Usability Studies

Email Encryption
Usable Security – CS 6204 – Fall, 2009 – Dennis Kafura – Virginia Tech

- Defines a standard of usable security
- Evaluated PGP 5.0 Using
  - Direct evaluation (cognitive walkthrough)
  - User experiments
- Conclusions:
  - PGP 5.0 does not meet the usability security standard
  - Confirms hypothesis that “security-specific user interface design principles and techniques are needed.”
Usable Security standard

Definition: Security software is usable if the people who are expected to use it:

1. are reliably made aware of the security tasks they need to perform;
2. are able to figure out how to successfully perform those tasks;
3. don’t make dangerous errors; and
4. are sufficiently comfortable with the interface to continue using it.
Design differences

Designing for security has unique challenges that must be accounted for in designing for usability:

- Unmotivated user (security is a secondary goal)
- Abstraction (policies/rules are unintuitive to general population)
- Feedback (security state is complex and difficult to depict)
- “barn door” (cannot make serious mistakes)
- “weakest link” (must attend to all aspects of security; cannot learn/manage incrementally as with other software)
Walkthrough evaluation

- Metaphor issues
  - Keys: a different key is used for encryption than for decryption unlike a single “real” key which does both
  - Signature: not clear that signing (quill pen icon) requires the use of the private key
  - Key types: distinction between RSA keys (blue) and Diffie-Hellman (brass) not clear
Walkthrough evaluation

- Key management issues
  - Key server:
    - no top-level visibility;
    - not identified as a remote operation;
    - no history of access
  - Key rating:
    - Validity (completely, marginally, invalid) – degree of confidence that key belongs to given user
    - Trust (completely, marginally, untrusted) – degree of confidence in another user as certifier of keys
    - Assigned automatically

- Problems
  - User may assign now meaning to “validity” and “trust”
  - Automatic assignment not visible
Walkthrough evaluation

- Reversibility:
  - Insufficient notice (e.g., deleting the private key)
  - Insufficient guidance on what actions are need to undo the effects of an otherwise irreversible operation (accidental key revocation)

- Consistency (terminology: “encode” vs. “encrypt”)

- Too much information (does not separate information relevant to novice vs. advanced users)
User experiment

- Task: send sensitive political campaign information via encrypted email to five others.
- Participants: 12, email proficiency, security novices
- Results:
  - only 1/3rd of the subjects were able to complete the task in 90 minutes
  - 1/4th of the subjects accidentally exposed the sensitive information
  - Subjects’ difficulties stemmed from inadequate understanding of the public-key model
Email encryption redux

- Repeats Whitten/Tygar experiments with 43 crypto-naïve users
- Uses newer systems (S/MIME) in combination with Key Continuity Management (KCM)

Claims/Results:
- Less secure (in principle) but more usable (in practice)
- Better interfaces needed for a specific situation
S/MIME and KCM

S/MIME
- Automatically attaches certificate (with public key) of user whose private key encrypted (signs) outgoing email
- Automatically decrypts received email which has an attached certificate and stores certificate in address book
- Obtaining a certificate is still difficult (requires trusted third party, certificate chains)

KCM
- Ignore certificate chains ("users are on their own"); directly associate identity in certificate with public key in certificate for email purposes
- Notify user if public key changes for that identity
- Tradeoff: less secure but more usable and scalable
- Added to Eudora mail client via CoPilot
Attacks and Feedback

Anticipated attacks
- New key attack (trust email which has a key different from one seen previously?)
- New identity attack (trust new key and new identity)?
- Unsigned message attack (trust unsigned message from known source)

Feedback (message border color-coded)
- Red (message contains new key from known identity)
- Yellow (first signed message from identity)
- Green (current message signed with known key)
- Gray (unsigned message from identity with known key)
- White (unsigned message from unknown identity)
Experiment

- Three groups
  - No KCM
  - KCM
  - KCM + briefing

- Results
  - KCM help users to resist attacks
  - Users were able to explain signing and sealing in follow-up interviews, however…
  - KCM users are less likely to encrypt message than those without KCM (apparently not understanding the difference between sealing and signing despite results of interviews)

<table>
<thead>
<tr>
<th>Cohort</th>
<th>n</th>
<th>% subjects resisting attacks sometimes</th>
<th>% subjects always</th>
<th>Clicked “encrypt” to seal email sometimes</th>
<th>Clicked “encrypt” always</th>
</tr>
</thead>
<tbody>
<tr>
<td>No KCM</td>
<td>14</td>
<td>43%</td>
<td>0%</td>
<td>50%</td>
<td>21%</td>
</tr>
<tr>
<td>Color</td>
<td>14</td>
<td>50%</td>
<td>29%</td>
<td>36%</td>
<td>36%</td>
</tr>
<tr>
<td>Color + Briefing</td>
<td>15</td>
<td>87%</td>
<td>33%</td>
<td>20%</td>
<td>13%</td>
</tr>
</tbody>
</table>

\[
\chi^2 = 6.13, \quad p = 0.013
\]

\[
p = 0.57, \quad p = 0.087
\]

Table 3: Summary Results of Johnny 2 User Study
## Attack Types

<table>
<thead>
<tr>
<th>Group</th>
<th>Maria 1</th>
<th>Maria 2</th>
<th>Ben</th>
<th>New key attack</th>
<th>New identity attack</th>
<th>Unsigned message attack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No KCM</strong></td>
<td>100%</td>
<td>92%</td>
<td>100%</td>
<td>71%</td>
<td>79%</td>
<td>75%</td>
</tr>
<tr>
<td></td>
<td>(14/14)</td>
<td>(11/12)</td>
<td>(14/14)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Color</strong></td>
<td>93%</td>
<td>100%</td>
<td>92%</td>
<td>64%</td>
<td>50%</td>
<td>58%</td>
</tr>
<tr>
<td></td>
<td>(13/14)</td>
<td>(13/13)</td>
<td>(11/12)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Color+Briefing</strong></td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>13%</td>
<td>60%</td>
<td>43%</td>
</tr>
<tr>
<td></td>
<td>(13/15)</td>
<td>(14/15)</td>
<td>(13/14)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\chi^2$</th>
<th>2.20</th>
<th>0.018</th>
<th>0.79</th>
<th>10.61</th>
<th>1.02</th>
<th>3.98</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>0.14</td>
<td>0.89</td>
<td>0.37</td>
<td>0.001</td>
<td>0.31</td>
<td>0.046</td>
</tr>
</tbody>
</table>

- KCM more successful against new key attack and unsigned message attacks
- KCM not more successful against new identity attack
Reflections

- To what extent is “technology” the answer? (What is the difference between the user performance in the two experiments?)
- Does usability engineering for security require a different set of methods/tools?
- Is a tradeoff between security and usability required (as suggested in the use of KCM)?
- Importance of repeatability.
- Utility of an experimental framework (the “Johnny2 Experimenter’s Workbench”).