CSE 484 / CSE M 584: Computer Security and Privacy

Usable Security

Autumn 2020

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, Ada Lerner, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Admin

- **Lab 2** due *today*
- **Homework 3** out, due Dec 4
- There will be a **Lab 3** (it is easier than 1+2)
  - Smart home security, more in section next week
- **No class** Wednesday or Friday this week (Thanksgiving)
Importance of Usability in Security

• Why is usability important?
  – People are the critical element of any computer system
    • People are the reason computers exist in the first place
  – Even if it is possible for a system to protect against an adversary, people may use the system in other, less secure ways
Usable Security Roadmap

• 3 case studies
  – HTTPS indicators + SSL warnings
  – Phishing
  – Password managers

• Step back: root causes of usability problems, and how to address
Case Study #1: Browser HTTPS Indicators

- **Design question 1:** How to indicate encrypted connections to users?
- **Design question 2:** How to alert the user if a site’s SSL certificate is untrusted?
  - You discussed this in section a couple weeks ago
The Lock Icon

• Goal: identify secure connection
  – SSL/TLS is being used between client and server to protect against active network attacker

• Lock icon should only be shown when the page is secure against network attacker
  – Semantics subtle and not widely understood by users
  – Whose certificate is it??
  – Problem in user interface design
Will You Notice?

Clever favicon inserted by network attacker

browser UI = "Chrome"
Do These Indicators Help? (2007)

• “The Emperor’s New Security Indicators”

<table>
<thead>
<tr>
<th>Score</th>
<th>First chose not to enter password...</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1 ∪ 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>upon noticing HTTPS absent</td>
<td>0 0%</td>
<td>0 0%</td>
<td>0 0%</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>1</td>
<td>after site-authentication image removed</td>
<td>0 0%</td>
<td>0 0%</td>
<td>2 9%</td>
<td>0 0%</td>
<td>2 4%</td>
</tr>
<tr>
<td>2</td>
<td>after warning page</td>
<td>8 47%</td>
<td>5 29%</td>
<td>12 55%</td>
<td>13 37%</td>
<td>25 44%</td>
</tr>
<tr>
<td>3</td>
<td>never (always logged in)</td>
<td>10 53%</td>
<td>12 71%</td>
<td>8 36%</td>
<td>22 63%</td>
<td>30 53%</td>
</tr>
</tbody>
</table>

**Total** | 18 | 17 | 22 | 35 | 57

**Lesson:**

Users don’t notice the **absence** of indicators!
Newer Versions of Chrome

c. 2017

Secure https://mail.google.com/mail/u/0/#inbox

2020

mail.google.com/mail/u/0/#inbox

Not Secure http-password.badssl.com
Case Study #1: Browser HTTPS Indicators

- **Design question 1:** How to indicate encrypted connections to users?
- **Design question 2:** How to alert the user if a site’s SSL certificate is untrusted?
  - You discussed this in section a couple weeks ago
  - Recall: Opinionated design
Challenge: Meaningful Warnings

See current designs for different conditions at https://badssl.com/.
Case Study #2: Phishing

• **Design question:** How do you help users avoid falling for phishing sites?
A Typical Phishing Page

Weird URL
http instead of https
Safe to Type Your Password?

*extended validation*
Safe to Type Your Password?

![Phishing Page](image.png)
Safe to Type Your Password?
Safe to Type Your Password?

“Picture-in-picture attacks”

Trained users are more likely to fall victim to this!
Phishing Warnings (2008)

Passive (IE)

Active (IE)

Active (Firefox)
Active vs. Passive Warnings

- Active warnings significantly more effective
  - Passive (IE): 100% clicked, 90% phished
  - Active (IE): 95% clicked, 45% phished
  - Active (Firefox): 100% clicked, 0% phished
If you don’t recognize your personalized “SiteKey”, don’t enter your Passcode.
Case Study #3: Password Managers

- **Password managers** handle creating and “remembering” strong passwords

- Potentially:
  - Easier for users
  - More secure

- Early examples:
  - PwdHash (Usenix Security 2005)
  - Password Multiplier (WWW 2005)
PwdHash

@@ in front of passwords to protect; or F2

sitePwd = Hash(password, domain)

Prevent phishing attacks

Password Multiplier

Activate with Alt-P or double-click

sitePwd = Hash(username, password, domain)

Both solutions target simplicity and transparency.
Usability Testing

• Are these programs usable? If not, what are the problems?

• Approaches for evaluating usability:
  – Usability inspection (no users)
    • Cognitive walkthroughs
    • Heuristic evaluation
  – User study
    • Controlled experiments
    • Real usage
## Task Completion Results

<table>
<thead>
<tr>
<th>Success</th>
<th>Potentially Causing Security Exposures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dangerous Success</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PwdHash</td>
<td></td>
</tr>
<tr>
<td>Log In</td>
<td>48%</td>
</tr>
<tr>
<td>Migrate Pwd</td>
<td>42%</td>
</tr>
<tr>
<td>Remote Login</td>
<td>27%</td>
</tr>
<tr>
<td>Update Pwd</td>
<td>19%</td>
</tr>
<tr>
<td>Second Login</td>
<td>52%</td>
</tr>
</tbody>
</table>

### Password Multiplier

<table>
<thead>
<tr>
<th>Success</th>
<th>Potentially Causing Security Exposures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Log In</td>
<td>48%</td>
</tr>
<tr>
<td>Migrate Pwd</td>
<td>16%</td>
</tr>
<tr>
<td>Remote Login</td>
<td>N/A</td>
</tr>
<tr>
<td>Update Pwd</td>
<td>16%</td>
</tr>
<tr>
<td>Second Login</td>
<td>16%</td>
</tr>
</tbody>
</table>
Problem: Mental Model

- Users seemed to have misaligned mental models
  - Not understand that one needs to put “@@” before each password to be protected.
  - Think different passwords generated for each session.
  - Think successful when were not.
  - Not know to click in field before Alt-P.
  - Don’t understand what’s happening: “Really, I don’t see how my password is safer because of two @’s in front”
Problem: Transparency

• Unclear to users whether actions successful or not.
  – Should be obvious when plugin activated.
  – Should be obvious when password protected.
• Users feel that they should be able to know their own password.
Problem: Dangerous Errors

• Tendency to try all passwords
  – A poor security choice – phishing site could collect many passwords!
  – May make the use of PwdHash or Password Multiplier worse than not using any password manager.

• Usability problem leads to security vulnerabilities.
  – Theme in course: sometimes things designed to increase security can also increase other risks
Root Causes? How to Improve?

- You aren’t your user
- What assumptions?
- Security vs. convenience
- People want things to “just work”
- Design challenge
- Limited threat models
- Accessibility / not all users are the same
Stepping Back: Root Causes?

• Computer systems are complex; users lack intuition
• Users in charge of managing own devices
  – Unlike other complex systems, like healthcare or cars.
• Hard to gauge risks
  – “It won’t happen to me!”
• Annoying, awkward, difficult
• Social issues
  – Send encrypted emails about lunch?...
How to Improve?

• Security education and training
• Help users build accurate mental models
• Make security invisible
• Make security the least-resistance path
• …?
Beyond Specific Tools: Different User Groups

• Not all users are the same!
• Designing for one group of users, or “generic” users, may lead to dangerous failures or reasons that people will not use security tools
• Examples from (qualitative) research at UW:
  – Journalists (most sources are not like Snowden!)
  – Refugees in US (security measures may embed US cultural assumptions!)