

Stronger Password Authentication Using Browser Extensions

Presented at 14th USENIX Security Symposium,
July 31 – August 5, 2005. Baltimore, MD

Blake Ross, Collin Jackson, Nick Miyake, Dan
Boneh, John Mitchell (Stanford University)

By John Leggio
Tuesday, October 17, 2006
ECE 1776

Objectives

- Enhance web password security authentication.
- Provide transparent method for customizing “per site passwords”
- No server changes & little or no change to user experience
- Want to reduce password attack threat

Result: PwdHash

- Allows users to invisibly generate site specific passwords
- Lightweight browser extension
 - Invisible to server
 - Invisible to user
- Helps stop password theft
- Easy to use

How PwdHash Works

- Generates unique per site passwords
- Uses password, domain name of site (the “salt”) & Pseudo Random Function (PRF):
$$\text{hash}(\text{password}, \text{domain}) = \text{PRF}_{\text{password}}(\text{domain})$$
- Implemented with Firefox extension & IE Browser Helper Object
- Uses the MD5 cryptographic hash function

How PwdHash Works (Continued)

- Uses 2 different approaches to validate user experience:
 - password-prefix & password-key
- Automatically generates per site passwords by prefixing password with @ @ or pressing F2 key
- Visual enhancement identifies when password mode is on



PwdHash In Use:

The screenshot shows the PayPal homepage with a warning dialog box in the foreground. The dialog box has a purple header with the text "PwdHash Warning" and a close button (X) in the top right corner. The main content of the dialog is on a grey background and contains a yellow warning triangle icon followed by the text: "You typed the PwdHash password prefix, but you are not currently in a password field that starts with the password prefix. It is possible, though unlikely, that the site trying to steal your password. Do not enter your PwdHash password into this page." At the bottom of the dialog is an "OK" button.

The background shows the PayPal website interface, including the logo, navigation menu (Welcome, Send Money, Request Money, Merchant Services, Auction Tools), and a Member Log-In section with fields for Email Address and Password. Other visible elements include a "Join PayPal Today" banner, a "Shop Without Sharing" banner, and a "Limited-time offers" section featuring a Netgear Router for \$39.95.

Why Use PwdHash?

- Most commercial websites have weak form of password authentication
- Combats password phishing problem
- Protection mechanism “built in” to per site password
- User decision not needed
- Same prefix works for everyone
- Distinguishes secure passwords from normal passwords & PINs

Challenges

- Password Reset
- Public computers
- Dictionary Attacks
- Spyware, DNS poisoning (no protection)
- Encoding hashed password
- Additional attacks and defenses

(Source: <http://crypto.stanford.edu/PwdHash/>)

Pros

Paper

- Problem thoroughly defined & analyzed
- Contains useful user study of PwdHash
- Related work that complements PwdHash
- Javascript attacks well defined

Implementation

- Usable & unobtrusive technique
- Good for phishing scams
- Simple & elegant solution
- Client-side security solution
- Useful tool

Cons

- Focus of the paper was implementation challenges
- Domain name changes?
- How it would manage access to a webpage using the IP address?
- Roaming
- Not sure if user study is generalizable.
- User study could have been defined better
- Traffic light maybe spoofed

Cons

- How it would handle length of password at different sites?
- People need to first reset their passwords.
- Having a remote website doing hashing for you is troublesome and insecure.
- Interactive applications using Ajax may communicate to the server the password character by character.
- Users may be feel uncomfortable with not knowing actual passwd

Cons

- Details of the implementation of the config file have not been revealed
- Inconvenience of using different passwords at sister websites like gmail.com and google.com
- Expecting the user to always type @@ before typing the password. It might even appear in the keystream
- User experience study was shallow as presumably a lot of learning and annoyance would be introduced as a result of '@@', change of length of password, roaming, password reset.

Related Work

- Brostoff, S. & Sasse, A. “Are Passfaces More Usable Than Passwords? A Field Trial Investigation”. People and Computers XIV - Usability or Else! Proceedings of HCI 2000, Sunderland University, 2000.
- Halderman, J. A., Waters, B., & Felten, E. W. “A Convenient Method for Securely Managing Passwords.” In Proc. 14th International World-Wide Web Conference, 2005.
- Juels, A., Jakobsson, M., & Stamm, S. “Active Cookies for Browser Authentication”. 2006.
- Yee, Ka-Ping & Sitaker, Kragen. “Passpet: convenient password management and phishing protection”. Proceedings of the second symposium on Usable privacy and security, p. 32 - 43, July 12-14, 2006, Pittsburgh, PA