(Under)mining Privacy in Social Networks

Monica Chew Dirk Balfanz Ben Laurie[*]
{mmc,balfanz,benl}@google.com
Google Inc.

1 Introduction

Social networking sites like Facebook or MySpace allow users to keep in touch with their friends, communicate and share content with them, as well as engage in other multi-user applications. What distinguishes such sites from other, pre-"web 2.0" applications is that users make explicit their social network [2]: who your friends are (and — by omission — who they aren't) is a constant presence in the user interface of such sites. It is perhaps because of this that social networking sites give the impression of a semi-public stage on which one can act in the privacy of one's social circle.

This assumption, however, is not always merited. In this paper, we point out three distinct areas where the highly-interlinked world of social networking sites can compromise user privacy. They are

- lack of control over activity streams,
- unwelcome linkage, and
- deanonymization through merging of social graphs.

In the following sections, we will define each of these privacy-sensitive areas, giving examples (most real, some hypothetical) for each. Finally, we will derive recommendations for usable designs from our observations.

2 Activity Streams

An *activity stream* is a collection of events associated with a single user[1]. These events may include changes the user made to their profile page, the fact that the user added or ran a particular application on the social networking site, that they shared a news item, or that they communicated with one of their friends. Different social networking sites use different nomenclatures: on Facebook, activity streams are called "mini-feeds", on Orkut they are called "updates." A user's activity stream is typically viewable by their friends, as defined by the social networking site.

There are two fundamental ways in which lack of control over activity streams may compromise a user's privacy. First, a user may not be aware of all the events that are fed into their activity stream. Second, a user may not be aware of the audience who can see their activity stream. Below, we mention three real-world examples: two in which users lacked control over the events going into the activity stream, and one in which users lacked control over the audience who could see the activity stream.

Facebook Perhaps the most widely publicized recent frustration of user expectations in social networking is Facebook's Beacon feature. With this feature, third-party websites such as Blockbuster, eBay, or Travelocity insert events into a Facebook user's activity stream whenever that user adds a movie to her queue, makes a purchase, books a trip, *etc.*. However, initially users could not easily monitor or control the events fed into their activity streams — the initial default user preferences were to send Beacon information from everywhere, and Facebook users could only opt-out of Beacon by specifying each website. Because there was no easy way to opt-out of Beacon, it garnered an immediate allergic reaction in the press and among Facebook users [6, 7]. Facebook eventually resolved the problem by providing a global opt-out, where users do not have to enumerate every website from which they do not want to send information.

coComment coComment, a comment tracking service[2], serves as a second example for unexpected events showing up in a user's activity stream. coComment tracks conversations that occur in the comments section of blogs by a client-side browser extension that records comments the user types, and publishes them on the coComment server. Initially, the coComment extension simply recorded everything the user typed, without regard to whether the website the user was visiting was public or not. A coComment user was surprised and dismayed

[*] The opinions expressed here are those of the authors and do not necessarily reflect the positions of Google.

[1] This collection is ordered in time and potentially infinite – hence our choice to call it a "stream".

[2] www.cocomment.com

to find their messages to Citibank published on coComment [9]. In resulting blogposts, the coComment user clearly did not understand the mechanisms involved and blamed Citibank for running an insecure website[3]. Although coComment can be used securely by enabling and disabling the extension every time the user posts a real comment, the cognitive burden placed on the user is too high. Citibank "solved" this problem by placing a warning on its customer service form, and coComment similarly solved the issue by adding a blacklist capability.

Google Reader Google Reader is a syndication service which allows users to subscribe to RSS feeds of various news and blog sites. Google Reader then aggregates these feeds for easy reading. Google Reader allows users to mark news items or blog posts of interest, then share these items with chosen friends. In essence, Google Reader allowed the user to create her own RSS feed of "interesting" stories. In December 2007, Google Reader launched a new feature that (after displaying a pop-up that allowed users to opt-out) shared this feed with the user's Google Talk contacts, creating an activity stream of shared news stories for that user[4].

Although the shared items had always been available as a public feed, some users had actively controlled who the audience of that feed was by disclosing the obfuscated feed URL to people of their choosing. When Google expanded the audience of that particular activity stream to include Google Talk contacts, users were caught by surprise [5]. Google Reader responded by explaining to users how to manage their shared items using existing tools. In addition, Google Reader included a feature that allowed users to migrate their shared items to a tag or to their starred items, which could then be shared via an obfuscated URL.

3 Unwelcome Linkage

Unwelcome linkage occurs when links on the Internet reveal information about an individual that they had not intended to reveal. Unwelcome linkage is not limited to social networking sites, but may occur wherever graphs of hyperlinks on the World Wide Web are automatically created to mirror connections between people in the real world.

Consider Bob, who has multiple personae on the Internet: would-be bounty hunter at the gun club, and amateur horticulturist at the rose garden. Given Bob's disparate interests, it is certainly possible that he would *not* want his buddies from the gun club to learn of his (in their eyes) anachronistic or useless plant-related hobby, but has no qualms showing his weapons collection to his fellow horticulturists. Maintaining separation of these different personae is sometimes called impression management [3] in identity management systems.

Trackbacks Suppose Bob maintains a personal blog about his bounty-hunting activities, and that blog enables *trackbacks URLs*. Trackback URLs on a web page are a way to keep track of new stories or blog posts linking to that web page. They can result from a list of referral urls kept by the hosting webserver, or the trackback protocol implemented by several blogging services. Even if Bob diligently masks personal information on his own blog (for example, by never posting the address of his home, which is well-known in his town for having ornate flower decorations), Bob can be "outed" by someone else through trackbacks. Suppose Bob's friend Alice, who is the chair of the local horticulturist club, writes in her own blog that she visited Bob for dinner, mentions Bob's name, and links to Bob's post. A casual reader *of Bob's pseudonymous blog* can find his real name, and learn about his *other* hobby despite Bob's vigilance, simply by following a trackback URL that an eager software system added to his blog post.

Accidental linkage Suppose Bob also keeps a Flickr account on which he hosts pictures of his guns. However, because of his interests in flowers, he not only belongs to the "We Like Guns" Flickr group, but also to the "Flower Lovers" group. So when Bob uses a feature on Flickr that allows him to easily add a photo of his favorite gun to a blog posting, he might be unaware of the fact that the photo added to his blog is, in fact, a link back to a Flickr page containing his Flickr name, revealing interest in flowers. An unexpected link, together with Flickr's social-networking-style feature of public profile pages caused Bob's horticultural persona to be conflated with his gun club one.

4 Merging Social Graphs

The third privacy-sensitive area comes from the fact that social networking sites tend to extract a lot of personally-identifiable information from people (from birth date and address to favorite books, to travel destinations, *etc.*). It may be possible to de-anonymize users by comparing such information across social networking sites, even if the information is partially obfuscated in each networking

[3] Citibank serves the online form for contacting customer service over HTTPS only after the customer logs in.

[4] Google Talk contacts are everyone with whom a user has chatted, which might include co-workers, supervisors, and friends.

site[5].

This issue becomes particularly pertinent today as people are getting tired of constantly redeclaring their friends in different contexts, leading to the idea that aggregation and centralization of these relationships is needed [4]. The problem here is preserving implicit and explicit expectations about the use of the data when it is divorced from its original context. If, for example, Bob, as a user of various networks, mines those for ways to correlate his friends' accounts across networks using data such as favorite books and movies to match them, then he has probably not violated any reasonable expectations of his friends.

But if he then publishes that aggregated view he may very well be revealing things his friends would prefer he did not. They may not even know he had access to that data; for example, suppose Alice has an alter-ego, Vinylgirl, and Bob has one, Leatherboy. Alice and Bob know each other on some ordinary site like Facebook or LinkedIn, but Leatherboy and Vinylgirl are also friends on some site for those of different sexuality. Bob (or rather his software) might perhaps correlate Alice and Vinylgirl but she may not be happy for Bob to learn this correlation, and certainly not to publish that information on some centralized social graph repository.

Of particular importance here is that Alice did not actively participate in revealing her alter-ago. Bob's decision to reveal the correlation between himself and Leatherboy has led to Alice being involuntarily outed. Furthermore, the outing could be even more indirect — for example, Bob might himself have been outed by some "friend" of his. A cascade failure could lead to some quite small change in the graph causing a huge amount of data to be correlate-able.

5 Recommendations

In the previous sections we introduced three new areas where social networking sites can compromise user privacy. All three problem areas had in common that while the systems worked as intended by their designers, users were unprepared for the change in the use of their data and their social networks.

In the case of the activity stream, users form certain expectations about which events will be fed into their activity stream, and who the audience is that gets to see their activity stream. Users will be surprised and confused if social networking sites don't meet those expectations.

In the case of unwelcome linkage, the compromising information about a user is already out there on the Web, but the user expects that this information is hard-to-find and that they will be able to keep it away from certain audiences. This expectation, however, is not in line with how certain software (like backtracking features of blogs) works. This area will be increasingly challenging, with new technologies that eagerly link up portions of the web that users thought separate.

Finally, merging social graphs of different social networking sites may upset the model that users express quite explicitly by creating different personae in different social networks. Those personae may overlap just enough to link them up and identify them as facets of the same human being.

From these observations flow naturally a few design implications for social networking sites and social networking features of Web applications, which we summarize below. These are not intended to be comprehensive or definitive solutions, but a set of possible design models as developers continue to innovate in social networking.

Activity Streams First, users should be explicitly aware of every event that gets fed into their activity stream. While that may not necessarily mean that every time such an event is generated the user receives a message, applications should be explicit about which activities of the user generate events for their activity stream.

Second, users should be given control over which events make it into their activity stream. For example, it would be good if a user could block a single event from being added to their activity stream, install filters that block classes of events, or disable whole applications from posting to their activity stream. Users should be able to remove events from the activity stream after they have been added to it by an application.

Third, users should be explicitly aware of who the audience is for their activity stream. The user interface of the social networking site should make it easy to list all the principals that can see certain events in the activity stream, both at event creation time and later, after the event has been added to the activity stream.

Fourth, users should be in control over who the audience is for their activity stream. It should be easy to remove and add principals from that audience, both from the activity stream as a whole as well as from single events.

Finally, application developers should build their applications such that the creation of activity stream events is more likely to be in sync with the user's expectation. For example, the coComment software could, instead of posting every entry the user makes into an HTML form, only post those entries that appear in an RSS feed discoverable

[5] See the deanonymization of the Netflix ratings data by joining publicly available data from the Internet Movie Database set by Narayanan and Shmatikov [8], or attacks on deanonymized social graphs by Backstrom *et al.* [1].

from the page to which the HTML form is posted[6].

All of these recommendations pose significant usability challenges in designing interfaces to minimize cognitive burden while preserving user choice.

Unwanted Linkage To combat unwanted linkage, we again propose to build tools that make explicit what information is available about users on the Internet (and potentially only one unanticipated trackback URL away). We propose building a tool for automatic link discovery — whenever a user creates content on the web, show the user what information would be revealed by finding the transitive closure of profile-related links, so the user can decide whether or not publishing that content creates too many leaks. The automated link discovery could mitigate nasty surprises in the case of Bob's blog, for example. Although this solution reduces the cognitive burden of the user to figure out available inferences from the social graph, it does not eliminate the burden entirely.

Recently Google released a social graph API that allows developers to find publicly available links to or from a particular URL. The social graph API is a potential building block for an automatic link discovery tool.

It would also be useful to give users a choice between "linked" and "link-free" versions of certain services. For example, Bob's blog should have an easy switch to turn off trackbacks. Similarly, Flickr could allow users to maintain different aliases for different photo sets and groups for easier separation of different personae.

Merging Social Graphs To address the problem of merging social graphs we propose tools similar to those used to detect inferences on the Web [10]. Armed with such tools, users who sign up for their $n+1$-th social networking site could be warned that the information they provide to that site can be used together with the information they provided on the previous n sites to infer their identity. Again, as in the previous two cases, the goal is to make explicit the consequences for the user's privacy if they go through with a certain action (in this case, giving information to yet another social networking site).

6 Conclusions

In this position paper, we identify three new privacy-sensitive areas in the world of interlinked social networks. A user's privacy can be compromised by

- feeding unanticipated events into their activity stream, or exposing their activity stream to an unanticipated audience,
- eagerly and automatically linking between pages representing users' different personae, and by
- mining different social networks for the purpose of merging users' social graph.

In each of those cases, we believe that the root of problems in the past has been a discrepancy between the mental model the user formed about the system, and how it actually worked. The key to solving the problems, then, is to align users' anticipations of the system with its actual workings.

This can either happen by adjusting the system to fit the mental model of the user (*e.g.,* by making coComment more selective about which items it posts to the user's activity stream), by making explicit how the system works (*e.g.,* by notifying users what information about them can be inferred from the data available in two different social networking sites), or by giving users more control over the system (*e.g.,* by letting the user have fine-grained control over the events fed into their activity stream).

More work is needed in particular in the latter two categories: how do we find out, and inform the user, about inferences that can be made about them, given the information in different social networking sites; and how do we give users controls they can understand and use that allow them to manage their privacy in an increasingly interlinked world of social networks.

References

[1] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, 2007.

[2] danah m. boyd and Nicole B Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 2007.

[3] Rachna Dhamija and Lisa Dusseault. The seven flaws of identity management: Usability and security challenges, 2008. IEEE Security and Privacy Special Issue on Identity Management, To Appear.

[4] Brad Fitzpatrick and David Recordon. Thoughts on the social graph, 2007. http://bradfitz.com/social-graph-problem/.

[5] Miguel Helft. Google thinks it knows your friends, December 2007. http://bits.blogs.nytimes.com/2007/12/26/google-thinks-it-knows-your-friends/?ref=technology.

[6] Many blogs have two RSS feeds discoverable from a blog post: the feed of the blog itself, and the feed for comments of this particular post. Looking for this pattern as a heuristic should drastically reduce the amount of "false positives", *i.e.,* events that were posted but shouldn't have been.

[6] Caroline McCarthy. MoveOn.org takes on Facebook's 'Beacon' ads, November 2007. http://www.news.com/8301-13577_3-9821170-36.html.

[7] Ellen Nakashima. Feeling betrayed, Facebook users force site to honor their privacy, November 2007. http://www.washingtonpost.com/wp-dyn/content/article/2007/11/29/AR2007112902503.html?hpid=topnews&sub=AR.

[8] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset, 2006.

[9] John Ratcliffe-Lee. Huge security hole in Citibank's online account center, March 2007. http://jratlee.tumblr.com/post/189652.

[10] J. Staddon, P. Golle, and B. Zimny. Web-based inference detection. In *Proceedings of USENIX Security 2007*, 2007.