Announcements

• Lab 2 and Homework 3 are ongoing
• If something seems off, please don’t hesitate to submit regrade requests.
• Friday: Mobile platform security
• No class next Wednesday or Friday
  – Happy 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 last week
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
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</td>
<td>0%</td>
<td>0</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>after site-authentication image removed</td>
<td>0</td>
<td>0%</td>
<td>0</td>
<td>2</td>
<td>9%</td>
</tr>
<tr>
<td>2</td>
<td>after warning page</td>
<td>8</td>
<td>47%</td>
<td>5</td>
<td>29%</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>never (always logged in)</td>
<td>10</td>
<td>53%</td>
<td>12</td>
<td>71%</td>
<td>8</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>18</td>
<td>17</td>
<td>22</td>
<td>35</td>
<td>57</td>
</tr>
</tbody>
</table>

**Lesson:**

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

c. 2017

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

2023

[Secure] mail.google.com/mail/u/1/#inbox

[Not Secure] https://revoked.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 last week
  – 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?
Safe to Type Your Password?

Bank of the West

Gives me your password!

User name:

Password:

Login
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)
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
Another Idea: Site Authentication Image

If you don’t recognize your personalized “SiteKey”, don’t enter your Passcode

But... users don’t notice the absence of indicators!
Modern Anti-Phishing

- Largely driven by Google Safe Browsing
  - Browser sends 32-bit prefix of hash(url)
  - API says: good or bad
Case Study #3: Password Managers

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

• Potentially:
  – **Easier** for users
  – **More secure**

• Early examples with some usable security lessons:
  – PwdHash (*Usenix Security 2005*)
  – Password Multiplier (*WWW 2005*)

Note: The goal of these case studies is not really about these specific (now very dated) tools, but to show you the process and lessons (see also HW3!).
PwdHash

@@ in front of passwords to protect; or F2

\[\text{sitePwd} = \text{Hash}(\text{pwd}, \text{domain})\]

Password Multiplier

Activate with Alt-P or double-click

\[\text{sitePwd} = \text{Hash}(\text{username, pwd, domain})\]

Prevent phishing attacks

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>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dangerous Success</td>
<td>False Completion</td>
<td>Failed</td>
<td>Failed due to Previous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log In</td>
<td>48%</td>
<td>44%</td>
<td>8%</td>
<td>0%</td>
<td>N/A</td>
</tr>
<tr>
<td>Migrate Pwd</td>
<td>42%</td>
<td>35%</td>
<td>11%</td>
<td>11%</td>
<td>N/A</td>
</tr>
<tr>
<td>Remote Login</td>
<td>27%</td>
<td>42%</td>
<td>31%</td>
<td>0%</td>
<td>N/A</td>
</tr>
<tr>
<td>Update Pwd</td>
<td>19%</td>
<td>65%</td>
<td>8%</td>
<td>8%</td>
<td>N/A</td>
</tr>
<tr>
<td>Second Login</td>
<td>52%</td>
<td>28%</td>
<td>4%</td>
<td>0%</td>
<td>16%</td>
</tr>
</tbody>
</table>

### PwdHash

| Log In  | 48%          | 44%                      | 8%       | 0%                   | N/A |
| Migrate Pwd | 42%        | 35%                      | 11%      | 11%                  | N/A |
| Remote Login | 27%        | 42%                      | 31%      | 0%                   | N/A |
| Update Pwd | 19%         | 65%                      | 8%       | 8%                   | N/A |
| Second Login | 52%        | 28%                      | 4%       | 0%                   | 16% |

### Password Multiplier

| Log In  | 48%          | 44%                      | 8%       | 0%                   | N/A |
| Migrate Pwd | 16%        | 32%                      | 28%      | 20%                  | N/A |
| Remote Login | N/A        | N/A                      | N/A      | N/A                  | N/A |
| Update Pwd | 16%         | 4%                       | 44%      | 28%                  | N/A |
| Second Login | 16%        | 4%                       | 16%      | 0%                   | 16% |
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?
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
- ...?
Closing Thought: 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!)