Fine Grained Access Control in Online Social Networks

Amin Tootoonchian Geoffrey Salmon Ahmad Ziad Hatahet

1. INTRODUCTION

Online social networks (OSNs) have become the primary way many people distribute personal content. People are using websites such as facebook, myspace and flickr[1] to share text and photos. Facebook alone stores 160 terabytes of user photos, with 60+ million photos added every week.[2] Access to this user content is subject to the controls supported on the individual OSNs, and the granularity of the available access control varies between different sites. Some sites only support hosting content that is either private or entirely public. Other sites allow some content to be seen only by the friends of a user. If the controls are too coarse, a user must choose between not publishing certain content or allowing unwanted people to view it. Neither option is desirable.

We propose a system where users can specify who can view their content on a per-friend basis. To our knowledge, no popular OSN provides this feature. However, we are not suggesting the creation of a new OSN. A key detail of our system is that the content appears seamlessly integrated within existing OSNs and benefits from any content-specific features the OSN provides.

Our current implementation, called ImageLock, allows a user to assign his or her friends to relations. For example, a user could have separate relations for relatives, coworkers and friends. Images that the user posts to an OSN, can be locked to a set of relations. Only people in those relations will be able to view the real image. Others will see a fake image in its place.

We will describe the design of ImageLock in depth in section 3. For now, note that we are not presenting ImageLock as an ideal solution. We would prefer to see existing OSNs support fine grained access controls themselves. However, ImageLock is important because it gives users a taste of what is possible. If ImageLock gains a userbase it will demonstrate that users wish greater control over their personal content. Most importantly, it may also provide an OSN with incentive to add similar access control features to their site.

2. BACKGROUND AND RELATED WORK

Authenticating access to shared content is a general problem that has been studied in many contexts. Personal data stored on OSNs is merely a new incarnation of the problem. Fundamentally, even sharing data between users on a single computer may require that the operating system enforce file permissions or access control lists (ACLs). In this environment, user authentication is provided by the operating system's normal login controls.

User authentication is more complicated when the users are on different computers. SFS [1] is a remote file system on top of which almost any user authentication scheme can be built. In [2], the authors add authentication and ACLs to SFS based on user public/private key pairs. Sharing file systems is very useful is many applications. However this approach is not applicable to content published on web-based services such as OSNs, which users interact with using a web browser.

To authenticate a user, it is necessary to know what defines a user. Currently, each social network maintains its own database of users, and many people create separate profiles on different OSNs. Efforts such as OpenID[3] allow a users to consolidate their many accounts into a single online identity. This applies not only to OSN accounts but can be used with any web-based account that supports it.

Recently, access control schemes have been proposed that leverage social networks. In [3], the authors propose creating social attestations corresponding to relationships between users. These attestations can be validated to prove the relationship exists. Securely validating the attestations is an online process that cannot be faithfully done by ImageLock alone. However, because the intent of ImageLock is only to demonstrate a concept to users, it is conceivable that ImageLock can insecurely validate attestations under the assumption that OSNs will correctly implement them.

[1] http://www.facebook.com/, http://www.myspace.com/ and http://www.flickr.com/

[2] As of May 21, 2007, according to a facebook developer http://blog.facebook.com/blog.php?post=2406207130

[3] http://openid.net/

3. DESIGN

ImageLock's main component is a Firefox extension. It can be easily installed by any user with only a few clicks. It's vital that there be as few barriers to using ImageLock as possible. Before describing the behaviour of the extension itself, let's examine the other elements of ImageLock.

There are two options for where a protected image will be stored. The first is to publish it on the OSN in the normal manner but encrypt it so that only authorized users can view it. We chose not to do this in ImageLock because the OSN will still provide a copy of the encrypted data to unauthorized users. Although this is not a problem in the short term, eventually the encryption used will be considered weak, either because a weakness is found in the algorithm or simply the key length is too short. If sufficiently motivated, an unauthorized user can save encrypted images until they can be decrypted easily. To avoid this, ImageLock publishes a fake image on the OSN and stores the real image elsewhere.

For simplicity, currently the real image is stored on our own server. Any server could be used as long as it provides unguessable URLs for the uploaded images and does not index them on publicly accessible web pages. To improve the scalability of ImageLock, we plan to store the real images on a service like Google's Picasa[4], which provides the required properties.

The fake image is uploaded to the OSN along with some extra information. For maximum flexibility this information would be embedded directly within the fake image in a way that can survive the resizing and other transformations that OSNs apply to images. However, our current implementation is restricted to the OSN flickr, and the extra information is simply stored in the text description attached to each photo.

The extra information associated with each fake image contains two things: an image id and a URL of a LocationMap. This information is considered public and is not encrypted in any way. The LocationMap is a publicly accessible XML document that maps image ids to the URL of the real location. In practice, this will likely be implemented as a two-level mapping from user to relationship and relationship to image location. The mapping is encrypted so that only the correct users can access the real image location.

The ImageLock extension serves two purposes. First it intercepts when the user uploads pictures to flickr and optionally replaces the image with a fake one, uploads the real image and the updated LocationMap to our server. Second it will scrape the HTML of flickr pages to extract the extra information stored in the image descriptions. Using this information, ImageLock downloads the LocationMap and access the real image URL. Finally it replaces the image the user sees on the flickr page with the real image.

4. EVALUATION

It is difficult to create a useful metric to evaluate ImageLock in it's current form. Once the cryptography elements are added to the LocationMap it will be important to measure the delay incurred when accessing a page with many locked images or when updating relations and images stored in a LocationMap.

We can also evaluate how well the LocationMap will scale. We can approximate the upperbound on the number of friends, relations and images a user can have when ImageLock stores everything in a single LocationMap. In practice, the LocationMap can be divided across many files, but it is useful to know at what point that extra complexity becomes necessary.

If we add the ability to embed information directly within images, several other performance metrics become important. We can evaluate the time required to extract information for many images on a single page. Determining the amount of information the can be reliably stored in images of different sizes is also important. OSNs often provide thumbnails of uploaded images so we should evaluate how robust the stored information is to image transformations like downsampling.

5. CONCLUSIONS

This paper describes ImageLock, a scheme to allow fine grained access control of images in existing online social networks. We implemented ImageLock as a Firefox extension which intercepts images as they are uploaded to an OSN and optionally replaces them with a fake image. When authorized users view the image, the extension seamlessly replaces the fake image with the real one again.

The intent of ImageLock is to let users evaluate if the addition of fine grained access control is useful. We do not expect it to be a long term solution. Instead we hope that OSN providers will be motivated to support ACL directly. Direct support will be more robust and avoid the need for a browser extension.

6. REFERENCES

[1] Separating key management from file system security, D. Mazieres, M. Kaminsky, M. F. Kaashoek, E. Witchel. Symposium on Operating Systems Principles (SOSP) 1999.

[2] Decentralized User Authentication in a Global File System., M. Kaminsky, G. Savvides, D. Mazieres, M. F. Kaashoek. Symposium on Operating Systems Principles (SOSP) 2003.

[3] A Social Networking-Based Access Control Scheme for Personal Content, Kiran K. Gollu, Stefan Saroiu, Alec Wolman (SOSP) 2007.

[4] http://picasa.google.com/