THE UNIVERSITY OF BRITISH COLUMBIA

Security and Usability

EECE 412
Session 23

What's More Important:

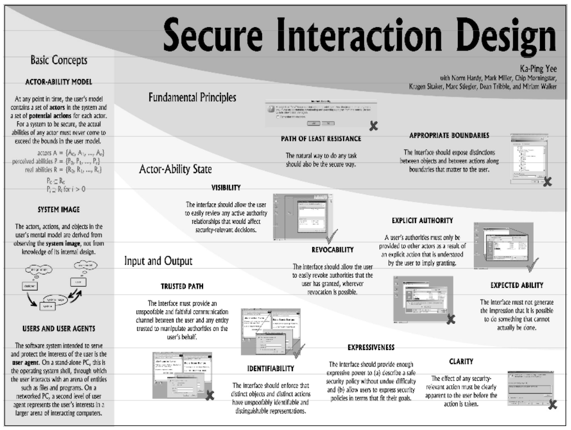
The correctness of security functions/mechanisms,
or
the correct use of them?

Outline

- Principles of secure interaction design
- Five lessons about usable security

Usability and Security Tradeoffs

- A computer is secure from a particular user's perspective if the user can depend on it and its software to behave as the user expects.
- Acceptable security is a requirement for usability.
- Acceptable usability is a requirement for security.


design model
user model
designer
user
system image
system

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

$P_0 \subseteq R_0$
$P_i \supseteq R_i$ 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

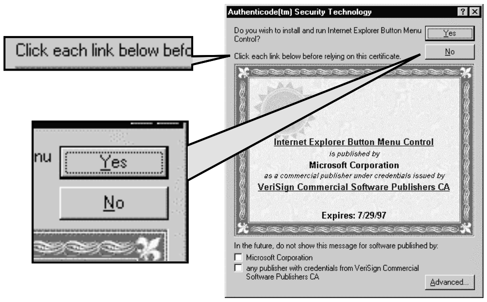
The effect of any security-relevant action must be clearly apparent to the user before the action is taken.

Principle 1:
Path of Least Resistance

To the greatest extent possible,
the natural way to do a task should be
the secure way.

Example 1: Least resistance

-



Authenticode(tm) Security Technology

Do you wish to install and run Internet Explorer Button Menu Control?

Click each link below before relying on this certificate.

Internet Explorer Button Menu Control
is published by
Microsoft Corporation
as a commercial publisher under credentials issued by
VeriSign Commercial Software Publishers CA

Expires: 7/29/97

In the future, do not show this message for software published by:

□ Microsoft Corporation
□ any publisher with credentials from VeriSign Commercial Software Publishers CA

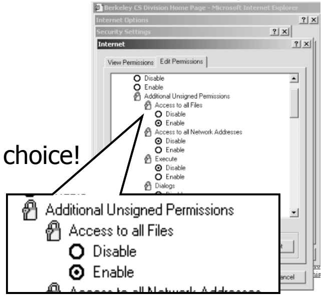
Principle 2: Appropriate Boundaries

The interface should expose, and the system should enforce, distinctions between objects and between actions that matter to the user.

I.e., any boundary that could have meaningful security implications to the user should be visible, and those that do not should not be visible.

Example 2: Bad boundaries

- A real dialog window in Internet Explorer:
- User is forced to make an all-or-nothing choice!



○ Disable
○ Enable
Additional Unsigned Permissions
Access to all Files
○ Disable
● Enable
Access to all Network Addresses
● Disable
○ Enable
Execute
● Disable
○ Enable
Dialogs

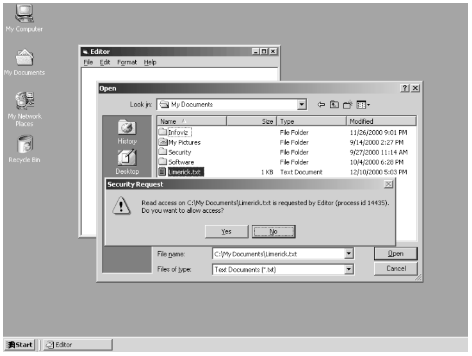
Additional Unsigned Permissions
Access to all Files
○ Disable
● Enable

Principle 3: Explicit Authorization

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

- Conflicts with Least Resistance?
- Authorizes the increase of privileges
- Combining designation with authorization

Example 3: When do we ask?



Look in: My Documents

Name	Size	Type	Modified
Infoviz		File Folder	11/26/2000 9:01 PM
My Pictures		File Folder	9/14/2000 2:27 PM
Security		File Folder	9/27/2000 11:14 AM
Software		File Folder	10/4/2000 6:28 PM
Limerick.txt	1 KB	Text Document	12/10/2000 5:03 PM

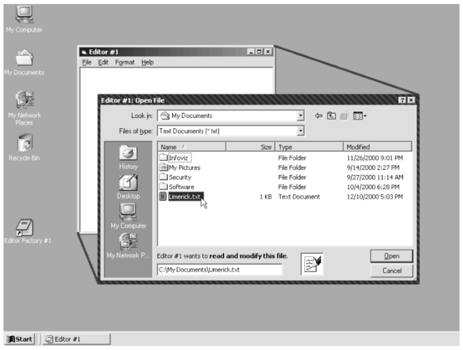
Security Request

Read access on C:\My Documents\Limerick.txt is requested by Editor (process id 14435).
Do you want to allow access?

File name: C:\My Documents\Limerick.txt

Files of type: Text Documents (*.txt)

Example 3: When do we ask?



Look in: My Documents

Files of type: Text Documents (*.txt)

Name	Size	Type	Modified
Infoviz		File Folder	11/26/2000 9:01 PM
My Pictures		File Folder	9/14/2000 2:27 PM
Security		File Folder	9/27/2000 11:14 AM
Software		File Folder	10/4/2000 6:28 PM
Limerick.txt	1 KB	Text Document	12/10/2000 5:03 PM

Editor #1 wants to **read and modify this file.**

C:\My Documents\Limerick.txt

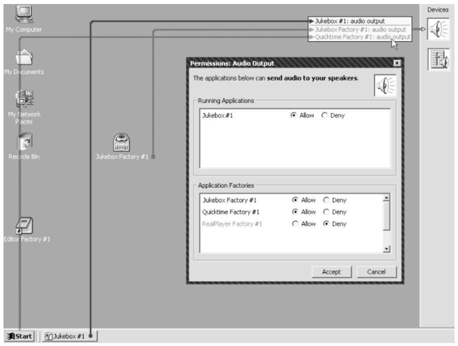
Principle 4: Visibility

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

Example 4: What do we show?

Not this:

Example 4: What do we show?



Permissions: Audio Output

The applications below can **send audio to your speakers**.

Running Applications

Jukebox #1	● Allow	○ Deny

Application Factories

Jukebox Factory #1	● Allow	○ Deny
Quicktime Factory #1	● Allow	○ Deny
RealPlayer Factory #1	○ Allow	● Deny

Principle 5: Identifiability

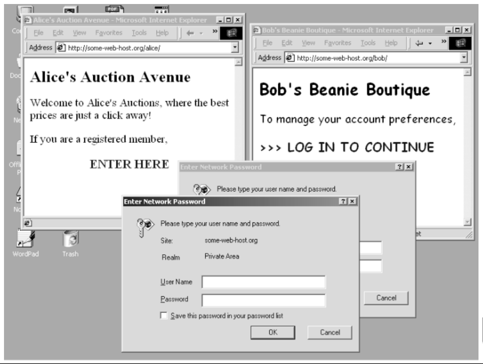
The interface should enforce that distinct objects and distinct actions have unspoofably identifiable and distinguishable representations.

two aspects

- Continuity: the same thing should appear the same
- Discriminability: different things should appear different

- *perceived* vs. *be* different/same

Example 5: Violating identifiability



Alice's Auction Avenue - Microsoft Internet Explorer

Address: http://some-web-host.org/alice/

Alice's Auction Avenue

Welcome to Alice's Auctions, where the best prices are just a click away!

If you are a registered member,

ENTER HERE

Bob's Beanie Boutique - Microsoft Internet Explorer

Address: http://some-web-host.org/bob/

Bob's Beanie Boutique

To manage your account preferences,

>>> LOG IN TO CONTINUE

Enter Network Password

Please type your user name and password.

Site: some-web-host.org

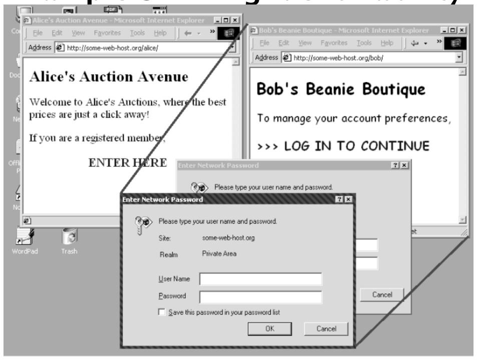
Realm: Private Area

User Name

Password

□ Save this password in your password list

Example 5: Fixing identifiability



Alice's Auction Avenue - Microsoft Internet Explorer

Address: http://some-web-host.org/alice/

Alice's Auction Avenue

Welcome to Alice's Auctions, where the best prices are just a click away!

If you are a registered member,

ENTER HERE

Bob's Beanie Boutique - Microsoft Internet Explorer

Address: http://some-web-host.org/bob/

Bob's Beanie Boutique

To manage your account preferences,

>>> LOG IN TO CONTINUE

Enter Network Password

Please type your user name and password.

Site: some-web-host.org

Realm: Private Area

User Name

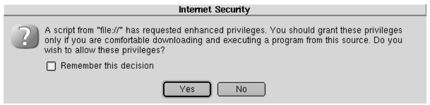
Password

□ Save this password in your password list

Principle 6: Clarity

The effect of any security-relevant action must be apparent before the action is taken.

Example 6: Violating Clarity



Internet Security

A script from "file://" has requested enhanced privileges. You should grant these privileges only if you are comfortable downloading and executing a program from this source. Do you wish to allow these privileges?

☐ Remember this decision

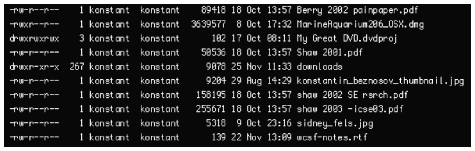
What program? What source?
What privileges? What purpose?
How long? How to revoke?
Remember this decision? *What* decision?

Might as well click “Yes”: it’s the default.

Principle 7: Expressiveness

In order for the security policy enforced by the system to be useful, we must be able to express a **safe policy**, and we must be able to express the **policy we want**.

Example 7: Unix File Permissions



```
-rw-r--r--    1 konstant  konstant    89418 18 Oct 13:57 Berry 2002 painpaper.pdf
-rwxr--r--    1 konstant  konstant  3639577  8 Oct 17:32 MarineAquarium206_OSX.dmg
drwxrwxrwx    3 konstant  konstant      102 17 Oct 08:11 My Great DVD.dvdproj
-rw-r--r--    1 konstant  konstant    50536 18 Oct 13:57 Shaw 2001.pdf
drwxr-xr-x  267 konstant  konstant     9078 25 Nov 11:33 downloads
-rw-r--r--    1 konstant  konstant     9204 29 Aug 14:29 konstantin_beznosov_thumbnail.jpg
-rw-r--r--    1 konstant  konstant   158195 18 Oct 13:57 shaw 2002 SE rsrch.pdf
-rw-r--r--    1 konstant  konstant   255671 18 Oct 13:57 shaw 2003 -icse03.pdf
-rw-r--r--    1 konstant  konstant     5318  9 Oct 23:16 sidney_fels.jpg
-rw-r--r--    1 konstant  konstant      139 22 Nov 13:09 wcsf-notes.rtf
```

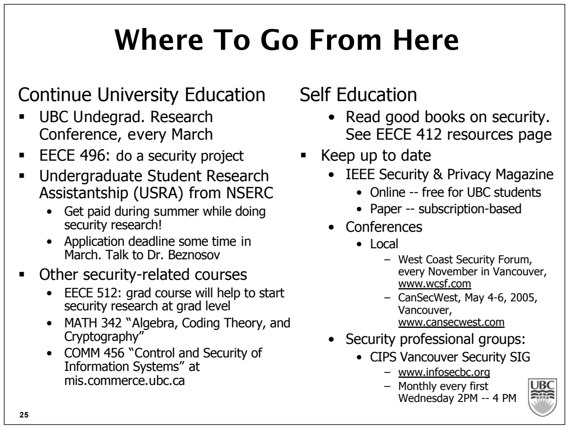
Design Principles Summary

In order to use a system safely, a user needs to have confidence in all of the following statements:

1. Things don't become unsafe all by themselves. (Explicit Authorization)
2. I can know whether things are safe. (Visibility)
3. I can make things safer. (Revocability)
4. I don't choose to make things unsafe. (Path of Least Resistance)
5. I know what I can do within the system. (Expected Ability)
6. I can distinguish the things that matter to me. (Appropriate Boundaries)
7. I can tell the system what I want. (Expressiveness)
8. I know what I'm telling the system to do. (Clarity)
9. The system protects me from being fooled. (Identifiability, Trusted Path)

Lessons learned about usable security

1. You cannot retrofit usable security
   - Adding explanatory dialogs to a confusing system makes it more confusing
2. Tools are not solutions
   - They are just Lego™ blocks
3. Mind the upper layers
   - Application-level security design allows intentional, implicit, application-specific security
4. Keep your users satisfied
   - Put your users’ needs first
   - Evaluate your solution on the target audience
5. Think locally, act locally
   - Don’t assume support from global infrastructure (e.g., PKI)
   - If a generic security tool can be used independently of application, the user(s) must deal with it directly



Where To Go From Here

Continue University Education

- UBC Undegrad. Research Conference, every March
- EECE 496: do a security project
- Undergraduate Student Research Assistantship (USRA) from NSERC
  - Get paid during summer while doing security research!
  - Application deadline some time in March. Talk to Dr. Beznosov
- Other security-related courses
  - EECE 512: grad course will help to start security research at grad level
  - MATH 342 "Algebra, Coding Theory, and Cryptography"
  - COMM 456 "Control and Security of Information Systems" at mis.commerce.ubc.ca

Self Education

  - Read good books on security. See EECE 412 resources page
- Keep up to date
  - IEEE Security & Privacy Magazine
    - Online -- free for UBC students
    - Paper -- subscription-based
  - Conferences
    - Local
      - West Coast Security Forum, every November in Vancouver, www.wcsf.com
      - CanSecWest, May 4-6, 2005, Vancouver, www.cansecwest.com
  - Security professional groups:
    - CIPS Vancouver Security SIG
      - www.infosecbc.org
      - Monthly every first Wednesday 2PM -- 4 PM