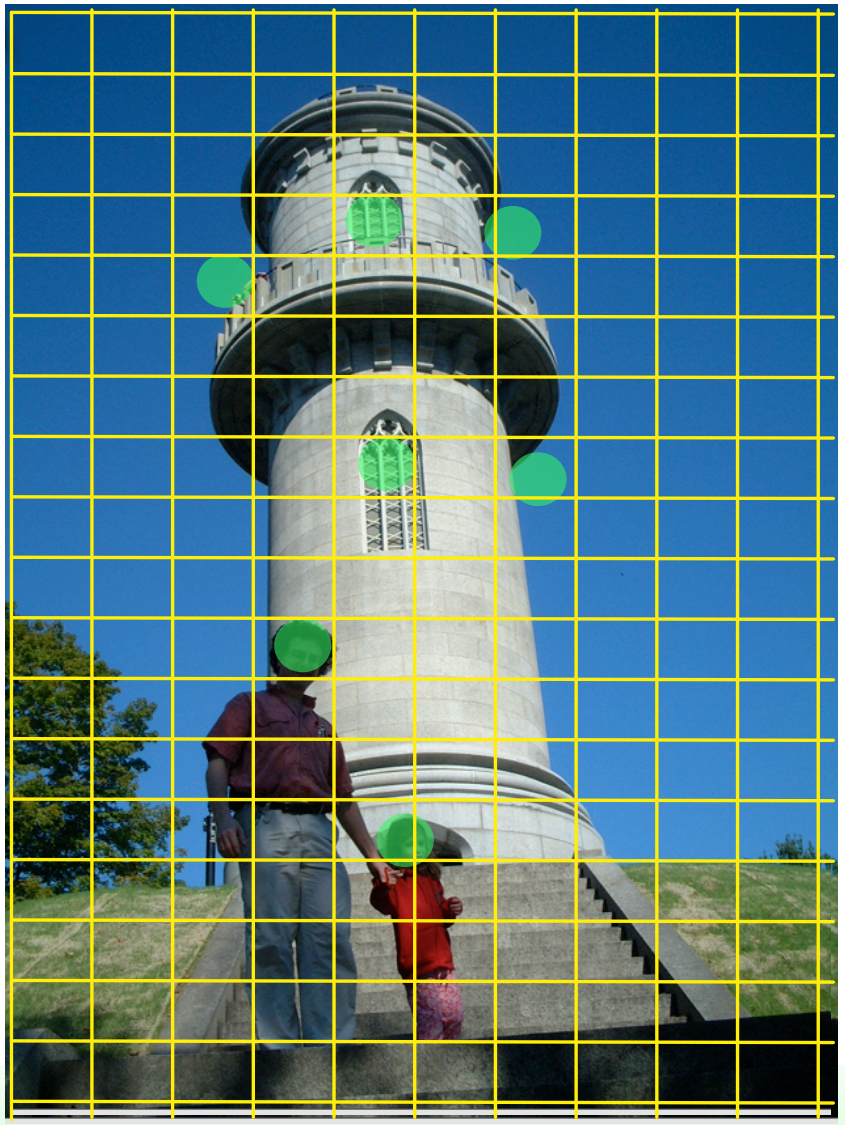
Graphical passwords in practice

Passpoints (Wiedenbeck et al., 1995)

- User clicks as wishes
- Clicks aligned with invisible grid

Problems:

- Points are predictable
- People forget points (70% recall)

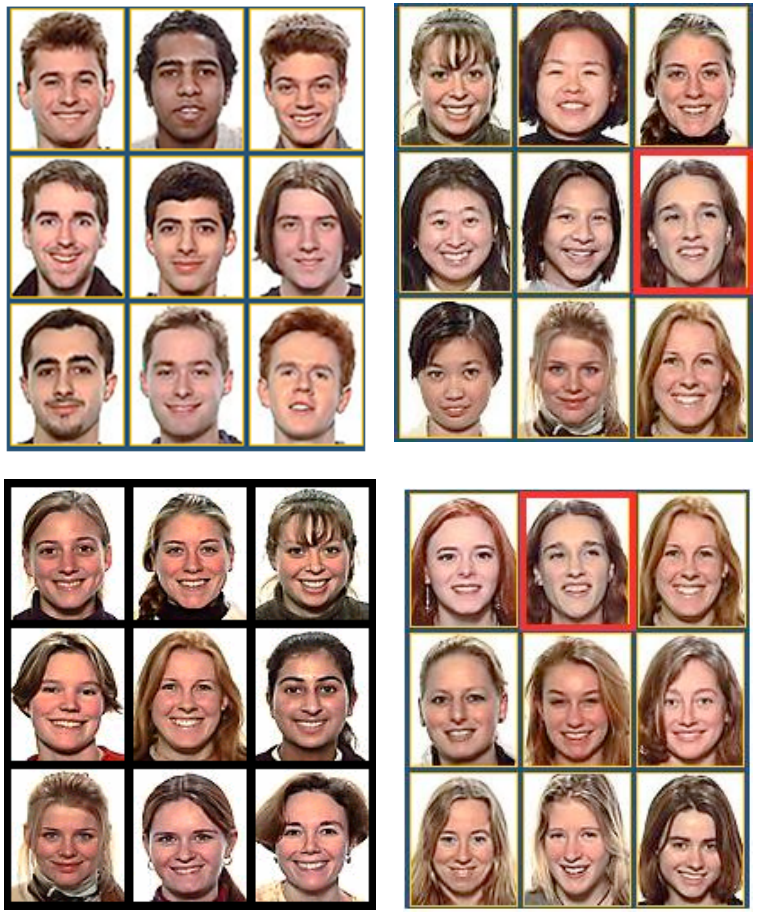


Passfaces

Multiple panels of faces.

User picks the same faces each time.

Based on ability of humans to remember faces.



Pros:

- Good recall, even after months

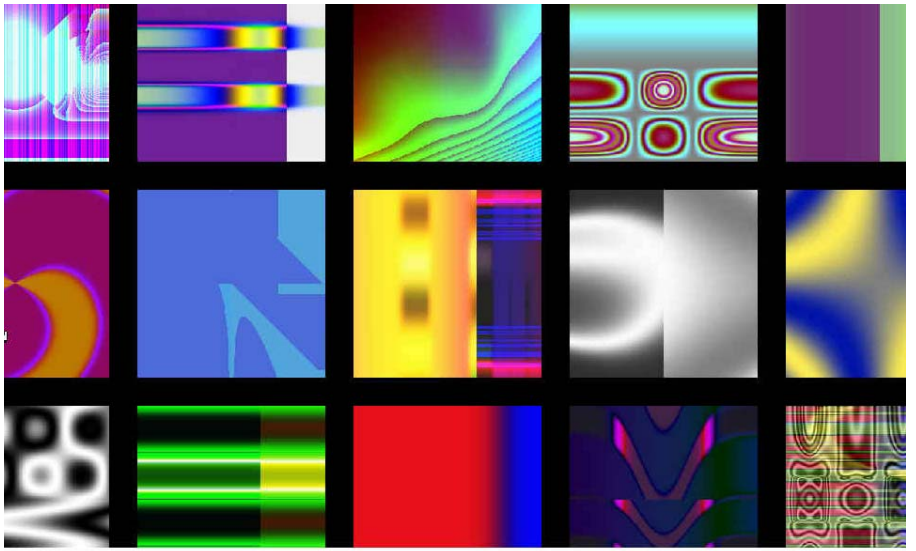
Cons:

- Face choice is predictable by race.
  (Davis, Monrose & Reiter, Security 04)
- Humans also good at describing faces.
  (Dunphy, Nicholson & Olivier, SOUPS 2008)

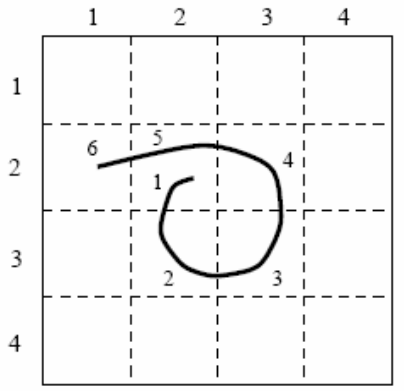
"I simply pick the best-looking girl on each page"

Déjà vu and Draw-a-secret

Déjà vu is pass faces with random art:



Draw-a-secret is like passpoints, with no background:



Both of these schemes have entropy problems.

Token-based authentication

Token-based authentication: "Something that you have."

Advantages:

- Tokens can provide end-to-end authentication
- Tokens are hard to forge.
- Token protocols defended against:
  - ✓ man-in-the-middle attack
  - ✓ replay attack
- Typically combined with PINs ("something you know")
- Can also accommodate *duress codes*.

Disadvantages:

- Cost
- Speed
- May require special hardware

Where to go from here?

Great articles you should read.

- The Protection of Information in Computer Systems, Saltzer & Schroeder, 1975, http://web.mit.edu/Saltzer/www/publications/protection/
- “Users Are Not the Enemy,” ACM Communications Dec. 1999,
- Whitten, “Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0” (Usenix Security, 1999)
- http://www.cs.auckland.ac.nz/~pgut001/pubs/usability.pdf

Websites and Conferences

HCI-SEC Bibliography:

http://www.gaudior.net/alma/biblio.html

Symposium on Usable Security and Privacy:

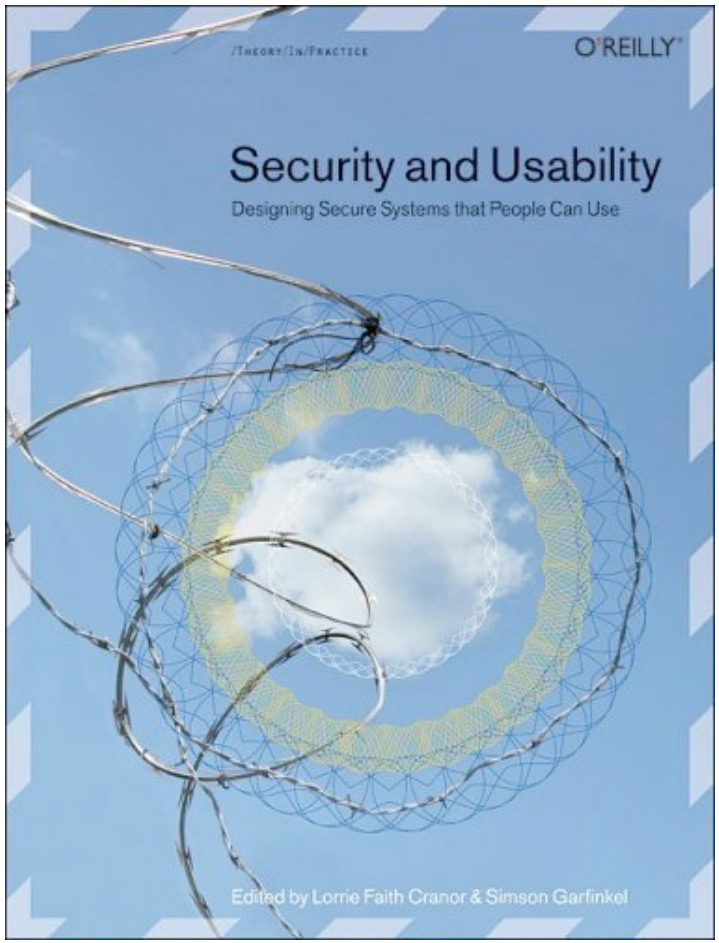
http://cups.cs.cmu.edu/soups/

Sasse's course:

http://hornbeam.cs.ucl.ac.uk/hcs/teaching/ga10.html

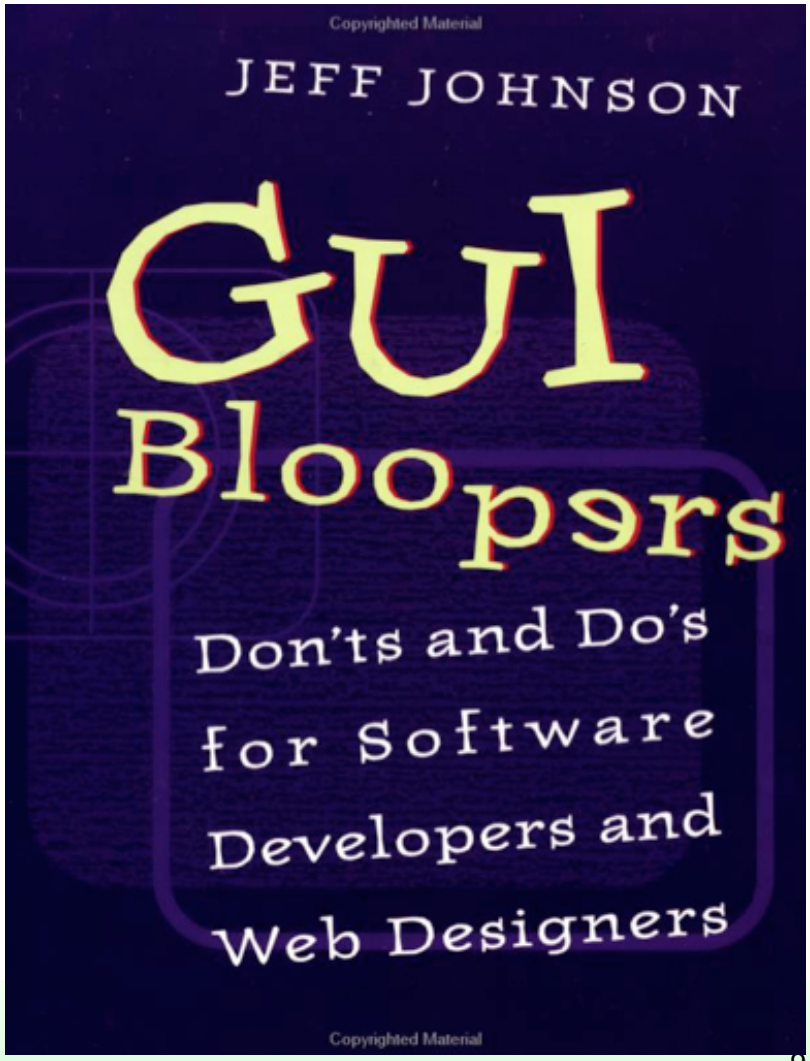
HCI-SEC: Usability & Security

Security and Usability
Cranor & Garfinkel
2005



Designing usable interfaces

Jeff Johnson, *GUI Bloopers:* Don't and Do's for Software Developers and Web Designers, Morgan Kaufmann, 2000



Please fill out your evaluations.