UGIS 188: Inquiry 4 Proposal

Brandon Chinn, November 13 2016

Introduction

Every person who owns a computer has had experience with creating and remembering passwords. Whether it's the password to log on to the computer or the password to their email account, today's society relies heavily on passwords. However, despite the importance of passwords in keeping confidential information private, people often choose poor passwords, such as "password", "123456", or "abc123" (Doel, 2012). Even requiring passwords to be of a certain format does not always create more secure passwords. Kelley et al. (2012) did a study where participants were asked to create a password in a certain format. For example, one participant might be asked to create a password of at least 16 characters (with no other requirements), while another participant might be asked to create a password of at least 8 characters without containing a dictionary word. Their study found that longer passwords were actually more secure than passwords with requirements (e.g. one number, one upper case letter, and no dictionary words), undermining the theory behind modern websites' password requirement schemes.

There has been some developing research in the area of image-based password schemes, where users supplement a text-based password with a series of pictorial challenges that is resistant to current password hacking techniques. J. Blocki et al. (2013) wrote a paper formalizing such a scheme, where, after the user creates a username and password, the website shows the user a set of images, randomly generated from the password. The user then writes labels for each image and sends the labels to the server. The server will then store the user's username, password (encrypted), and the labels. To log back in, the user will give their username and password, then the server will re-generate the random images. If the user gives the same password, the images would be exactly the same as before; otherwise, the images will look different (but the server still returns the images!). The server will also send back the labels, and the user will have to match the label they wrote to the corresponding image.

J. Blocki et al. did include a user study in their report, but the user study focused on the amount of time it took users to create an account and log back in. This study seeks to test the security behind the scheme, testing to verify that it would, in fact, be infeasible for an attacker to crack passwords using this scheme.

Research Questions

- Does the accuracy of responses go down as the number of images in the password goes up?
- What are the best values to use for the scheme's security parameters in order to optimize user experience while maintaining security?

Regarding the second question, there are three security parameters that can be customized for higher security guarantees, but usually at a cost to user experience and usability. These parameters are:

- **Number of images**: as the number of images goes up (and the number of labels the user makes for each image), the security goes up in a similar fashion to a longer password. However, it might be harder for a user to remember all the labels for the images, and it would be impractical for a user to spend more than a couple minutes just to log in.

- **Accuracy threshold**: the server would typically allow some percentage of images to be wrong for the user to still log in. A lower threshold means more security (the user would have to be closer to 100% accurate) but might generate more false negatives (the correct user is locked out). A higher threshold means less security (an attacker has more opportunities to guess the right labels), so it might generate false positives. In addition, a higher threshold would take longer to authenticate, since the server has to enumerate every possible combination of labels, and there are more combinations when more labels are allowed to be wrong.
- **Hash iterations**: servers typically store hashed passwords in their databases rather than the actual password. A hash function is a function that always returns the same thing for the same password, but it's really hard to get back to the original password from the hash. In order to deter attackers, passwords usually go through multiple iterations of hashing so that it takes longer for an attacker to check each password (usually with negligible delay to the user). However, this scheme requires passwords be hashed multiple times to verify that the password is correct, so it might not be as negligible to the user as with usual text-based password schemes.

Method

Implement server that runs this authentication process. Have users (Berkeley CS group, other facebook groups) create accounts (randomize number of images, 3-7). A day later, have them login again (store and show percentage of labels matched). Mention reduction in security parameters

Data Analysis

Regression line between number of images and accuracy. Scatter plot accuracy vs time (graph plot for each num_images, hold iterations constant). Scatter plot iterations vs time (graph plot for each num_images, hold accuracy constant).

Run analysis on pilot data

References

J. Blocki et al. (2013). *GOTCHA password hackers!* Retrieved from http://dl.acm.org/citation.cfm?id=2517319

K. Doel (2012). *Scary logins: Worst passwords of 2012 and how to fix them.* Retrieved from http://www.prweb.com/releases/2012/10/prweb10046001.htm

P. G. Kelley et al. (2012). *Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking* Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6234434&isnumber=6234400