Formal mental models for inclusive privacy and security

Adam M. Houser
Matthew L. Bolton, Ph.D.

Department of Industrial and Systems Engineering

University at Buffalo The State University of New York

Presentation Outline

- HF engineering: why privacy and security?
- Folk models, mental models, and analysis with formal methods
- Brief conceptual demonstration with a use case and specifications
- Modified from presentation at SOUPS 2017 (a USENIX conference)

Why inclusive (digital) privacy and security?

- Inclusivity: solution suitability for different user groups
- Privacy: preventing undesired information disclosure
- Security: maintaining the integrity of a digital system

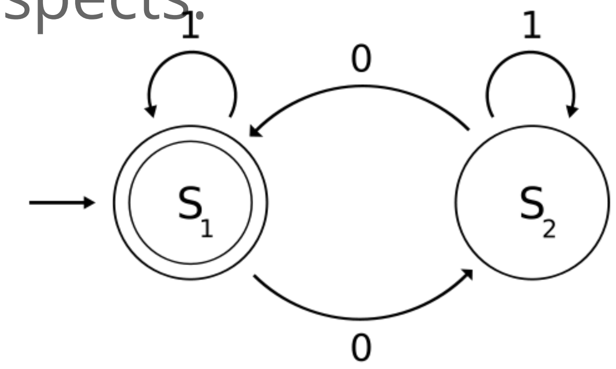
“So what?” → Are SV apps suitable for use cases abroad?

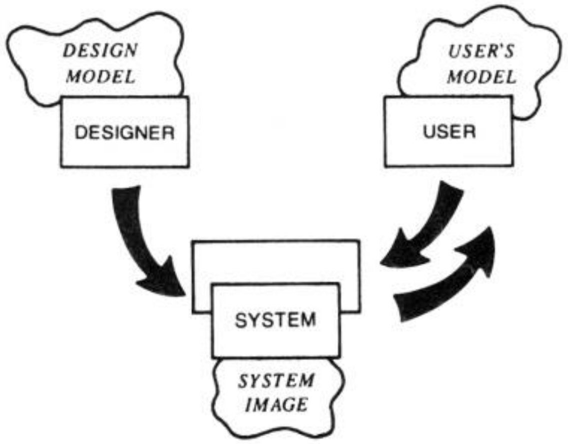
Mental models in human factors engineering

- Internalized representations of system functionality
- Different representational strategies:
  - “Pictures in the mind” (de Kleer & Brown, 1981)
  - Descriptive system abstractions (Rasmussen, 1971; Rouse & Hunt, 1986)
  - “Structured knowledge” (Dutton & Starbuck, 1971)
- Strategies are not mutually exclusive (Sanderson, 1990)

Mental models in human factors engineering

- For this work, Norman (1983) outlines key aspects:
  - “Runnability” of mental models
  - Agreement between the user’s model and the system image (Norman, 1986)





Folk models in cybersecurity

- Similarities and differences between folk and mental models
  - Description of user expectations about system behavior
  - Folk models rely more heavily on metaphor (Camp, 2009)
  - Mental models more heavily emphasize runnability

- Some work moving towards mental models (Blythe & Camp, 2012)

Folk models in cybersecurity

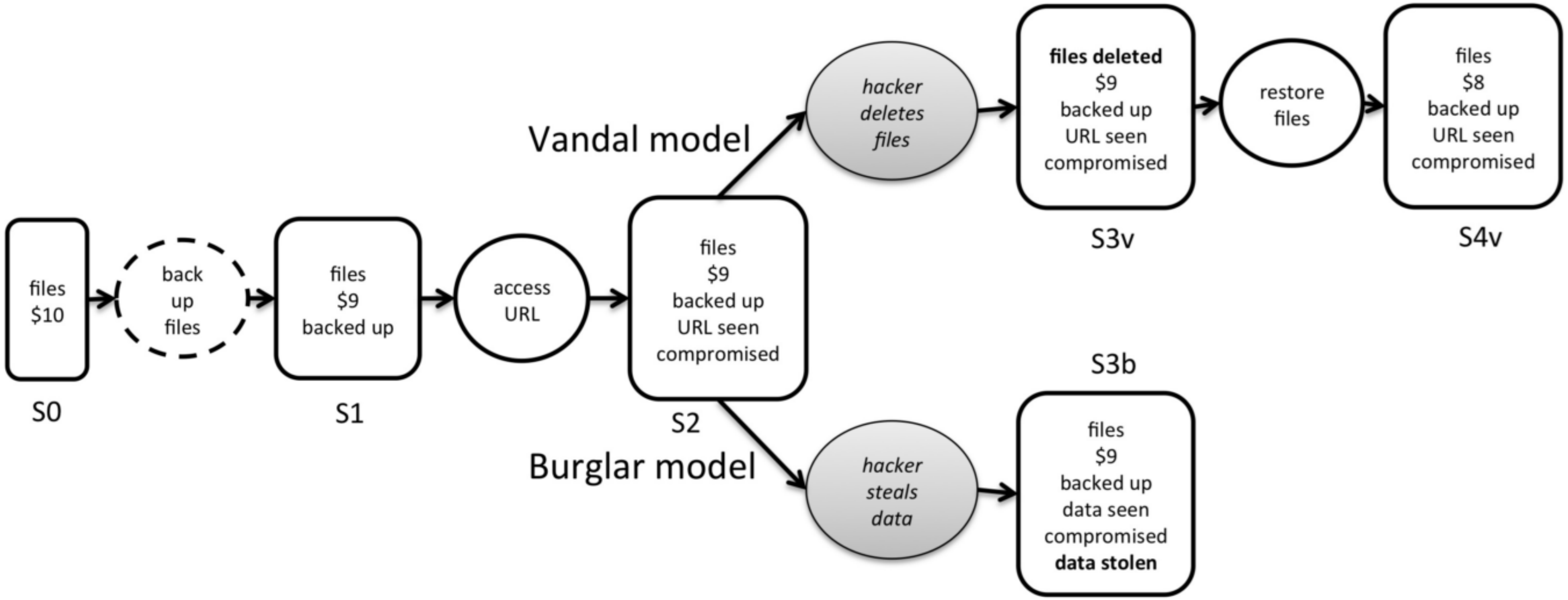
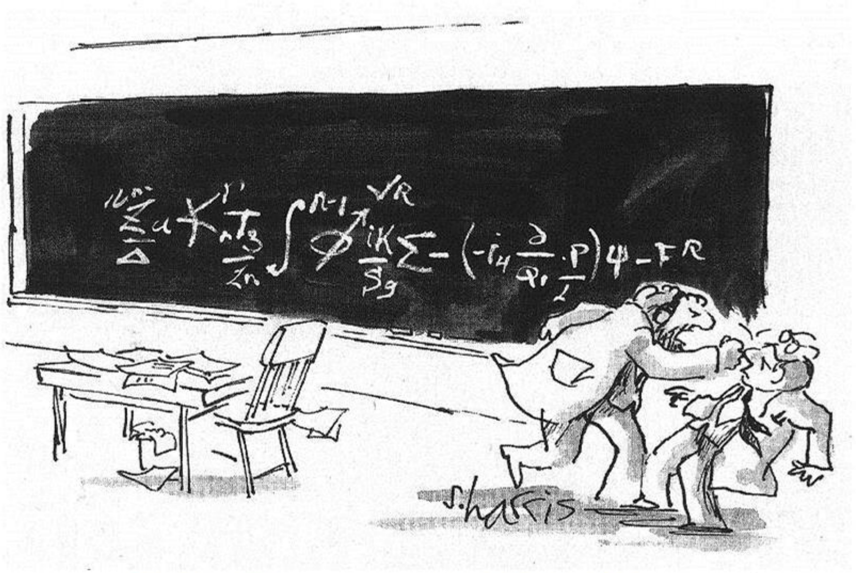


Figure 1. Simulation of a decision to “back up files” run against Wash (2010)’s vandal and burglar hacker models (Blythe & Camp, 2012, p. 89).

Mental model analysis with formal methods

- Well-defined mathematical languages and techniques for modeling, specifying, and verifying systems (Wing, 1990).

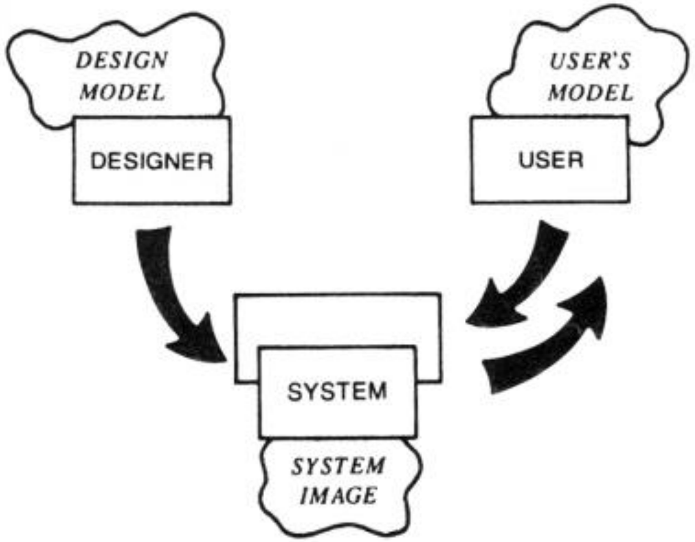
- Proofs
- Counterexamples
- Exhaustive search



YOU WANT PROOF? I'LL GIVE YOU PROOF!

Examples of analysis with formal methods

- Particular success with finding user-system mismatches
  - Aircraft autopilot (Degani & Heymann, 2002)
  - Aircraft autoland (Oishi, et al., 2002)
  - Vehicle cruise control (Degani, 2004)
- “Killer feature” is the discovery of unanticipated user-system mismatches through exhaustive statespace search



A research objective

By synergistically integrating work from human factors, cybersecurity, and formal methods, we can discover unanticipated interactions between user mental models and application features or behaviors.

This can help ensure that privacy and security solutions are appropriate, useful, and effective for a plurality of users.

Conceptual demonstration: encrypted chat

- Consider two user groups: domestic violence victims and protesters
- Threat models, environments for use, capability needs
  - Developer sensitivity to these issues?
- Extract, create formal representation of user mental models, system
- Create suite of specifications capturing “dangerous conditions”
- Does the chat app keep both users safe and secure? If not, why?

University at Buffalo The State University of New York

Questions?

Adam M. Houser

appliedcaffeine.org

adamhous@buffalo.edu

@neutrinos4all

Reserve Slides

Example specifications

- User expects app data to always be available
- Screenshots should always be allowed
- User expects to be always notified about communication network changes
- User should never be allowed to ‘upload’ or ‘share’ a private key

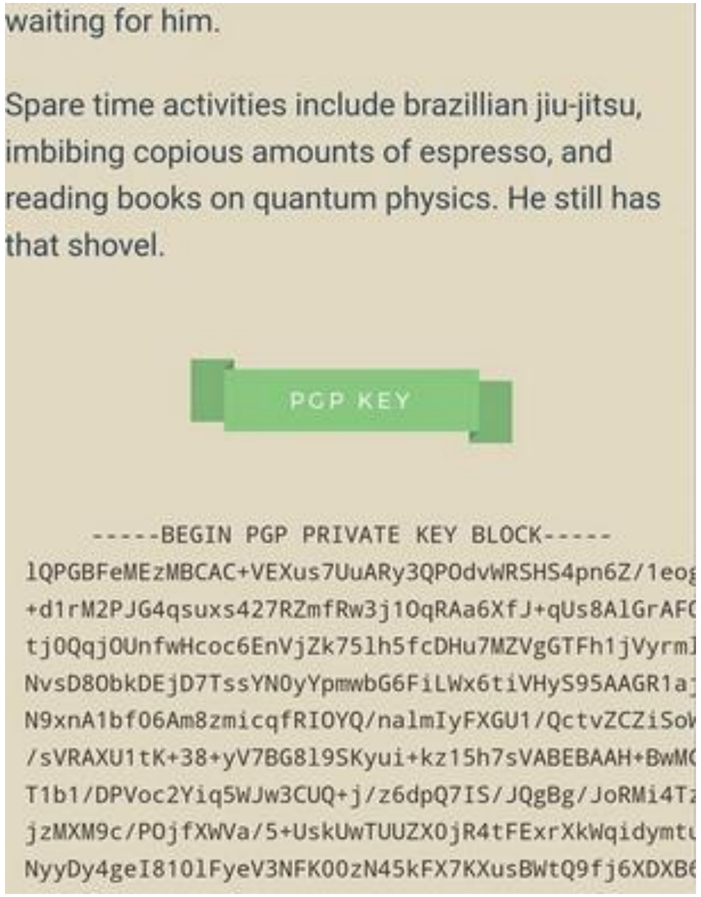
Mental model elicitation

- There exist a number of methods for model extraction
  - Card-sorting tasks (Asgharpour, et al., 2007)
  - Structured and semi-structured interviews (Wash, 2010)
  - Task observations (Dutton & Starbuck, 1971)
  - Cognitive walkthroughs (Ford & Sterman, 1997)
  - Training artifact analysis (Rushby, 2001)

Future work

- We plan to exercise this synergistic approach for the dissertation
- Elicit mental models for ‘attacker’ and ‘defender,’ perhaps with a simple program or program feature set
- Discover unanticipated interactions that defenders *think* are safe, but attackers can use to their advantage
- Potentially evaluate software countermeasures from user perspective

End-user key management is still hard



waiting for him.

Spare time activities include brazillian jiu-jitsu, imbibing copious amounts of espresso, and reading books on quantum physics. He still has that shovel.

```
     -----BEGIN PGP PRIVATE KEY BLOCK-----
 lQPGBFeMEzMBCAC+VEXus7UuARy3QPOdvWRSHS4pn6Z/1eog
 +d1rM2PJG4qsuxs427RZmfRw3j1OqRAa6XfJ+qUs8AlGrAF
 tj0QqjOUnfwHcoc6EnVjZk75lh5fcDHu7MZVgGTFh1jVyrm
 NvsD8ObkDEjD7TssYN0yYpmwbG6FiLWx6tiVHyS95AAGR1a
 N9xnA1bf06Am8zmicqfRIOYQ/nalmIyFXGU1/QctvZCZiSo
 /sVRAXU1tK+38+yV7BG8l9SKyui+kz15h7sVABEBAAH+BwM
 T1b1/DPVoc2Yiq5WJw3CUQ+j/z6dpQ7IS/JQgBg/JoRMi4T
 jzMXM9c/POjfXWVa/5+UskUwTUUZX0jR4tFExrXkWqidymt
 NyyDy4geI810lFyeV3NFK0OzN45kFX7KXusBWtQ9fj6XDXB
```

https://twitter.com/thesl3ep/status/876066176589336576