Privacy Suites: Shared Privacy for Social Networks

Joseph Bonneau
University of Cambridge
Computer Laboratory
jcb82@cl.cam.ac.uk

Jonathan Anderson
University of Cambridge
Computer Laboratory
jra40@cl.cam.ac.uk

Luke Church
University of Cambridge
Computer Laboratory
lec40@cl.cam.ac.uk

1. INTRODUCTION

Creating privacy controls for social networks that are both expressive and usable is a major challenge. Lack of user understanding of privacy settings can lead to unwanted disclosure of private information and, in some cases, to material harm. We propose a new paradigm which allows users to easily choose "suites" of privacy settings which have been specified by friends or trusted experts, only modifying them if they wish. Given that most users currently stick with their default, operator-chosen settings, such a system could dramatically increase the privacy protection that most users experience with minimal time investment.

2. USABILITY CHALLENGES

Configuring privacy in a social network is a challenging usability problem for several reasons. Using the terminology of Cognitive Dimensions, most privacy setting UIs are both $diffuse_{\mathrm{CD}}$, having a large number of settings, and $viscous_{\mathrm{CD}}$, requiring a large amount of time and effort to understand and configure [4]. Facebook, for example, presents its users with 61 privacy settings on 7 different configuration pages, LinkedIn has 52 settings on 18 pages, and Windows Live Spaces has 27 pages, each with only one setting [3].

Even with dramatic improvements in usability, privacy suffers from the *secondary goal* problem [7]; users will always prefer connecting with their friends on social networks to managing access control lists. Web-crawls and user surveys have estimated that over 80% of social network users do not change their privacy settings at all from the default [1, 5] and less than 1% of users opt-out of several obscure privacy-violating features on Facebook [2].

In addition, social networks are a rapidly evolving technology and new features are constantly introduced. Often, all users are opted-in to these features despite their adverse privacy implications [2], meaning that users must frequently update their settings to maintain control of their data.

Despite these problems, we find evidence that users do want better control of their privacy. The majority of users city privacy as a concern [1]. More compelling though, is evidence from the web that users are struggling to make do with the available privacy controls: a recent blog posting "Ten Privacy Settings Every Facebook User Should Know" was viewed over 500,000 times and became the top *de.li.cio.us* bookmark for February 2009 [6].

3. SHARING PRIVACY

Given the above problems, and users' apparent desire for protecting their private data, we argue that a radical change in paradigm may be necessary. We propose embracing the nascent pattern of experts recommending appropriate setting to casual users, building in support for users to quickly and painlessly adopt another user's privacy settings.

This approach has many parallels in other security configuration domains in which delegating security policy to a trusted authority is common. These tasks are similar to the social networking privacy problem in that they are tedious and require frequent updates. For instance, more than 50 million users have installed the AdBlock Plus plugin for Mozilla Firefox, which allows users to select a trusted source to create a blacklist of advertising domains. Automatic patching and anti-virus software have become ubiquitous in modern operating systems, allowing users to select a trusted source for updates as new vulnerabilities are discovered.

4. PRIVACY SUITES

We propose social networks formalise the notion of shareable "Privacy Suites" and build in support for users to find and adopt them.

4.1 Abstraction

The first building block is an abstract specification format for privacy settings. Ideally, this should be Turing-complete, allowing the specification of new and arbitrarily complex policies. By de-coupling the specification from the UI, we can enable arbitrarily sophisticated settings to be crafted, while still supporting simple GUIs when needed.

Experts can thus define a Privacy Suite via *privacy programming*, as in Figure 1. Privacy Suites could also be created directly through existing configuration UIs, exporting them to the abstract format. Hybrid design interfaces could also be designed, enabling new public interfaces to be built for users to manipulate their settings.

The disadvantage of a rich programming language is less understandability for end users. Given a sufficiently high-level language and good coding practice, motivated users should be able to verify a Privacy Suite. The main goal is transparency, which is essential for convincing influential users that it is safe to use.

4.2 Distribution

Once a privacy suite has been created, it must be distributed to the social network's members. This could be done via existing distribution channels, such as an expert posting a recommended suite on their web site, or through the social network itself, by users adopting a suite which is used by a trusted friend.

```
def showPhotoStream(self, user):
   if user in self.friends: return True

   mutualFriends = self.friends.intersection(
                     user.friends)

   if len(mutualFriends) > 10: return True
   else: return False
```

Figure 1: Privacy programming

An important challenge is establishing trust. However, a social network already has a powerful channel for building trust via social cues. Each user can choose to have a link placed on their profile indicating which Privacy Suite they are using. A curious friend can then view social information to establish trust in the suite, as shown in Figure 2.

This trust will be much stronger if good graphical tools are developed to explain, given the suite's specification, exactly what effects it will have to the visiting user. It may also be possible to use static analysis tools to automatically find inputs for which a given suite differs from a user's current suite, which could be used to display exactly what will change if the user adopts the new suite.

4.3 Installation

After importing a suite and customising it if desired, a key challenge is for a user to map his friends into the roles for used by the suite for role-based access control decisions (e.g. identifying friends, family members or co-workers). If designed poorly, this process could be as arduous as managing privacy settings under the current UI.

Thus, a key requirement is effective interfaces for quickly assigning friends and groups of friends into roles defined by a newly adopted suite. This could be initially seeded by automatically placing friends into groups based on the network they are in, or clustering friends into highly-connected groups which are likely to be friends from a similar social context. A user could then graphically manipulate these groups, dragging them into the necessary roles and overriding them as needed.

4.4 Maintenance

In an environment where new features cause continual change in available privacy controls, a system for applying privacy suites should have a mechanism to update users' settings. While users could choose to adopt a suite "statically" if they don't trust the suite's author to perform automatic updates, they may also choose to adopt the suite "dynamically," in which case they will automatically receive the owner's changes as new features are introduced. Ideally, the network operator would give advance notice of new feature roll-out, giving authors time to update their suites before new features are deployed.

Maintenance may become difficult if users have excessively customised a suite, similar to the headaches experienced in patch management as users customise their operating system distribution. In our case, simple local changes such as blocking a specific individual are unlikely to cause conflicts with updates. Similarly, most changes will be relatively minor and deal with new features, not breaking local customisation. Thus, we think that requiring manual intervention

Figure 2: Establishing trust in a Privacy Suite

in these cases is reasonable, and will not cause problems for the majority group of privacy-concerned but pragmatic users we are targeting.

5. REFERENCES

[1] A. Acquisti and R. Gross. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In *Privacy Enhancing Technologies – LNCS 4258*, pages 36–58. Springer Berlin / Heildelberg, 2006.

[2] J. Bonneau, J. Anderson, and G. Danezis. Prying data out of a social network. In *ASONAM 2009 : Advances in Social Networks Analysis and Mining*, 2009.

[3] J. Bonneau and S. Preibusch. The Privacy Jungle: On the Market for Privacy in Social Networks. *WEIS 09: The Eighth Workshop on the Economics of Information Security*, March 2009.

[4] T. Green and M. Petre. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages and Computing*, 7(2):131 – 174, 1996.

[5] B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *WOSN '08: Proceedings of the First Workshop on Online Social Networks*, pages 37–42, New York, NY, USA, 2008. ACM.

[6] N. O'Neill. Are Granular Privacy Controls Too Complicated For Users? http://www.allfacebook.com/2009/03/facebooks-complex-privacy-settings/, 5 Mar 2009.

[7] A. Whitten. *Making Security Usable*. PhD thesis, Carnegie Mellon University, 2004.