Client-side Defenses for Context-Aware Phishing and Transaction Generator Spyware

Collin Jackson
Dan Boneh       John Mitchell
Stanford University
Web Threats

- **Phishing**
  - Spoof website convinces user to log in

- **Common password problem**
  - Same password for different sites

- **Keylogger spyware**
  - Malicious software observes login

- **Transaction generator spyware**
  - Hijacks login session, sends requests
Web Threats

- **Phishing**
  - SafeHistory
  - SpoofGuard
  - SafeCache

- **Common password problem**
  - PwdHash

- **Keylogger spyware**
  - SpyBlock (no server changes)

- **Transaction generator spyware**
  - SpyBlock (with server changes)
Phishing Problem

- User has existing relationship with target site
- User cannot reliably identify fake site
- Captured password can be used at target site
New Phishing Sites by Month December '04 - December '05

Anti-Phishing Working Group: antiphishing.org
Context-aware phishing

◆ **Bank of America customers see:**
  - “Please log in to your Bank of America account”

◆ **Wells Fargo customers see:**
  - “Please log in to your Wells Fargo account”

◆ **Works in all major browsers, Outlook 2002**
Protecting Browser State

C. Jackson, A. Bortz, D. Boneh, J. C. Mitchell (WWW ’06)

- Snooping violates same-origin principle:
  Only the site that stores some information in the browser may later read or modify that information.

- Stylesheets applied to hyperlinks
  SafeHistory narrows policy to safe cases

- Javascript cache timing techniques
  SafeCache partitions cache for safety
Common Password Problem

Phishing attack or break-in at site B reveals pwd at A

- Server-side solutions will not keep pwd safe
- **Solution**: Strengthen with client-side support
Generate a unique password per site

- \( \text{HMAC}_{\text{fido:123}}(\text{banka.com}) \Rightarrow Q7a+0ekEXb \)
- \( \text{HMAC}_{\text{fido:123}}(\text{siteb.com}) \Rightarrow OzX2+1Ciqc \)

- Hashed password is not usable at target site
User Interface Spoofing

- Attacker can display password fields or dialogs:

- Password is sent to attacker in the clear
Trusted Password Interfaces

- Password prefix
  - PwdHash

- Secure attention sequence

- Isolated screen area

- Trusted image or phrase
  - Passmark
  - SpyBlock

Starts with @@
Keylogger Spyware Problem

- Attacker observes login on local machine
- Password is sent to attacker for later use
- Screenshot can observe “screen keyboards”
Transaction Generator Problem

- Why bother with passwords?
- Once user is logged in, attacker can:
  - Corrupt user requests
  - Issue unauthorized requests
SpyBlock

C. Jackson, D. Boneh, J. C. Mitchell

- Isolated component for authentication
- Untrusted environment for user apps
# Authentication modes

<table>
<thead>
<tr>
<th>Authentication</th>
<th>Threat</th>
<th>Common Password</th>
<th>Phishing</th>
<th>Spyware keylogger</th>
<th>Network password sniffing</th>
<th>Network cookie sniffing</th>
<th>Pharming</th>
<th>Malware session hijacking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password hashing</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password injection</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hashing and injection</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Strong Pwd Auth (PAKE)</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Transaction Confirmation</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Hashing, injection require no server assistance
- Server support for additional protection
Password injection

- Intercept outbound requests and insert password
- Check for password fields in HTML to deter reflection
Strong password authentication

- Application environment does not learn user password
- HTTPS is verified to prevent network man-in-the-middle
Transaction confirmation

- Application environment cannot MAC fake transaction
- Unique transaction ID prevents replay attacks
Project websites

- Phishing
  - www.safehistory.com
  - www.safecache.com

- Common password problem
  - www.pwdhash.com

- Keylogger spyware

- Transaction generator spyware
  - www.getspyblock.com
Browser Access Control

- Noncooperative: Same origin policy
- Semicooperative: Third party blocking policy
- Cooperative: ???
Why use Password Prefix?

- Protection mechanism “built in” to password
- Does not rely on user to make a decision
- Same prefix works for everyone
- Distinguishes secure passwords from
  - normal passwords
  - social security numbers
  - PINs
- Only use it when you want to
Why use Password Prefix?

- Protection mechanism “built in” to password
- Does not rely on user to make a decision
- Same prefix works for everyone
- Distinguishes secure passwords from
  - normal passwords
  - social security numbers
  - PINs
- Only use it when you want to
Other Challenges

- Password Reset
- Internet Cafes
- Dictionary Attacks
- Spyware, DNS poisoning (no protection)
- Other issues (described in the paper)
  - Choosing salt for hash
  - Encoding hashed password
  - Additional attacks and defenses
After install, PwdHash can’t protect existing pwds
• Only passwords starting with @@ are secure
• User can choose where to use PwdHash
• User must enter old password unhashed into password reset page

Pwd Prefix makes it easy
• Old passwords won’t be accidentally hashed
• New, secure passwords are automatically hashed

Password Reset

Old password: *******
New password: *******
Re-enter password: *******

Save Changes

Starts with @@
Internet Cafes

- Users cannot install software at Internet Cafes.
- Would not be a problem if PwdHash were universally available
- **Interim solution**: A secure web site for remote hashing, e.g.
  
  https://www.pwdhash.com

- Hash is computed using JavaScript
  - Server never sees password
  - Resulting hash is copied into clipboard
  - Can also be used as a standalone password generator
Dictionary attacks

- After phishing attack or break-in to low security site, attacker can repeatedly guess password and check hash.
  - Succeeds on ≈15% of passwords (unlike 100% today)
  - Less effective on longer, stronger passwords
- Solution: better authentication protocol (SPEKE, SRP, etc.)
  - Requires server-side changes
- Defense: user specifies a global pwd to strengthen all pwd hashes
  - Creates a new pwd management problem for shared machines
- Defense: slow hash function (Halderman, Waters, Felten '05)
  - Increases time of dictionary attack