Privacy Butler: A Personal Privacy Rights Manager for Online Presence

Ryan Wishart, Domenico Corapi, Anil Madhavapeddy, Morris Sloman
Department of Computing,
Imperial College London,
London, U.K.

Email: {r.wishart, d.corapi, a.madhavapeddy, m.sloman}@imperial.ac.uk

***Abstract*—The online presence projected by a person is comprised of all the information about them available on the Internet. In online communities and social networking services, it is often possible for third-parties to modify this content by, for example, commenting on existing content or uploading new content. This has the potential to negatively impact the privacy of a presence owner (the person referred to by the on-line content) by disclosing information about them without consent. In this paper we propose a Privacy Butler, an automated service that can monitor a person's online presence and attempt to make corrections based on policies specified by the owner of the online presence.**

***Index Terms*—privacy, online presence, social networking**

I. Introduction

As people increasingly interact online, the way a person is perceived within their peer group, amongst colleagues and by current (as well as future) employers is affected by the person's online presence. This presence is an abstract impression of a person obtained by collating the online information about them. Common sources for this information include: social networking services (like Facebook[1] and Twitter[2]), blog postings, news articles, Wikipedia entries and online photograph albums such as Flickr[3].

In many cases it is possible for the information supporting a person's presence to be modified by external parties, often without the consent of the person identified by the information (who we term the owner of the online presence). This has been demonstrated recently on the Wikipedia collaborative encyclopaedia website where entries for prominent Australian politicians were vandalised [1]. In the online social networking context, user modification of content is directly encouraged. Examples of this, drawn from the Facebook social networking service, include commenting on a user's status and tagging friends in newly uploaded photographs. These unauthorized modifications can severely impact the privacy of a person, if sensitive information is disclosed that should not have been. One example of this is when photographs or comments are uploaded for everyone (including employers) to see resulting in people losing their jobs [2].

[1]www.facebook.com
[2]www.twitter.com
[3]www.flickr.com

Within this paper we employ the definition of privacy given by Minch [3] in which privacy is considered to be the right of an individual to *control* how information about them is collected, stored, used and communicated to other parties.

One possible route to managing privacy is to reduce online interactions and avoid social networking services. Unfortunately, the prevalence of such services means that such actions have negative social consequences [4]. Alternatively, it is possible to manually audit one's online presence, though this is increasingly difficult given the rapid proliferation of social networking sites, blogs and wikis.

In this paper we take the stance that social networking services and online presence are increasingly a part of modern society. To reduce the privacy ramifications associated with using them, we propose the *Privacy Butler*, an automated service that can audit a person's online presence, addressing the communication aspect of Minch's definition of privacy. The service improves user privacy by providing notice of changes in online content, and facilitates modification of content that does not meet owner-specified policy. In this way, the Privacy Butler monitors and controls the online presence of its owner.

The remainder of this paper is structured as follows. In Section II we discuss the Personal Container technology supporting the Privacy Butler before providing a detailed discussion of the butler's operation in Section III. We then outline a prototype implementation of the system and provide an evaluation in Section IV. This is followed by a critical survey of the related work (Section V) before we conclude the paper in Section VI.

II. Background

A key aspect of the Privacy Butler is its ability to gather information from a variety of different online sources. This functionality is provided by the Personal Container technology developed previously by the authors [5].

A Personal Container is an advanced lifelogging system designed to capture and store a record of all the digital data generated throughout a user's lifetime. This data can come from a variety of different sources including: local applications

like Outlook and iPhoto; online services (e.g., Google or Facebook); as well as devices such as mobile phones.

Data gathering for the Personal Container is performed by plug-in software components. These plug-ins are programs (typically a few hundred lines of code) that target a particular data source, such as an IMAP email server. Plug-ins can also be written to upload and obtain information from social networking sites.

Information obtained by each plug-in is dumped into a database from where it can be searched, mined and accessed by other applications.

As with all systems that offer large-scale data collection and storage, there are privacy and security concerns that arise. Personal Containers address these issues by (1) using authentication and access control mechanisms to determine who can access the database and (2) supporting on-disk encryption of the database rendering it unreadable to those without the correct decryption keys.

III. The Privacy Butler

The Privacy Butler provides a wrapper around a Personal Container that enhances the functionality by: offering a storage and evaluation mechanism for presence owner policies, and including software extensions to modify online content. The basic operation of the service is shown in the flowchart of Figure 1.

The major functions performed by the Privacy Butler service include:

- Capturing the user's online presence by gathering information from online services and websites
- Detecting changes in that online presence and evaluating the changes with respect to the owner's policy
- Performing actions based on the results of the evaluation. This could include: (1) informing the user, and (2) removing or modifying the offending online content
- Learning from feedback provided by the presence owner

A. Capturing online presence

The Privacy Butler draws heavily on the data collecting capabilities of the underlying Personal Container framework in order to profile the owner's online presence. This Personal Container uses its plug-ins to gather information from different sources on the web and stores that information locally in a database.

Updates on the database (representing changes in online presence) are also converted by the Privacy Butler to facts. Facts are items of information about the world expressed in a form understandable by the system. For example, a new tag with ID 'tag132323' on a photograph (with ID 'photo1') on the Flickr service by user 'Bob' is converted into the two facts below:

```
tag('tag132323', 'Alice', 'Bob', 'photo1',
   'tag text').

service('photo1', 'Flickr').
```

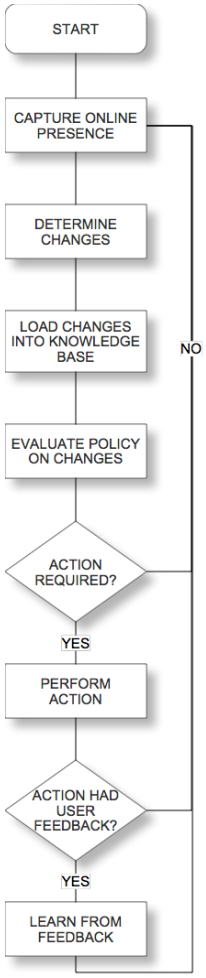

Fig. 1. Flowchart of Privacy Butler operation.

All new facts created by the Privacy Butler are loaded into a store (referred to as a knowledge base) and evaluated against the presence owner's policy.

B. Policy evaluation and resulting actions

In the Privacy Butler system, policies represent formal descriptions of the owner's wishes expressed using rules in the form of Horn clauses. These rules are structured with an *action* statement (i.e. the head of the clause) that specifies the action to be performed when the conditions, provided in the body of the clause, are satisfied. The head is separated from the body of the clause with a ":-". Four example rules are given below:

```
//Informs owners if a comment is placed on a
//photo on the Flickr social networking service
//and the comment is not authorized by the owner
inform(Owner, Photo, Comment_ID) :-
    comment(Comment_ID, Owner, Commenter,
      Photo, Comment_Text),
    service(Photo, 'Flickr'),
    not authorized(comment(Comment_ID, Owner,
      Commenter, Photo, Comment_Text)).
```

```
//Informs owners about a tag on Flickr that is
//not authorized by the owner
inform(Owner, Photo) :-
    tag(Tag_ID, Owner, Tagger, Photo, Tag_Text),
    service(Photo, 'Flickr'),
    not authorized(tag(Tag_ID, Owner, Tagger,
      Photo, Tag_Text)).

//Deletes tags marked for deletion
untag(Owner, Photo, Tag_ID) :-
    tag(Tag_ID, Owner, Tagger, Photo, Tag_Text),
    delete(tag(Tag_ID, Owner, Tagger, Photo,
      Tag_Text)).

//Removes comments marked for deletion
uncomment(Owner, Photo, Comment_ID) :-
    comment(Comment_ID, Owner, Commenter,
       Photo, Comment_Text),
    delete(comment(Comment_ID, Owner,
      Commenter, Photo, Comment_Text)).
```

It should be noted that our system considers all new content not posted directly by the presence owner to be unauthorized.

Policy evaluation results in actions being taken. Actions can be used to:

- Make modifications to online content
- Initiate communication with presence owners to report changes in online presence, request permission for a task or ask for further instruction. This communication can be performed using third-party services (e.g., Twitter, SMS or email)
- Make local modifications to the way the Privacy Butler operates (e.g., start logging comments posted to the owner's Facebook account)

These actions, specified in the action statements of rules, are implemented through system calls to plug-ins. In general, Privacy Butler can be easily expanded to support new actions by (1) specifying the rule for that action and (2) providing a plug-in to support the required actions.

C. Learning from presence owner feedback

Actions performed by presence owners giving permission or further instruction to the system are fed into a learning framework which can generate rules to automatically perform the actions. In the remainder of the paper we will refer to these as 'user actions' to distinguish them from actions as specified in Privacy Butler policy.

The learning framework performs automatic revision of the rules based on historical user actions. This can include adding new rules for special cases or rewriting existing rules to make them consistent with historical user actions.

A detailed description of the learning process is outside the scope of this paper, but is provided in Corapi *et al.* [6].

Support for this learning process within the Privacy Butler system is a significant benefit of the approach. It enables the system to start with a simple set of default rules. As user actions are fed back into the system these default rules can be revised. With sufficient historical user actions from which to learn, the system can operate with minimal intervention from the presence owner.

Manual editing of the rules is also supported for presence owners that require more control over the system. We are currently working on a Graphical User Interface to support this functionality.

IV. Implementation

In this section of the paper we present a scenario and then an implementation to provide the functionality described in the scenario.

A. Motivational Scenario

Alice returns from a vacation and uploads her holiday snaps to the Flickr website. Three of her work colleagues, Bob, Carol and David, add comments and tags to the newly uploaded photos. Prior to going on holiday, Alice had a confrontation with Bob. He is still sour about the event and his tags and comments are quite inappropriate especially since Alice's friends and family can also view them. We assume she is not simply able to unfriend Bob without socio-political ramifications.

Alice's Privacy Butler detects the modifications to the recently uploaded photographs. The Privacy Butler sends Alice an email informing her of these new tags and comments and asks which of the modifications should be authorized. Alice opts to permanently delete all modifications posted by Bob while keeping the others. The Privacy Butler performs these user actions and feeds Alice's choices back into the learning framework. The Privacy Butler adjusts its rules so that any future tags and comments on Alice's Flickr photographs made by Bob will be permanently deleted from Flickr.

B. Implementation and Evaluation

To provide the functionality described in the motivational scenario several software components were needed:

- A Flickr service plug-in to add/remove tags and comments on photographs
- A policy evaluation engine
- A user interface to inform the presence owner of changes in the online presence and request an action to take

For the evaluation four new user accounts were created on the Flickr website: Alice Example and Bob Example, Carol Example and David Example. The accounts of Bob, Carol and David were added to Alice's friends list enabling them to view her photographs as well as add tags and comments. Example photographs were then added to Alice's account. The Privacy Butler system was started and set to sweep Flickr every 5 minutes for changes. Each of the three work colleagues' accounts were then used to manually add tags and comments on the new photographs.

A Flickr service plug-in was developed in the Python programming language to interface with Flickr. This plug-in used a freely available [7] Python version of the Flickr API via which we could determine all photos in Alice's collection modified since the last sweep. This included the photos on which the work colleagues added tags and comments. These

were then dumped into the Personal Container database underlying the Privacy Butler. These updates were also converted into facts and fed into an expert system for evaluation. For brevity we only present the fact form of the tags and comments made on photo1 in Alice's photo album. These are shown below:

```
tag('tag_id1001', 'Alice', 'Bob', 'photo1',
  'tag1').
service('photo1', 'Flickr').

comment('comment_id1001', 'Alice',
  'Bob', 'photo1', 'comment1').
service('photo1', 'Flickr').

tag('tag_id1002', 'Alice', 'Carol', 'photo1',
  'tag2').
service('photo1', 'Flickr').

comment('comment_id1002', 'Alice', 'Carol',
  'photo1', 'comment2').
service('photo1', 'Flickr').

tag('tag_id1003', 'Alice', 'David', 'photo1',
  'tag3').
service('photo1', 'Flickr').

comment('comment_id1003', 'Alice', 'David',
  'photo1', 'comment3').
service('photo1', 'Flickr').
```

The Privacy Butler used a reasoning system implemented in Prolog to evaluate Alice's policy. The default rules provided to the system were those shown in Section III-B. Calls to our Python plug-in were made from within Prolog by using *system* calls.

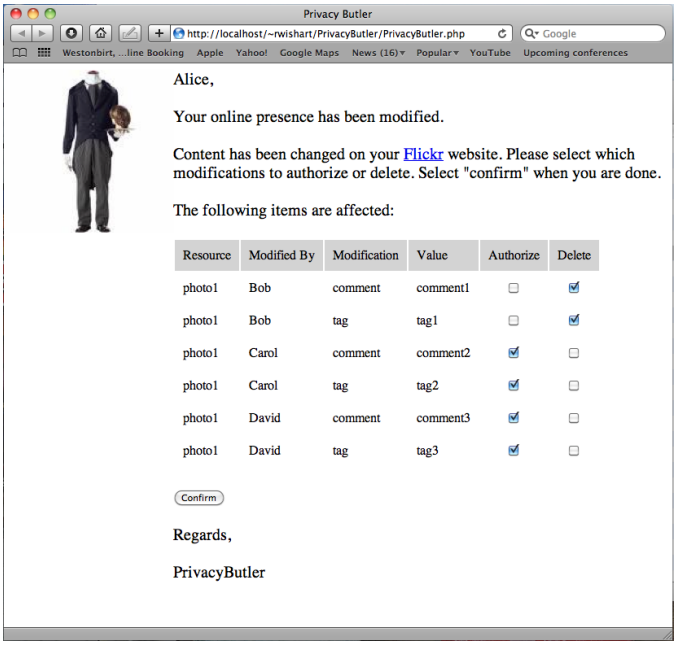

Fig. 2. Screenshot of the Privacy Butler requesting Alice to authorize recent modifications to her online presence.

For the purposes of this implementation, a PHP-generated webpage was used to inform Alice of the modifications to her Flickr photos and request authorization to keep the tags and comments. The Privacy Butler system communicated this webpage URL to Alice via email (orchestrated with the Flickr plug-in). This webpage is shown in Figure 2.

The user actions indicated by Alice via the webpage were then performed by the Flickr plug-in. In our scenario Alice deletes all the comments and tags added by Bob while keeping all the modifications by Carol and David. Representations of these user actions were converted into fact form and loaded into the learning framework.

In our example scenario the learning framework has only user actions in which all tags and comments from Bob were deleted. The framework identified the author of the tags and comments, Bob, as the common factor and added the following rules to the knowledge base:

```
untag(Owner, Photo, Tag_ID) :-
    tag(Tag_ID, Owner, Tagger, Photo, Tag_Text),
    Tagger = Bob.

uncomment(Owner, Photo, Comment_ID) :-
    comment(Comment_ID, Owner, Commenter, Photo,
       Comment_Text),
    Commenter = Bob.
```

The first rule states that tags from Bob should be automatically removed. Similarly, the second states that all comments from Bob should be removed.

If more historical user actions were available to the learning framework, such as other comments posted by Bob that were not deleted by the presence owner, the learning framework would seek to find the common link between the deleted comments that distinguished them from those not deleted. This could involve analysing the comment text for key words or phrases. Once the link is identified, an appropriate rule would be inserted into the knowledge base.

The simple implementation described here offers the functionality required by the scenario outlined above. In doing so, it demonstrates the utility of a Privacy Butler. It is important to remember that the Privacy Butler is also able to interact with other online services (such as Twitter or Facebook) provided the necessary plug-ins are available. Additionally, the system's behavior can be heavily customized. For example, Alice might prefer all unauthorized modifications of her online presence to be immediately removed and only reinstated once permission is given. Alternatively, she could set up the system so that she only receives notification of changes to her online presence that contain key words or phrases.

V. Related Work

The related work relevant to the Privacy Butler can be divided into two groups. The first addresses online presence and privacy while the second looks at lifelogging tools for users.

Krishnamurthy and Wills [8] developed a metric to measure the size of a user's online footprint. This metric value

was calculated based on the number of aggregator websites contacted, and how frequently particular aggregators (such as doubleclick.com) were contacted. The idea behind the work is that if one visits several websites that provide information to the same aggregator website, that aggregator can build a better, and thus potentially more invasive, profile of the user. The approach is limited in that (1) it applies only to web sites with embedded advertising that use aggregator services and (2) it offers no way for presence owners to change what is stored about them.

In the work of Bylund *et al.* [9] a Privacy Mirror is proposed. This system is intended to show users what information about them is available online. The approach is theoretically based on existing search engine technology (such as Google) and enables users to maintain and track the parts of their online information they deem relevant. While offering notice to users, the approach has no mechanism to respond to changes nor does it enable presence owners to actively go about shaping their online presence.

The Addict-o-matic [10] website enables users to search for terms across a range of websites and social networking services, providing an implementation of the ideas proposed in Byland *et al.*[9].

Google Alerts [11] is a tool intended for companies to track their online presence. To use the tool, companies register a callback email address as well as key search terms. Whenever new content is indexed by Google that matches the search terms, the company is sent an email on the callback email address. Similar technology could be used by users to provide notice of changes in their online presence, though it does not have access to online social networking sites (which typically require a login).

A webcrawler-based privacy evaluator was developed by Jensen *et al.* [12]. In their approach the iWatch tool is used to crawl the web searching for P3P [13] data handling policies. The approach could conceivably be extended to show users how the sites they interact with handle their data. As with the other related work listed so far, the approach does not facilitate management of the online presence by attempting to modify content.

In terms of the lifelogging functionality employed by Privacy Butler, there are several items of related work in the field. Due to space restrictions we only present two prominent items of work.

The MyLifeBits project conducted at Microsoft Research [14], [15], [16] examined ways in which a person's entire life experience could be digitized. The project yielded an easily searchable database of information fed by a variety of different software plug-ins. These plug-ins provided web history, voice and video recordings, email etc.

Lifelogging is also explored in the Semantic Logger project developed by Tuffield *et al.* [17]. Their approach is designed to aggregate personal information into a knowledge base that is accessible to context-based systems. Various plug-ins for the Logger enable it to collect data from user email, calendar appointments, geo-data and file system information.

A problem with the Lifelogging approaches is that they are more concerned with capturing their owner's digital content (e.g., emails, calendar appointments and photographs) than monitoring that content for changes by external parties.

As can be seen in this overview of related work, systems do exist that enable users to monitor their online presence. Such systems, however, are not capable of responding to those changes: users still have to manually interact with the online content to make any desired modifications.

The system that we propose in this paper, Privacy Butler, not only monitors online presence, but can also act on presence owner policy to effect changes in online content. A further benefit of our approach is that it is based on a learning framework able to learn from past user actions (i.e. what the presence owner did in similar previous situations) and modify the presence owner's policy to better suit their needs.

VI. Conclusion

A person's digital presence is comprised of the information available about that person on the Internet. This includes information from social services, blogs, online photograph albums (like Flickr), etc. In this paper we describe Privacy Butler, a service to monitor this information and shape it in accordance with policies specified by the presence owner. In doing so, we return *control* over the presence back to the owner and enable them to manage their online privacy by (1) providing awareness of the information available about them online, and (2) offering a means to automatically modify offensive or disagreeable content.

The Privacy Butler is supported by a Personal Container - a centralised store of user data. Various plug-ins to the Personal Container can be used to trawl the Internet (and social networking sites) for content mentioning the owner. The owner can then be notified of this content (with the decision to notify based on rules specified by the owner).

In this paper we have focussed on modifications to content hosted by social networking services as it is these services that offer APIs to manipulate user content. The openness of these APIs varies widely. For example, our choice of Flickr for the implementation was driven by the need to add or remove tags and comments from photographs. Such functionality is not provided in the current Facebook API, which permits only the addition of tags and not their removal. That said, it is expected that more functionality will become available in the future. This is supported by the recent expansions to the Facebook API that have introduced the ability to add and remove comments on user uploaded content. This trend of increasing openness will likely be followed by other social networking services as interoperability platforms like Google's Open Social[4] become more widely adopted.

Acknowledgment

This work was funded by EPSRC Grant EP/F024037/1 as part of the PRiMMA: Privacy Rights Management for Mobile Applications project.

[4]code.google.com/apis/opensocial/

REFERENCES

[1] D. Cohen. (2008, August) Wikipedia vandals target west australian politicians. online. Last referenced 2nd October 2009. [Online]. Available: http://www.news.com.au/story/0,,24260092-1245,00.html

[2] (2009, February) Facebook remark teenager is fired. online. BBC. [Online]. Available: http://news.bbc.co.uk/1/hi/england/essex/7914415.stm

[3] R. Minch, "Privacy issues in location-aware mobile devices," in *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS)*, vol. 5. IEEE Computer Society, 2004.

[4] B. Danah, "Why youth love social network sites: The role of networked publics in teenage social life," *Youth, Identity, and Digital Media*, pp. 119–142, 2008.

[5] A. Madhavapeddy, R. Wishart, and M. Sloman, "Personal containers," Department of Computing, Imperial College, London, Tech. Rep., 2009.

[6] D. Corapi, O. Ray, A. Russo, A. Bandara, and E. Lupu, "Learning rules from user behaviour," in *Proceedings of the 5th International Conference on Aritificial Intelligence Applications and Innovations (AIAI 2009)*, Thessaloniki, Greece, April 2009.

[7] S. Stvel. (2009) Python flickr apis. Last accessed 14th October, 2009. [Online]. Available: http://stuvel.eu/projects/flickrapi

[8] B. Krishnamurthy and C. E. Wills, "Generating a privacy footprint on the internet," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006, pp. 65–70.

[9] M. Bylund, J. Karlgren, F. Olsson, P. Sanches, and C.-H. Arvidsson, "Mirroring your web presence," in *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*. New York, NY, USA: ACM, 2008, pp. 87–90.

[10] (2008) Addict-o-matic.com. Last accessed 14th October, 2009. [Online]. Available: www.addictomatic.com

[11] G. Inc. (2009) Googlealerts. Last accessed 14th October, 2009. [Online]. Available: http://www.google.com/alerts

[12] C. Jensen, C. Sarkar, C. Jensen, and C. Potts, "Tracking website data-collection and privacy practices with the iwatch web crawler," in *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*. New York, NY, USA: ACM, 2007, pp. 29–40.

[13] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification," 2001, http://www.w3.org/TR/P3P/, last accessed 24 June 2007.

[14] G. Bell, "A personal digital store," *Commun. ACM*, vol. 44, no. 1, pp. 86–91, 2001.

[15] J. Gemmell, R. Lueder, and G. Bell, "The mylifebits lifetime store," in *ETP '03: Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*. New York, NY, USA: ACM, 2003, pp. 80–83.

[16] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell, "Passive capture and ensuing issues for a personal lifetime store," in *CARPE'04: Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*. New York, NY, USA: ACM, 2004, pp. 48–55.

[17] M. M. Tuffield, A. Loizou, and D. Dupplaw, "The semantic logger: supporting service building from personal context," in *CARPE '06: Proceedings of the 3rd ACM workshop on Continuous archival and retrival of personal experences*. New York, NY, USA: ACM, 2006, pp. 55–64.