Web Password Hashing

Description Publications Download Challenges Project Staff

Description

The Common Password Problem. Users tend to use a single password at many different web sites. By now there are several reported cases where attackers breaks into a low security site to retrieve thousands of username/password pairs and directly try them one by one at a high security e-commerce site such as eBay. As expected, this attack is remarkably effective.

A Simple Solution. PwdHash is an browser extension that transparently converts a user's password into a domain-specific password. The user can activate this hashing by choosing passwords that start with a special prefix (@@) or by pressing a special password key (F2). PwdHash automatically replaces the contents of these password fields with a one-way hash of the pair (password, domain-name). As a result, the site only sees a domain-specific hash of the password, as opposed to the password itself. A break-in at a low security site exposes password hashes rather than an actual password. We emphasize that the hash function we use is public and can be computed on any machine which enables users to login to their web accounts from any machine in the world. Hashing is done using a Pseudo Random Function (PRF).

Phishing protection. A major benefit of PwdHash is that it provides a defense against password phishing scams. In a phishing scam, users are directed to a spoof web site where they are asked to enter their username and password. SpoofGuard is a browser extension that alerts the user when a phishing page is encountered. PwdHash complements SpoofGuard in defending users from phishng scams: using PwdHash the phisher only sees a hash of the password specific to the domain hosting the spoof page. This hash is useless at the site that the phisher intended to spoof.

Publications

- Blake Ross, Collin Jackson, Nicholas Miyake, Dan Boneh and John C. Mitchell
  Stronger Password Authentication Using Browser Extensions.
  Proceedings of the 14th Usenix Security Symposium, 2005.

- PowerPoint presentation

- Related project: SpoofGuard.

Download

Please note: These prototypes are intended for demonstration purposes only. We reserve the right to change the hashing algorithm in future versions, which may require you to reset your passwords if you want to upgrade.

- Version 1.7 for Mozilla Firefox 3.5 and earlier: Installer (16KB), Source code (16KB)
- Version 0.6 (beta) for Internet Explorer 7: Installer (537KB) Source code (144KB)
- Version 0.5 (beta) for Internet Explorer 6: Installer (628KB) Source code (147KB)
- Opera users may be interested in Haris Andrianakis's PwdHash for Opera, but read about the limitations of user scripts in our paper.
- Send us feedback

Deployment Challenges

PwdHash preserves the benefits of password authentication such as mobility without any hardware requirements. Our primary design goals are *not to change to the user experience* and *not to require server-side changes*. To do so, we had to overcome a number of challenges:

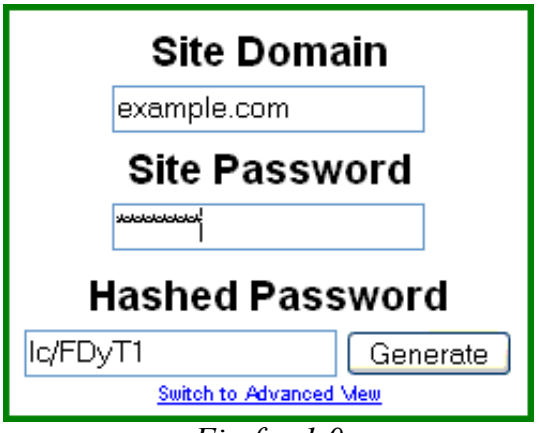
- PwdHash must defend against JavaScript at a phishing site that may confuse users into typing their passwords in an insecure location (such as a text field that is made to look like a password field). PwdHash includes a number of clever mechanisms to defend against such attacks.

- After PwdHash is installed users can set up hashed passwords at the various sites they use by resetting their password. Typically, reset pages ask the user to type in the old password and then enter the new password twice. PwdHash must somehow recognize the old password field and avoid hashing it. PwdHash relies on the user to pick new passwords that start with `@@` so that they can be distinguished from regular passwords. Alternatively, the user can press the password key (F2) before entering the password to indicate that the password should be hashed.

- We found a small number of sites (i.e. one site) where the password reset page is hosted on a different domain than the password use page. As a result the wrong password hash is registered at the site after password reset. One solution is to create a list of such sites so that PwdHash can tell how to handle them. Other solutions involving server-side changes are also possible.

- Occasionally, users want to login to their web accounts on machines where they cannot install browser extensions (e.g. at Internet cafes). In this case users can connect to our web site
  https://www.pwdhash.com
  where they are presented with one of the following forms, depending on whether their browser supports clipboard operations via script:

Internet Explorer 6.0



Firefox 1.0

  Javascript computes the password hash on the local machine (so that the user's password is never

communicated to Stanford). The resulting hashed password can be pasted into the user's web login form.

Project Staff:

- Dan Boneh
- Collin Jackson
- John Mitchell
- Nick Miyake
- Blake Ross

Stanford Security Lab
