Usability and Security

Glenn Fink
finkga@vt.edu
http://csgrad.cs.vt.edu/~finkga/
Papers

♦ Reading List
  – Whitten & Tygar, Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0
  – Balfanz, et al., In Search of Usable Security: Five Lessons from the Field
  – Zurko & Simon, User-Centered Security
  – Yee, Aligning Security and Usability

♦ Other Papers
  – Good & Krekelberg, Usability and Privacy: A Study of Kazaa P2P File-Sharing
  – Adams & Sasse, Users are not the Enemy
Why Johnny Can’t Encrypt

♦ Intelligent users couldn’t figure out how to encrypt their e-mail within 90 minutes
♦ Usable security cannot be achieved via normal usability approaches
  – Neat organization and display are insufficient
  – Must evaluate security as a secondary user goal
  – Need a new usability standard
♦ Inherent UI problems for security software
  – Users are unmotivated by security and overly optimistic
  – Security is intrinsically abstract
  – System must figure out user’s intent
  – Security is rife with irreversible actions
  – Security is only as strong as the weakest link
In Search of Usable Security

♦ Usability is an essential element of secure systems
♦ Decentralization: Users must be empowered to make security decisions
♦ Five Lessons Learned:
  – You can’t retrofit usable security
  – Tools are building blocks, not solutions
  – Solve security problems at user-level in user terms
  – Conduct usability studies with normal users
  – Tailor the solution to the local problem
User-Centered Security

♦ Goal: Make usability a primary goal of security

♦ Traditional problems with security usability
  – Secure systems were notoriously unusable
  – Underlying mathematical models do not guarantee a system that models user intentions well
  – Least privilege is hard to implement in a user-friendly way

♦ Three possible approaches to usable security:
  – Apply usability to existing secure systems
  – Apply security to existing usable systems
  – User-centered security design (from the start)
Aligning Security and Usability

- Develop usability and security goals together in an iterative design
- Infer program authorization from user’s actions by paying attention to indications of his intent
- Security by Admonition or Designation:
  - **Admonition**: Start with full user authority and require explicit user authorization when a user requests a risky action
  - **Designation**: Start with minimal authority and take users actions as indication that the software has his authorization
- Security by designation may require finer-grained access controls but is often achievable transparently
Kazaa Usability and Privacy


♦ Kazaa is a P2P file-sharing application

♦ Problems noted:
  – Not clear what files are being shared
    • More than just the “My Shared Folder” contents
  – Not clear how to start and stop sharing
    • Misleading indications about sharing status
  – Easy to unintentionally share private files
  – Lack of confidence in correct operation

♦ EULA Problems:
  – Unwittingly become part of a grid computer!
Users are not the Enemy


♦ Security policy may be its own worst enemy
  – Myth: Users are always against security
  – Fact: Users sometimes subvert intrusive security measures
    • Ex: Having multiple, different, complex passwords
    • Ex: Being forced to change passwords too often

♦ Communication with users is required
  – Users sometimes do not understand security
    • But this is often the fault security personnel
  – Security personnel often have not taken the time to understand the users’ work context
Goals of Usable Security

♦ Ultimate Goal: Safe, functional systems
  – Usability: an essential element of secure systems
  – Security: contributes to safety, predictability, and thus Usability

♦ Intermediate Goal: Raise the awareness of future designers about:
  – The difficulty of the problem
  – The criticality of finding solutions
Problems with Security Usability

♦ **Usability itself is hard**
  - HCI is really mediated HHI
    - Human interactions present intractable problems
  - Computers must support users’ intentions

♦ **Secure usability is harder than normal HCI**
  - There is an active, intelligent enemy
  - The threat is often invisible to users
  - Underlying security mechanisms are often obscure
  - Usability and Security can appear to be at odds
    - Security seen as a way to watch and restrict users rather than to help them with their tasks
An Active, Intelligent Enemy

♦ Rate of new attacks continues to increase
  – Worms, virii, phishing, etc.
♦ Number of attackers growing rapidly
  – Hacking tools built for reuse
♦ Most every security tool can be used offensively as well as defensively
♦ Builders’ crisis of imagination w.r.t. security
An Invisible Threat

- Users cannot see suspicious activity of their machines unless it interferes with usability
  - Surreptitious network connections
  - Accesses to system address book
  - Suspicious kernel routine calls
  - New software installed in system areas

- Users prefer to believe that the system security is working and protecting them

- Users do not believe they will be attacked until it is too late
An Obscure Protection

- Protection mechanisms are based on mathematical models
  - Access calculus, Bell and Lapadula model
  - Inaccessible to average users
  - Often designed for military use
  - Often not designed for user’s needs or intentions

- Security by Admonition
  - Too many false alarms
  - Makes users repeat their intentions
  - Misdiagnoses user intentions
A Conflict of Interests?

- Legacy security systems are unusable
- Legacy user applications are insecure
- Hackers pay more attention to the human side than the machine side
- Users will try to circumvent onerous security
Costs of Unusable Security

♦ Danger of serious user error
  – Loss of privacy
  – Potential for fraud
  – Lost revenue
  – Lost productivity

♦ Danger of serious program error
  – Unexpected program behavior
  – Irreversible actions
Approaches to Usable Security

- Respect the complexity of the problem
- Build security and usability in from the start
  - Conduct formative usability evaluations
- Build them together in an iterative design
- Build security around the users’ needs
  - Solve the problem in the users’ terms and at their levels of abstraction
Respect the Complexity

♦ Realize that users have complex needs
♦ Software is seldom used as designed
♦ Successful software will be used in unpredictable ways
♦ Attackers will use the software in perverse ways
Right from the Start

♦ Realize that the user is not the enemy
  – Keep the user informed
  – Avoid an authoritarian mindset
  – Tap into the user’s desire to be secure

♦ Tailor security to work practices
  – Study the user: Contextual Design
  – Avoid cookbook approaches to security
  – Challenge textbook security thinking
Usability Evaluations

♦ Conduct “Discount” Usability Evaluations
  – User testing with low-fidelity prototypes
  – Heuristic Evaluation
  – Cognitive Walkthrough

♦ Conduct Full Laboratory Evaluations
  – Use tasks that are representative of common user activities

♦ Conduct Contextual Inquiry
Ham in Hand

- Build usability and security together
  - Not as separate, competing features
    - Requires communication on the software team
    - Implies iterative design
- Consider both as part of the software engineering process
User-Centered

- Design security to meet users’ needs
  - But first you must find out what these are
- Employ lessons learned from the literature
  - Common security pitfalls
  - Typical usability failures
- Use Security by Designation where possible
Conclusions

♦ Security is not intrinsically antithetical to usability
♦ Usability and security are both required
  – By users’ desires and needs
  – By good engineering practice
♦ Security and usability are both hard problems to solve
  – To reap the benefits, be prepared to pay the costs