11. Usability Issues
ENEE 757 | CMSC 818V

Prof. Tudor Dumitraș
Assistant Professor, ECE
University of Maryland, College Park

http://ter.ps/757
https://www.facebook.com/SDSAatUMD

Today’s Lecture

• Where we’ve been
  – Security basics
  – Analytics
  – Passwords
  – Biometrics
  – Authorization logic
  – Web authentication

• Where we’re going today
  – Usability issues in access control

• Where we’re going next
  – Internet protocols and denial of service
  – Monday: Project checkpoint #1
Security and Usability

- Software systems interact with humans
  - Administrators, end users

- Humans are participants in the security protocols
  - Trusted participants => human failures can compromise the system’s security!
  - Security features are no good if humans can’t use them
    - Or if users turn them off
  - A system isn’t secure if users always make mistakes
    - Example: choosing weak passwords

- In practice, a system cannot be secure unless it is usable

Example: Access Control Lists

UNIX permissions:
- rwx rwx rwx
- ownr grp othr

AFS permissions

What are my effective permissions? Who else has access?
2003 Senate Judiciary Committee Data Breach

- Republican clerk discovers that he can access the home directories of Democratic staffers
  - Reports, memos, etc. make their way to Republican staff and press

- Home directories set up with open permissions by confused system administrator
  - Was not instructed on setting the correct permissions
  - Assumed that access was restricted by some other means

Many Other Examples of Incorrect Permissions

- Students suspended after seeing topless picture of teacher on school-issued iPad
- Boss Allegedly Downloaded Nude Photos From Employee’s Cell Phone, Showed Them Around The Office (Forbes.com)
How to Make User Interfaces Usable?

1. Strive for consistency
   - Prompts, commands, actions
2. Enable frequent users to use shortcuts
   - Experts desire to increase the pace of interaction
3. Offer informative feedback
   - What does “child died” mean to user who never took an OS class?
4. Design dialog to yield closure
   - Confirm outcome of sequence of actions
5. Offer simple error handling
   - Design system so that user cannot make errors
6. Permit easy reversal of actions
   - Undo capability makes user more comfortable with the system
   - How to undo a security breach?
7. Support internal locus of control
   - Users should be initiators of actions rather than responders
8. Reduce short-term memory load
   - Keep displays simple, consolidate multiple pages, reduce window motion

Eight Golden Rules of Interface Design
[Ben Shneiderman]
1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialog to yield closure
5. Offer simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load
Usability Problems for ACL Interfaces

Typically present a list of permission rules for users and groups

- **Scale**
  - Only one rule at a time is visible

- **Hierarchies**
  - Effective permissions may depend on the permissions of the parent folder

- **Groups**
  - Group membership information is not visible

- **Rule conflicts**
  - When rules interact, it isn’t clear what the outcome will be

The Expandable Grid Interface

[Reeder, Bauer, Cranor, Reiter, Vaniea CHI 2008, CHI 2011]

- Shows effective policy instead of policy rules
- Shows both user hierarchies (groups) and file hierarchies (folders)
- Shows entire policy on one screen
Tag Based Policy

- Determine access policies to semantic groups of files (specified by tags), rather than individual files and folders
- Alice lets Bob view files with type=photo and album=vacation_photos
  - Minimize policy mismatch: intended vs. specified
  - Allow fine-grained policy
  - Users prefer tags to folders (think iTunes)
  - File hierarchies vs. tags (semantic naming)

Logic-Based Access Control

- Need Bob speaksfor Alice or similar credential
  - Options presented to Alice:
    - Grant Bob credential to access VacationPhotos
    - Add Bob to group CloseFriends who already have access
    - Give Bob same rights as Charlie, who has access
- Proof of Bob says open(vacation_photos) ➔ Alice says open(vacation_photos)
- Proof of Alice says open(vacation_photos) ➔ Bob says open(vacation_photos)
Specifying Access Control Policies Reactively
[Mazurek et al., CHI 2011]

- File owner responds to requests for access
  - There may be several ways to create a credential for granting access
  - Owner can control the granularity of access granted

- Answering explicit requests makes users think about the consequences

- Trade-off
  - Per-file requests: list of files must be public
  - Per-tag requests (or search conditions, using tag combinations): granting coarse-grain access

Benefits of Logic-Based Access Control

- Provides high assurance of correctness

- Meaningful audit trail

- Allows fine-grained control, flexible policy
  - Users can create temporary access credentials
    - Useful for controlling access to physical resources, e.g. doors
  - Provide temporary access to email without revealing information that could be used at a later time or to access a different resource
    - Such as passwords
  - Can create complex access control credentials
    - Example: any three colleagues may access the resource when acting together, but at least three must cooperate to gain access
Application Example: The Grey Access Control System
[Bauer et al., ISC’05]
- Access control system for virtual and physical resources (e.g. doors)
- Deployed in CMU’s Collaborative Innovation Center
  - Approximately 40 Grey-capable doors and 30+ users
  - Could also log into workstations using Grey-compatible Windows XP and Linux login modules
  - Access-control module for web servers
- Used distributed theorem prover
  - System design led to new requirements
    - Interactive credential creation (to react to access requests)

Application Example: The Perspective File System
[Salmon, Schlosser, Cranor, Ganger, FAST’09]
- Distributed file system aimed at home storage management
  - Aimed at non-technical users
  - Semantic groups (tags)
  - Peer-to-peer architecture (no master storage server)
- Key concept: view (semantic query + device)
  - Files matching the query are stored on the device
  - Allows users to manipulate file replicas
    - Using expandable grid–like interface
  - Access control
    - To grant access, allow the requester to store a replica on his device
    - Done reactively
  - File replication also provides fault tolerance
    - System computes replication degree automatically
Review of Lecture

• What did we learn?
  – Usability challenges for access control lists
  – Potential solutions: expandable grid, semantic tags, logic-based access control
  – System examples: Grey, Perspective
  – Rules for interface design and for user studies

• Sources
  – Lorrie Cranor, Lujo Bauer and Michelle Mazurek

• What’s next?
  – Internet protocols and denial of service

• Deadline reminder
  – Homework 2 is due today