Privacy-Enhanced Sharing of Personal Content on the Web*

Mohammad Mannan, Paul C. van Oorschot
School of Computer Science, Carleton University
Ottawa, Ontario, Canada
{mmannan, paulv}@scs.carleton.ca

ABSTRACT

Publishing personal content on the web is gaining increased popularity with dramatic growth in social networking websites, and availability of cheap personal domain names and hosting services. Although the Internet enables easy publishing of any content intended to be generally accessible, restricting personal content to a selected group of contacts is more difficult. Social networking websites partially enable users to restrict access to a selected group of users of the same network by explicitly creating a "friends' list." While this limited restriction supports users' privacy on those (few) selected websites, personal websites must still largely be protected manually by sharing passwords or obscure links. Our focus is the general problem of privacy-enabled web content sharing from any user-chosen web server. By leveraging the existing "circle of trust" in popular Instant Messaging (IM) networks, we propose a scheme called IM-based Privacy-Enhanced Content Sharing (`IMPECS`) for personal web content sharing. IMPECS enables a publishing user's personal data to be accessible only to her IM contacts. A user can put her personal web page on any web server she wants (vs. being restricted to a specific social networking website), and maintain privacy of her content without requiring site-specific passwords. Our prototype of IMPECS required only minor modifications to an IM server, and PHP scripts on a web server. The general idea behind IMPECS extends beyond IM and IM circles of trust; any equivalent scheme, (ideally) containing pre-arranged groups, could similarly be leveraged.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication, Unauthorized access*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, Human Factors

Keywords

access control, sharing, circle of trust, privacy

*Version: February 24, 2008.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

1. INTRODUCTION

Through social networking and photo-sharing websites, and personal blogs, it is becoming increasingly popular to make personal content available on the Internet. For some users, these sites provide a textual and/or pictorial documentary of life. Primarily because it is the easiest mode of operation, many users of these services allow their personal web content to be accessed by all other Internet users, often with the false impression that none other than their family or friends would look into their personal online posts [29]. Privacy concerns are largely being ignored (sometimes unknowingly) in the current rush to online *lifecasting*.

Social networking websites such as Facebook and MySpace provide access control mechanisms for partially restricting personal content to a known circle of contacts; photo-sharing websites such as Flickr and Shutterfly provide similar mechanisms. A user can invite her friends and family to be added to her *permitted list*, and can authorize only such people to view her web content, but only if they create accounts at the publishing user's social networking site. Although users reportedly disclose personal data in abundance at these social networking sites, a relatively small number of users limit access to their profiles only to a friends' circle; several studies provide evidence of such behaviour [21, 29, 39, 51]. While this limited restriction might help users' privacy, this applies only for the content on those (few) sites.

We focus on the general problem of privacy-enabled web content sharing from any user-chosen web server. Many users now own domain names for hosting personal websites, facilitated by the very low price; as of October 2007, a top-level domain name may cost less than $6/year, with $4/month commercial hosting fees. Most ISPs also offer free web spaces for home users. It is thus cheap and easy to make any personal data available to anyone around the globe through a website; however, restricting such content to a selected group of people is more difficult. Currently this is achieved primarily by either (i) advertising an obscure link through personal email, i.e., a URL which is not linked from any other web page; or (ii) protecting a web page with a password, and distributing that password among chosen contacts through email, instant messaging (IM), or phone.

Emailing an obscure URL to many contacts (friends and family members) is a rather cumbersome approach, especially if the shared URLs are often updated. Password protection (e.g. HTTP Authentication [18], forcing a login dialogue/page) is not uncommon among the more technically inclined, but this leads to yet one more password to share and maintain, and once a password is shared with some-

one, the access grant cannot be retracted without changing the password (which also requires distributing the new password to all other contacts). Also, anyone who learns the shared password can view the protected content without the publishing user's consent; anyone knowing the password can pass it on to others, and such transitive access is not generally preventable.

Relying on the immense popularity of public instant messaging (IM) networks,[1] we propose a scheme called IM-based Privacy-Enhanced Content Sharing (`IMPECS`) to disseminate personal web content by leveraging the established "circle of trust" on IM networks. We assume both publishing and viewing users can, or already do use IM. A user's web content can be viewed only by her IM contacts. Further restrictions can be applied depending on which group of users (e.g. family, friends, co-workers) a specific contact is placed in by a *publishing user*, i.e., one who originally makes personal content available for her IM contacts. A *viewing user* is one who wants to view such content. We assume that a web server and an IM server share a user-specific content sharing key; a 'ticket' (similar to a session cookie) is generated by the IM server for a viewing user using the content key of a publishing user, and the web server validates the ticket before serving data from a user's web folder (cf. Kerberos [31]). Our primary goal is to enhance privacy (i.e. confidentiality) of users' personal web content; we do not aim for very high-end or military-grade security, as the security of IMPECS is limited by the underlying IM and web communication protocols, which in current practice transfer most content in plaintext although authentication passwords are generally sent over SSL (cf. [26, 11]). The main intended feature of IMPECS is that total strangers are precluded from direct access to a user's personal web content, but "friends" as designated by the user's IM contact list are allowed access (without requiring any special shared password). IMPECS also prevents large-scale web crawlers and auto-indexers from tagging personal data and pictures (see e.g. [4, 27]). However, malicious IM contacts of a publishing user may of course re-post the user's private content to a public web forum, and we are not proposing any form of digital rights management (DRM) control.

In summary, our proposal for privacy-enhanced personal web content sharing offers the following features and benefits.

1. Privacy-Enhanced Sharing. A publishing user's personal web content can be viewed only by the IM contacts that she pre-approves. Thus privacy of a user's web content is restricted to a designated group. For many existing IM users, such groups can be leveraged without additional setup costs.

2. Usable Security. The privacy enhancement does not require a viewing user to separately update his IM client, or remember the publishing user's URL, or have access to a site-specific password to view the publisher's content. Similarly, the publishing user need not carry out any extra steps beyond existing management of an IM contact list, although finer granularity lists can optionally be created by advanced users.

3. Interoperability. In contrast to social networking websites, a user can publish her web content at any web server of her choice, and yet be able to maintain greater access control on her content.

4. Decreased Risks Related to Sharing. By restricting open access to personal details, IMPECS reduces opportunities for launching context-aware, targeted phishing attacks [30, 44, 50].

5. Protection Against Web Server Compromise. A variant of IMPECS (Section 4) can prevent en masse *drive-by-downloads* [36, 47] as currently being enabled by the compromise of a hosting provider with a large number of customers.

[1]For example, according to one estimation [6], there are about 350 million user accounts in MSN and Yahoo! IM networks in total.

To test our design, we built a prototype of IMPECS using the IETF standardized Extensible Messaging and Presence Protocol (XMPP [40, 41], i.e., the Jabber IM protocol). This required only minor modifications to the IM server, and PHP scripts on a web server. Our implementation source code is available on request.

Organization. In Section 2, we discuss the proposed IMPECS scheme, threat model and operational assumptions. Our prototype implementation is discussed in Section 3, along with brief comments on deployment issues. A variant of IMPECS is discussed in Section 4. Section 5 provides further motivation, an overview of existing and proposed work related to personal content sharing, and a comparison of IMPECS with these in terms of user convenience and usability. Section 6 concludes.

2. IM-BASED PRIVACY-ENHANCED CONTENT SHARING (IMPECS)

In this section, we describe the proposed IMPECS scheme, threat model and operational assumptions. Table 1 summarizes our notation. We assume readers are familiar with basic IM definitions such as *presence* and *contact list* (e.g. see [25]).

A, B	Two IM users Alice and Bob, both members of each other's respective contact lists. A is the publishing user; B is the viewing user.
S_i, S_w	IM and web servers, respectively. Both A and B have accounts with S_i, and A maintains an account with S_w.
ID_{Aw}	A's user ID at S_w (unique in S_w's domain).
K_{Aw}	A's content sharing key, shared with both S_w and S_i.
$\{data\}_K$	Authenticated encryption [20, 10] of *data* using symmetric key K.
URL_A	The URL of A's publishing web folder on S_w.
R	Access restrictions on URL_A as imposed by A.
T_{iw}	An access control ticket for viewing URL_A (generated by S_i, and validated by S_w).
URL_{AR}	A 'registration' URL generated by S_w when requested by A. The content sharing key and restrictions are shared between S_w and S_i through this URL.
URL_{AT}	A 'viewing' URL (for accessing URL_A) containing a ticket T_{iw}, generated by S_i at the request of B.

Table 1: Notation used in IMPECS

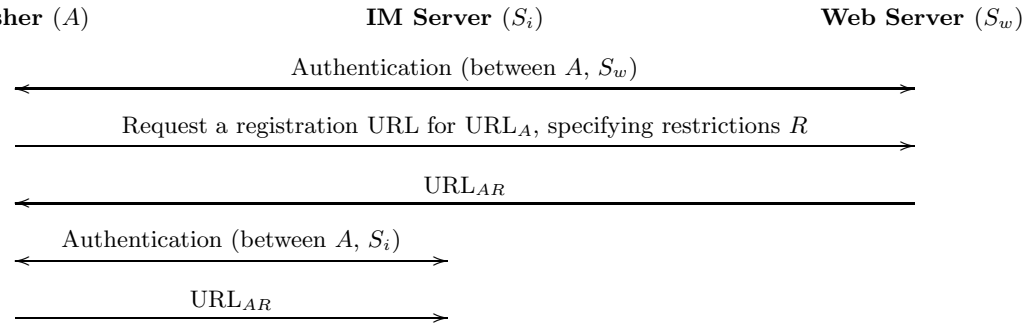

Figure 1: Registering a URL in IMPECS

Overview of IMPECS. Assume user A maintains a website on a web server S_w. A registers her site with an IM server S_i, and sets permission for the site, e.g., which contacts can access which pages/folders. For example, contacts in the group "friends" may have different permissions than the group "family." S_w and S_i share a user-specific content sharing key for A. IM contacts of A can see (through their IM clients) whether A offers any personal URL which they are permitted to view. When a contact B wants to visit A's advertised personal website (or any pages thereon), B sends a request to S_i to visit the website. Depending on restrictions R (e.g. duration, frequency) for viewing web pages at URL$_A$, S_i generates a 'ticket' (similar to a session cookie), and sends a special URL to B along with the ticket. B receives the URL instantly (e.g. as an IM text message) from S_i, and can visit URL$_A$ within a time period as specified by the ticket. Note that A need not be online to provide this permission. We now describe the scheme in greater detail.

Setup. A and B are two IM users who maintain IM accounts at the same IM server S_i. (Note that A and B may use different IM servers, as long as their IM servers facilitate communication between the users, e.g., as in distributed XMPP [40], Windows Live/Yahoo! Messenger.) Both users have added each other into their contact lists; adding someone to a contact list requires explicit permission from the user being added (a common practice in most IM networks). A also puts B into an appropriate group of her contact list (e.g. "family", "friends", "co-workers"). A maintains an account with a web server S_w, and uploads some personal pictures or files under a web folder URL$_A$ at S_w. A wants to share URL$_A$ with a select group of IM contacts including B.

Registering a URL with the IM server. We now describe the steps for publishing a content-hosting URL in IMPECS. Figure 1 outlines the following steps.

1. A logs into S_w (e.g. using a pre-established password over SSL).
2. A uploads her personal files and sets restrictions on URL$_A$, e.g., the length of time a ticket will remain valid after being generated by S_i (using e.g. HTML check-boxes or drop-down lists). A then requests S_w to generate a registration URL for URL$_A$.
3. S_w generates a random content sharing key K_{Aw} (e.g. 128 bits, sufficient for precluding offline dictionary attacks) and stores it in a protected database, or in a file under A's private space. S_w constructs the registration URL, URL$_{AR}$ = `http://<URL`$_A$`>/?userid=`ID_{Aw}`&key=`K_{Aw}`&restrictions=`R, and sends URL$_{AR}$ to A (e.g. through HTTPS). Here, by <URL$_A$> we mean the actual URL (without the 'scheme name'), not a label for that URL (i.e. not the string "URL$_A$").
4. A logs into S_i (e.g. using her regular IM password over SSL).
5. A forwards URL$_{AR}$ to S_i, for the purpose of registering this information with S_i. S_i stores URL$_A$, ID_{Aw}, K_{Aw} and R for future ticket generation.

Viewing a protected URL via an IM server. We now describe the steps for viewing a content-hosting URL in IMPECS. Figure 2 outlines these steps.

1. B logs into S_i (e.g. using his regular IM password over SSL), and receives his contact list as usual in IM. As part of IMPECS, B also receives a list of private URLs, offered by his contacts, which are authorized to be accessed by B.
2. B sends a request to S_i for a ticket to view one of these URLs, say A's web content at URL$_A$.
3. S_i generates a ticket $T_{iw} = \{ID_{Aw}, R\}_{K_{Aw}}$, constructs URL$_{AT}$ = `http://<URL`$_A$`>/?userid=`ID_{Aw}`&ticket=`T_{iw}, and sends URL$_{AT}$ to B.
4. B forwards URL$_{AT}$ to S_w. S_w retrieves K_{Aw} using ID_{Aw} as embedded in B's request. Then S_w decrypts the ticket T_{iw}, and compares whether A's user ID in the URL is the same as inside the ticket. S_w also checks the restrictions; e.g., R could be as simple as a timestamp, in which case S_i encrypts the current timestamp into the ticket and S_w accepts that ticket if received within a specific time period (e.g. 60 seconds, as set by A).
5. S_w sends the content hosted at URL$_A$ to B after validating B's ticket T_{iw} in URL$_{AT}$ (as in step 4). If a valid ticket is not supplied, S_w denies access to URL$_A$.

Caveats. A malicious user B can compromise the privacy of content hosted at URL$_A$, by making a copy of the website and posting it on a publicly accessible site, or sending a valid ticket to anyone B wants. Although A cannot stop

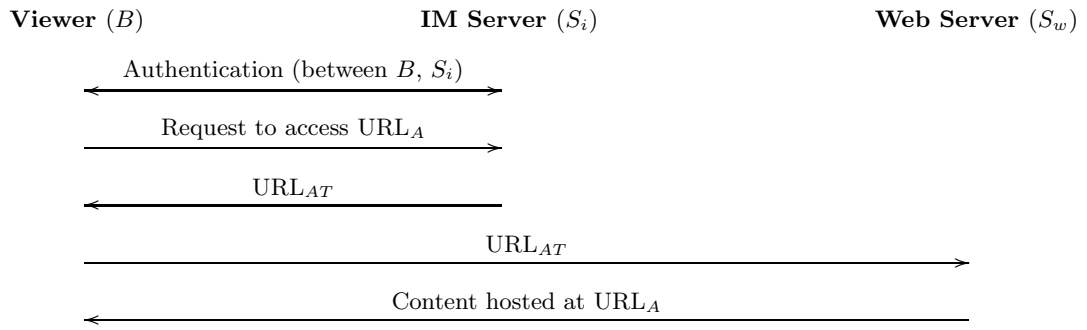

Figure 2: Viewing a personal URL in IMPECS

copying of her personal content, she may limit (to some extent) forwarding of a valid ticket with the help of S_i and S_w in the following way. S_i can encrypt B's current IP address into the ticket, and S_w can check whether it receives the ticket from the specified IP address as embedded inside the ticket (assuming both S_i and S_w have access to the same IP address of B).

If a content key K_{Aw} is leaked, anyone can generate valid tickets with that key, and thus compromise the privacy of content hosted at URL$_A$. If A changes her content key K_{Aw}, this threat can be minimized. Note that A's modifications to her web content, and key updates, are transparent to viewing users. Although valid tickets can be generated with a compromised K_{Aw}, this key does not enable access to modify A's content on S_w.

Most IM and web accounts are currently authenticated by user-chosen (generally *weak*) passwords. A compromised IM account enables an attacker to add any malicious link (as personal URLs) to that account. A compromised web account enables an attacker to post any content on the compromised user's web space, and modify content keys (although he cannot update the content key at S_i). However, these threats exist currently for both IM and web accounts; IMPECS does not increase these existing risks nor does it attempt to address them.

If user content is distributed across many different hosting sites (rather than being concentrated only to few sites as in current social networking sites), then an adversary cannot easily track users by collecting their personal web content from only a few selected sites. However, in IMPECS if the IM server S_i is compromised (or cooperates with the adversary), privacy of user content is lost for all IMPECS users of S_i even if their content is hosted at different providers; from compromised content keys, anyone can generate valid tickets for accessing user data. Thus the IM server is a potential single point of privacy breach (if compromised or hostile).

If attackers can compromise the web server of a publishing user A, they can display whatever content they want from A's site, or spread malware to users visiting the site [36]. Compromise of a web server that hosts content from a large number of users is particularly more risky, and has been reported in the past (e.g. [47]). We briefly outline a a variant of IMPECS to mitigate such a large scale compromise in Section 4.

Threat model and operational assumptions. We assume that the circle of trust as built into IM networks is reliable, i.e., a viewing user is not malicious. A publishing user A cannot be added to anyone's contact list without being explicitly approved by A (as is the common practice in most IM networks). To achieve fine-grained access control, we also assume that a publishing user groups contacts appropriately, and authorizes access to these groups conscientiously (e.g. which group can access which URLs). IMPECS trusts that the IM server checks publishing user A's permissions properly, and only sends tickets to authorized users. The web server is trusted to deliver A's content only after validating an appropriate access control ticket. The availability of usable site maintenance tools (e.g. HTML editing, file uploading) is also assumed for publishing users.

If a publishing user A's IM client offers a user interface for setting a personal URL (which is the norm in many IM clients, e.g., Yahoo! Messenger), we can use that to send the registration URL (containing the content key and restrictions), and thus may avoid changing A's IM client. A viewing user B's IM client can also remain the same if it offers viewing IM contacts' personal URLs (e.g. the 'View Profile' option in Yahoo! Messenger provides a 'Home Page' field in a profile webpage). We require only minor modifications to a web server through server-side scripts (assuming the server allows such scripts). The web server may optionally maintain a database of user-specific content keys; otherwise, the content key of a user must be stored in the user's private space on that web server. For an IM server, enforcing restrictions (in ticket generation) is easy; the server already restricts text (and other request) messages sent to a user from any other IM users according to the receiving user's preferences. However, users must register their URLs with the IM server; most IM services currently enable users to register personal URLs on their profiles. Leaking these URLs (without the corresponding content keys) will not by itself authorize access to any web content; they are inaccessible unless someone gets a valid ticket from the IM server.

Communication in most public IM networks (client-server and client-client) and web servers (client-server) is plaintext, although a password for authentication is generally sent over SSL. Note that our design involves the content key K_{Aw} (i.e. URL$_{AR}$) being sent over SSL. An attacker with access to the communication link may eavesdrop on private content of a user when the user uploads content to the web server, or when content is served to a (valid) viewing user. Using a variant of IMPECS (see Section 4), or at the added cost of SSL, these attacks can be addressed.

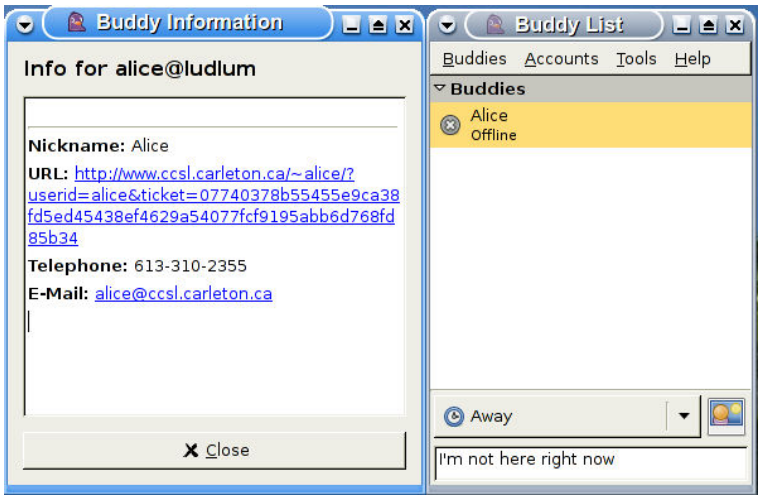

Figure 3: A viewing URL instance in IMPECS

3. IMPLEMENTATION

In this section we discuss our prototype implementation, and computational and deployment costs of IMPECS.

We implemented a prototype of IMPECS using the Extensible Messaging and Presence Protocol (XMPP [40, 41], based on the popular Jabber[2] IM protocol). As XMPP server and client, we chose `jabberd2` [22] and `Pidgin` [34] (previously known as `Gaim`) respectively, on a Linux platform. For cryptographic library, we use `OpenSSL` and the PHP `mcrypt` module; we use `AES-CBC-128` for symmetric encryption, and `/dev/urandom` for random number generation. `MySQL` is used for database support. Our implementation source code for the prototype is available on request.

We assume that the publishing user A can run PHP scripts on the web server S_w. S_w also stores A's content sharing key in a database. We create a web folder for A on S_w which is accessible for writing (and viewing) when A logs into S_w. Other than login as A, for viewing any content of the folder, one must supply a ticket containing a valid timestamp (and ID_{Aw}) encrypted under A's content key. We assume that system clocks of S_i and S_w are (more or less) synchronized. S_w checks whether a requesting URL contains a valid ticket; we accept a timestamp to be valid if it arrives within 60 seconds of being generated by S_i. A and B also add each other to their respective contact lists.

XMPP uses the `vCard` [15] format for personal profile information storage, which facilitates advertising one's personal URL. We use this field in vCard for storing a user-specified URL, and added one field called `content-key` into the vCard table for storing a user's content sharing key (along with ID_{Aw}).[3] Ideally an XMPP user can set vCard values from any XMPP client. However, as the Pidgin implementation we used (version 2.0.1) lacks any such user interface for setting vCard values, we directly inserted URL_A and K_{Aw} to A's vCard table on the jabberd2 server database. For viewing a contact's vCard, a user can select the contact from the Pidgin contact list, and choose the "Get Info" option from the context menu. When S_i receives such a request for A's vCard from B, S_i retrieves A's content key K_{Aw}, and generates a ticket by encrypting the current time and ID_{Aw} with the key. S_i then constructs a URL using URL_A as the base, and ID_{Aw} and the (hexadecimal encoded) ticket as parameters. Figure 3 shows one example of S_i's response to B. Then B can click on the link and be able to view URL_A, if validated by S_w.

Computational and deployment costs. In addition to retrieving A's vCard information from a database (as required by a regular XMPP server), IMPECS requires one symmetric-key encryption by S_i. One symmetric-key decryption is required by S_w when a viewing URL is received (for ticket validation). S_w also must generate a 128-bit random number when A requests a registration URL (for the content key generation). These operations are relatively light-weight for the IM and web servers; no practical deployment barrier in terms of performance is expected.

In a distributed IM service such as XMPP or Windows Live/Yahoo! Messenger, where A and B may have accounts with different IM servers, IMPECS does not require any changes to B's server or client software. (Note that as of Feb. 2008, XMPP is supported by several large IM services, e.g., Google Talk, IBM Lotus Sametime, and AOL/ICQ.) We require changes to A's IM and web servers. The changes in S_w are mostly achieved through PHP scripts. A's content key and restrictions can be stored in a file under a private folder (on A's web space), or in a database if S_w provides database access. Also, B remains anonymous to S_w in IMPECS; i.e., B does not need an account at S_w for viewing A's content, as opposed to social networking websites (although a ticket is required in IMPECS). Note that all publishing users at S_w can reuse the same PHP scripts for our scheme; i.e., users are not required to write or modify the PHP scripts (these scripts may be provided by, e.g., S_w or the open-source community).

Why not to implement IMPECS as a Facebook application. For ease of deployment, we could implement IMPECS in Facebook Platform[4] or Google OpenSocial.[5] Instead we chose to base our IMPECS design and implementa-

[2] `www.jabber.org`

[3] Instead of inserting the content-key field, ID_{Aw} and K_{Aw} could be embedded into the URL field, allowing S_i to remain in conformance with the vCard standard.

[4] `http://developers.facebook.com/`

[5] `http://code.google.com/apis/opensocial/`

tion on IM for the following reason. We believe that storing relationship information and user data at the same site may undermine privacy; for example, a single entity then learns too much about users and may use that knowledge to launch unfriendly (in regard to users' privacy) campaigns such as targeted advertisements, sharing user data with government agencies and third-party businesses. This also makes such sites an attractive target to compromise. These threats are quite evident from the short history of Facebook and MySpace. IM networks have also been targeted for malicious purposes such as spreading worms and phishing URLs; however, such attacks generally compromise relationship information (i.e. email addresses) but not user content.

4. A VARIANT OF IMPECS

In this section, we briefly outline a variant of IMPECS that can prevent malware-spread from a compromised web hosting provider. We have not implemented this variant yet.

Some large hosting providers (e.g. `godaddy.com`) currently facilitate web hosting for thousands of personal and corporate sites. If many IMPECS users host their content at such a provider, a successful attack against the provider might possibly affect all those IMPECS users. The compromised user sites could be used for malicious purposes, e.g., hosting malware for *drive-by-downloads* [36, 47]. This could be particularly bad for IMPECS users as private URLs as shared through IMPECS may appear to be more trustworthy. Here we outline a proposal that can guard against such en masse exploits.

Additional steps during URL registration. The following additional steps are required from a publishing user.

1. A uses a local application (in-browser JavaScript plug-in or an independent content editing application) to generate an encryption key K_{enc}, 128 bits long. A then uses K_{enc} to encrypt her personal files and upload the result (i.e. $\{datafiles\}_{K_{enc}}$) to the web server S_w. This is done at the beginning of step 2 in URL registration of IMPECS (see Fig. 1 in Section 2).

2. A appends K_{enc} to the registration URL received from S_w before sending the URL to the IM server S_i. This is done at the end of step 3 in URL registration of IMPECS (see Fig. 1 in Section 2).

Additional steps for a viewing user. The following additional steps (although transparent) are required from a viewing user.

1. When S_i generates URL_{AT} (step 3 in Fig. 2; see Section 2), it also appends the URL with K_{enc} as a URL fragment, i.e., $<\text{URL}_{AT}>\#K_{enc}$. When B visits this URL, URL_{AT} is forwarded to S_w but not the fragment, i.e., S_w does not receive K_{enc} (cf. [2]).

2. In step 5 (see Fig. 2 in Section 2), S_w sends the requested (encrypted) content. B's browser uses K_{enc} as received from S_i to display the decrypted content.

The encryption key K_{enc} is not accessible to S_w at any time. Thus by compromising S_w, an attacker cannot control what is served to the visiting IMPECS users. Note, however, that regular visitors to such a site are not protected by this technique. The publishing user A may update K_{enc} in a similar way to the content key K_{Aw}. However, an update to K_{enc} does not mandate updating K_{Aw} or vice-versa, and both key updates are transparent to viewing users.

5. MOTIVATION, RELATED WORK AND COMPARISON TO IMPECS

In this section we discuss existing and proposed work related to personal web publishing, and contrast the IMPECS scheme with these in terms of privacy and user convenience.

Popular IM networks, e.g., Yahoo!, AOL, and Windows Live enable users to maintain a profile accessible as a webpage. Microsoft offers free web spaces for sharing personal web content (e.g. profile, photos, blogs, guestbook) through its Windows Live Spaces social networking website at `www.spaces.live.com`. Live Spaces is integrated with the Windows Live Messenger IM client. User A can control who may view her Live Spaces' webpage. A can invite friends to join the Windows Live Messenger network to view her content. A may authorize only her IM contacts (or a subset of the contacts) to view her space. Alternatively, A may make her space accessible to anyone on the web. If A's space is restricted to IM contacts, a contact B (from A's contact list) can login to Live Spaces using B's Windows Live Messenger login credential for viewing A's space. If logged into the IM network, B can also select A's profile from a context menu from the Live Messenger client; from A's profile, B can access A's space without further authentication. Yahoo! is also extending its IM service to offer a social networking site called Mash (`mash.yahoo.com`).[6] However, in either case, similar to the common social networking practice (e.g. as in Facebook or MySpace), B must join A's network to view any access-restricted content. In contrast, when using IMPECS, B does not need to know where his (IMPECS-enabled) IM contacts host their content.

To partially relieve users from the necessity of creating multiple web credentials, Microsoft permits third-party businesses to use its Windows Live ID Web Authentication[7] (previously known as Microsoft Passport). Similarly, Yahoo! offers the Browser-Based Authentication[8] (BBAuth) service that enables third-party web applications to be authenticated through widely used Yahoo! IDs. OpenID (`openid.net`) is an initiative from the open source community to unify online authentication, also reducing the burden of creating multiple web credentials. AOL has enabled the use of OpenID (through `openid.aol.com`) for its IM service and AOL Pages social network. OpenID can also be used for Yahoo! login (through `openid.yahoo.com`). Liberty Alliance (`projectliberty.org`) is another 'holistic' approach to establish an open standard for online identity. If any such unified identification framework becomes widely accepted in the long-run, IMPECS would become even more appealing (e.g. through a common login credential). However, IMPECS does not address user authentication across websites per se, but rather focuses on how the existing trust network and interactiveness of a popular service like IM can be leveraged to offer privacy-enhanced personal content sharing on the web.

[6] As of Feb. 10, 2008, this is an invitation-only 'beta' service.
[7] `http://msdn2.microsoft.com/en-us/library/bb676633.aspx`
[8] `http://developer.yahoo.com/auth/`

Most IM networks offer file sharing from user machines generally through custom-built file transfer protocols. An IM user can restrict which contacts in her IM contact list can access the shared files. However, IM file transfer protocols may not work in some cases (e.g. due to firewall restrictions), and a publishing user must be online to make her files available to others.

YouServ [9] is an end-user P2P application designed by IBM to enable people to easily share personal content (e.g. photos, music, presentations, work documents) with little to no cost.[9] Instead of a specialized P2P protocol, all YouServ content is served through standard web protocols (i.e. DNS with HTTP). An implementation of YouServ was used by thousands of users internally at IBM and Carnegie Mellon University (apparently the web interface for this service at `YouServ.com` is now defunct). YouServ requires two centralized components called YouServ Coordinator (for authentication and peer coordination) and YouServ Dynamic DNS (for finding a peer site's dynamic IP address). A user's YouServ content remains available even when the user's PC is offline (through a peer hosted site), or firewalled (through a proxy site). Authentication is provided using a single sign-on password scheme (valid for any YouServ site). Access to any specific file can be limited to certain members of the YouServ community. Using YouServ, Bayardo et al. [8] proposed a technique to make IM file transfer easier by making local files available through transient web links; the web link of a file is sent to the recipient simply as an IM text message. In contrast to YouServ, publishing users in IMPECS make their personal content available from a third-party hosting site (as is the current common practice) instead of their own PC (or any of their peers' PC).

The popularity of social networking websites, e.g., Facebook, MySpace, Twitter, Bebo, is apparently comparable to the early years of large-scale IM networks. By joining Facebook or MySpace, users can search and connect with friends, share personal content such as photos, videos, blogs, contact information, and preferences. In Facebook, users generally locate friends from groups, e.g., classmates from the same school or university, co-workers, geographical locations. MySpace generally categorizes user groups by interests, e.g., music, photography. To add to the interactive power of IM, MySpace offers its own IM client called MySpaceIM (accessible only to MySpace users). Facebook also has recently (Oct. 2007) added IM capability through the FriendVox browser-based IM client. Twitter enables users to send short messages to selected friends through the web, SMS messages, or IM. Most social networking sites enable limited access control through explicitly creating a "friends' list." Online photo sharing website Flickr offers creation of a list of friends through Yahoo! login credentials. Other photo-sharing websites such as Shutterfly offer similar privacy-enhancing mechanisms. We discuss the effectiveness of such access control mechanisms below.

Privacy issues in social networking websites. Although social networking sites enable publishing users to partially restrict access to their personal content, privacy concerns are emerging quickly regarding the use of these networks. People have been denied or lost jobs because of their comments on MySpace or Facebook profiles (e.g. [32, 33]), a grocery chain dismissed employees for comments on Facebook (e.g. [19]), and students were suspended for their Facebook comments (e.g. [13]). Government agencies such as the CIA are suspected of tracking users with special interests (e.g. [35]); apparently under the U.S. Patriot Act, state agencies can look into a job interviewee's Facebook profile, even if the profile is "privacy-protected," i.e., permitted to be viewed only by the publisher's circle of friends (e.g. [28]). If a user removes content from his/her profile that may be deemed offensive or was posted as a momentary emotional response, or even if the user deletes the entire profile, personal content may still reside in (incremental) archives for a long time (cf. [27]).

Many users of social networking sites keep their profiles and friends list publicly accessible. A user survey [29] of social networking websites reported that 74% of adult users of those sites exposed their personal information such as email address, name, birthday, home and work address, and even Social Security Number (SSN). Only 39% of respondents chose to restrict their personal profiles only to friends. Initial results from another survey [48] of Facebook users reported that 67% of the participants kept their personal profile open for all. Another study [37] of the LinkedIn social networking website (used mostly for business purposes, e.g., to find potential clients, service providers, business opportunities, job listings) reported that people generally expose detailed and (possibly) confidential information on their profiles. Dwyer et al. [16] compared information disclosure and perceptions of trust and privacy in an online survey of Facebook and MySpace users. Facebook users were reported to reveal more identifying information than MySpace users. For example, real name, email address, and IM screen name have been disclosed by 100%, 94%, and 71% of Facebook users respectively (in contrast to 66.7%, 40%, and 49.8% of MySpace users respectively).

Gross and Acquisti [21] investigated patterns of personal information revelation and associated privacy implications using more than 4,000 publicly available Carnegie Mellon University (CMU) users' Facebook profiles. Most users provided (seemingly highly accurate) personal information including profile image, full birth date, hometown, current residence, and phone number. Personal preferences, interests, and political views were also disclosed by the majority of CMU users. Although Facebook offers privacy control, most users did not change the default privacy preferences which grant access to a user's full profile by any member of the user's groups/networks (e.g. place, institution, interest); only three CMU users' profiles (0.06%) were precluded from view by *unconnected* users (i.e. not a friend or friend-of-a-friend). Based on the revealed personal information, the authors outlined a number of privacy implications including online and real-world stalking, digital dossier of participants (by any third-party), and demographics and face re-identification (i.e. relating seemingly anonymous data to explicitly identifying information). The authors also discussed how a user's SSN may be estimated from disclosed birth date, hometown, current residence and phone number. A similar study [17] on 20,000 MySpace user-profiles reported that 68% of users kept their personal profiles open for all. Almost half of a randomly selected 1000 users' group provided global access to all elements of their personal profile. Rosenblum [39] analyzed privacy risks of social networking

[9]Note that when this research [9] was published in 2002, the cost of hosting a personal website at a third-party hosting company was much higher than today.

sites, including privacy options as provided by major networking sites and limitations of such privacy settings. In addition to highlighting privacy issues of social networking sites, Barnes [7] emphasizes that a significant educational effort from parents, schools, social networking sites, and government agencies, is required to address the emerging privacy issues related to these sites.

Jagatic et al. [23] collected publicly available "circles of friends" data from several social networking websites by using web crawlers; this enabled the researchers to quickly build a database of tens of thousands of relationships. When a (benign) phishing attack was launched by using the collected social network database, 72% of social networking targets fell victim to the phishing attack, while only 16% of regular users were fooled by the attack. In fact, social networking websites are specifically being targeted for launching *context-aware* phishing attacks (see e.g. [30, 44, 5, 50]), spreading spyware [12] and malware [45], and even for building botnets [43]. Cross-site scripting flaws in the MySpace website have been reported [49] in the past which could have been exploited to disclose even privacy-protected user content. Social networking websites with personal details of millions of users would also seem to be lucrative targets to online attackers (e.g. for targeted phishing or identity theft), and government agencies (e.g. for tracking citizens' digital identities). Equifax, a leading consumer credit reporting firm, has recently (July, 2007) warned [38] that user profiles on social networking sites are a "goldmine" for ID thieves. MySpace acknowledged [1] that as of July 2007, it had removed more than 29,000 registered sex offenders profiles from the MySpace website, indicating that criminals with other than monetary motives are also exploiting the abundance of personal information freely available at social networking sites.

Ahern et al. [3] examined privacy decisions in mobile and online photo sharing using Flickr. Most interviewed users in the study showed little or no concern regarding exposure of aggregated contextual information, e.g., time, location (embedded with some uploaded photo files), arising from their photo-sharing habits. In addition to manual photo-tagging as offered by common photo-sharing websites such as Flickr and Shutterfly, Polar Rose (`www.polarrose.com`) uses facial recognition algorithms for tagging unknown images of a subject if there is a tagged image of the subject on Polar Rose's image database (see [4] regarding the inadequacy of current privacy laws in this regard). Search engines, e.g., Spock (`www.spock.com`), customized for finding personal profiles posted at different websites, may provide even easier access to personal web content. Since September 2007, Facebook is allowing non-members to search for user profiles that are not access-restricted; third-party search engines such as Google and Yahoo! are also authorized to index such profiles (as of Feb. 2008).

Convenience and usability of IMPECS. IM contact lists are already in place for IM users, whereas social networking sites require users to invite friends and family members through, e.g., email to join a user's "friends' list"; sometimes these standardized, impersonal invitation emails simply irritate the recipients. IM is more interactive than social networking sites despite the immense recent popularity of those sites. For many IM users, IM clients start automatically after users log into their PC, and many IM users remain signed-on to an IM network as long as they use their computer. Social networking sites require a user to open a web browser, load a site, and sign into that site for maintenance or to view a friend's profile. IM users can view and control more effectively what content is being shared at any given time; information regarding who viewed what, and how frequently, may also be gathered from the IM server's ticket-issuing statistics.

We believe the following factors make IMPECS appealing. The viewing user B's role in IMPECS is simplified in comparison to the current social networking practice. B need only log into his IM client, and select an intended contact's URL for viewing. In contrast to social networking sites, B can remain unaware of who hosts his contact's web content. B need not even store or memorize A's URL; in fact, a bookmarked URL may not work depending on A's restrictions. However, B must realize that private URLs as shared through IMPECS are different than regular static URLs. The publishing user A's content sharing key K_{Aw} must be shared between S_i and S_w. This can be accomplished by any of the following means (in increasing order of convenience): (i) A manually copies the registration URL (containing K_{Aw}) from S_w to S_i using an interface provided by her IM client; (ii) S_w forms an XMPP URI (`xmpp:` [42]) embedding the key with URL_A, and A activates the URI (e.g. by a mouse click) to be processed by a locally installed XMPP client;[10] the client sends URL_A and K_{Aw} to S_i; or, (iii) S_w forwards K_{Aw} to S_i if there exists a pre-established relationship between the servers. A content key update is also similar to updating a URL link at S_i. To revoke B's viewing permission, A can simply place B on a separate IM contact group which does not have access to URL_A (or remove B from her contact list). Thus it is natural to expect that IMPECS is more convenient than current content sharing/limiting techniques on the web (e.g. password protection, obscure links). However, we hesitate to make any stronger usability claims without formal user testing (cf. [14]).

Once published on the Internet, private content may become permanent, e.g., through archived search engine queries and web crawlers [27]; in essence, the Internet does not forget anything published on it, although much of the personal information on the web (e.g. blogs, emotional responses, criticisms of friends and authorities) is meant to be transient. Unfortunately, momentary emotional responses to an event, if posted as text or image on the publicly accessible Internet, may bring unpleasant consequences at a later time. Our approach can enhance "forgetfulness" of the web by not making personal content public in the first place (cf. [11]). Web pages meant for certain personal contacts, friends and family will remain among the pre-established circle of trust as long as none of the trusted IM contacts make copies of a web page and republish it on the public Internet.

6. CONCLUDING REMARKS

Privacy is typically violated as a consequence of any of a number of factors. These seem to include: (i) oppressive administrations or large corporations (sometimes by exploiting the common misconception of "I've got nothing to hide" [46]); (ii) a shortage of usable tools to guard online

[10]Most popular IM protocols provide custom URI handlers, e.g., `ymsgr:` (Yahoo! Messenger), `aim:` (AOL Instant Messenger).

privacy; (iii) apathy towards privacy; and (iv) a misunderstanding of the implications of lost privacy. In our opinion, easy access to usable privacy tools may change the actions of ordinary web users towards online privacy; IMPECS is designed to be such a tool to enhance privacy of personal web content (i.e. we focus on addressing factor (ii) as listed above). We leverage the existing circles of trust among IM contacts, as well as encourage further refinements of trust in popular IM networks. Unlike current social networking websites, users do not need to (re-)build a "friends' list" in parallel to IM contact lists. In addition, users can publish their content at any website of their choice, and still be able to maintain privacy of their content (without being limited to use only a particular social networking site). Note that the general idea behind IMPECS extends beyond IM and IM circles of trust; any equivalent scheme, (ideally) containing pre-arranged groups, could similarly be leveraged (cf. Liberty Alliance People Service [24]).

As reported in a user survey [29], even most adult users of social networking websites keep their personal profiles open for all. We believe that such behaviour results largely from practical issues such as difficulties in ensuring close contacts join the same social networking site as the publishing user (just to view a friend's profile), or simply ignorance of the privacy implications of posting personal details on the Internet. IM is a very popular Internet application with a greater user base than social networking sites. Distributed IM services such as XMPP and Windows Live/Yahoo! networks enable IM communication between users of different IM networks. Therefore, we believe that IMPECS has significant deployment advantages over other personal content sharing techniques (e.g. password protection). By restricting personal content to a closed group of IM contacts, we believe IMPECS reduces opportunities for launching context-aware, targeted phishing attacks [30, 44, 50] where fraudsters collect social context of a target victim from their seemingly innocuous unprotected personal data, and enhances "forgetfulness" [27] of transient personal content on the web.

Acknowledgements

We thank anonymous reviewers for their comments and members of Carleton's Digital Security Group for enthusiastic discussion on this topic. The first author is supported in part by an NSERC CGS. The second author is Canada Research Chair in Network and Software Security, and is supported in part by an NSERC Discovery Grant, and the Canada Research Chairs Program.

7. REFERENCES

[1] ABC News. MySpace finds 29,000 sex offenders. News article (July 25, 2007). `http://www.abcnews.go.com/Technology/wireStory?id=3409947`.

[2] B. Adida. Beamauth: Two-factor web authentication with a bookmark. In *ACM Computer and Communications Security (CCS)*, 2007.

[3] S. Ahern, D. Eckles, N. Good, S. King, M. Naaman, and R. Nair. Over-exposed? Privacy patterns and considerations in online and mobile photo sharing. In *ACM Computer/Human Interaction (CHI)*, 2007.

[4] Anonymous. In the face of danger: Facial recognition and the limits of privacy law. *Harvard Law Review*, 120(7), May 2007.

[5] Anti-Phishing Working Group. Phishing activity trends report for April 2007. `http://www.antiphishing.org/reports/apwg_report_april_2007.pdf`.

[6] ArsTechnica.com. Yahoo Messenger and Windows Live Messenger get together. News Article (Sep. 27, 2006). `http://arstechnica.com/news.ars/post/20060927-7846.html`.

[7] S. B. Barnes. A privacy paradox: Social networking in the United States. *First Monday: Peer-reviewed Journal on the Internet*, 11(9), 2006.

[8] R. J. Bayardo and S. Thomschke. Exploiting the web for point-in-time file sharing (poster). In *World Wide Web (WWW) Conference*, 2005.

[9] R. J. Bayardo Jr., R. Agrawal, D. Gruhl, and A. Somani. YouServ: A web hosting and content sharing tool for the masses. In *World Wide Web (WWW) Conference*, 2002.

[10] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *AsiaCrypt*, 2000.

[11] N. Borisov, I. Goldberg, and E. Brewer. Off-the-record communication, or, why not to use PGP. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2004.

[12] BusinessWeek. Social-networking sites a 'hotbed' for spyware. News article (Aug. 18, 2006). `http://www.msnbc.msn.com/default.aspx/id/14413906/`.

[13] CBC.ca. 4 charged after school protest over Facebook suspensions. News article (Mar. 23, 2007). `http://www.cbc.ca/canada/toronto/story/2007/03/23/protest-birchmount.html`.

[14] S. Chiasson, P. van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security*, 2006.

[15] F. Dawson and T. Howes. vCard MIME directory profile, 1998. RFC 2426, Status: Standards Track.

[16] C. Dwyer, S. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Americas Conference on Information Systems (AMCIS)*, Keystone, Colorado, USA, Aug. 2007.

[17] R. Feizy. An evaluation of identity on online social networking: MySpace (poster). In *ACM Hypertext and Hypermedia (HT)*, 2007.

[18] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP authentication: Basic and digest access authentication, June 1999. RFC 2617, Status: Standards Track.

[19] M. Geist. Facing up to Facebook fears. BBC news article (May 9, 2007). `http://news.bbc.co.uk/2/hi/technology/6639417.stm`.

[20] V. D. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. In *Workshop on Fast Software Encryption*, Yokohama, Japan, Apr. 2001.

[21] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *ACM Workshop on Privacy in the Electronic Society (WPES)*, 2005.

[22] jabberd project. jabberd2 XMPP server. Version 2.1.6. `http://jabberd.jabberstudio.org/2/`.

[23] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10), Oct. 2007.

[24] Liberty Alliance. Liberty ID-WSF People Service – federated social identity. White paper (Dec. 5, 2005). http://www.projectliberty.org.

[25] M. Mannan and P. C. van Oorschot. Secure public instant messaging: A survey. In *Privacy, Security and Trust (PST)*, Fredericton, NB, Canada, Oct. 2004.

[26] M. Mannan and P. C. van Oorschot. A protocol for secure public instant messaging. In *Financial Cryptography and Data Security (FC)*, Anguilla, British West Indies, 2006.

[27] V. Mayer-Schönberger. Useful void: The art of forgetting in the age of ubiquitous computing. Harvard KSG Faculty Research Working Paper Series, article number RWP07-022, Apr. 2007.

[28] NACE Spotlight Online. The issues surrounding college recruiting and social networking web sites. News article (June 22, 2006). http://career.studentaffairs.duke.edu/undergrad/find_job/consider/nace_socialnetworks.html.

[29] National Cyber Security Alliance. CA/NCSA social networking cyber security survey. Online article (Sep. 2006). http://staysafeonline.org/features/SocialNetworkingReport.ppt.

[30] Netcraft.com. MySpace accounts compromised by phishers. News article (Oct. 27, 2006). http://news.netcraft.com/archives/2006/10/27/myspace_accounts_compromised_by_phishers.html.

[31] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications*, 32(9), Sept. 1994.

[32] New York Times. For some, online persona undermines a résumé. News article (June 11, 2006). http://www.nytimes.com/2006/06/11/us/11recruit.html.

[33] New York Times. How to lose your job on your own time. News article (Dec. 30, 2007). http://www.nytimes.com/2007/12/30/business/30digi.html.

[34] Pidgin project. Pidgin: A multi-protocol IM client. Version 2.0.1. http://www.pidgin.im/.

[35] PrisonPlanet.com. The Facebook.com: Big brother with a smile. News article (June 9, 2005). http://www.prisonplanet.com/articles/june2005/090605thefacebook.htm.

[36] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser: Analysis of web-based malware. In *USENIX HotBots*, 2007.

[37] D. Rand. Threats when using online social networks. CSIS Security Group (a Danish IT security company; article published on May 16, 2007). http://www.csis.dk/dk/forside/LinkedIn.pdf.

[38] Reuters UK. Networking sites a goldmine for ID fraudsters. News article (July 19, 2007). http://uk.reuters.com/article/personalFinanceNews/idUKHIL95513120070719.

[39] D. Rosenblum. What anyone can know: The privacy risks of social networking sites. *IEEE Security and Privacy*, 5(3), May 2007.

[40] P. Saint-Andre. Extensible messaging and presence protocol (XMPP): Core, Oct. 2004. RFC 3920, Status: Standards Track.

[41] P. Saint-Andre. Extensible messaging and presence protocol (XMPP): Instant messaging and presence, 2004. RFC 3921, Status: Standards Track.

[42] P. Saint-Andre. Internationalized resource identifiers (IRIs) and uniform resource identifiers (URIs) for the extensible messaging and presence protocol (XMPP), July 2006. RFC 4622, Status: Standards Track.

[43] SANS Internet Storm Center. MySpace phish and drive-by attack vector propagating Fast Flux network growth. SANS handler's diary (June 26, 2007). http://isc.sans.org/diary.html?storyid=3060.

[44] SecurityFocus.com. Image attack on MySpace boosts phishing exposure. News article (June 11, 2007). http://www.securityfocus.com/brief/522.

[45] SecurityFocus.com. QuickTime worm uses MySpace to spread. News article (Apr. 12, 2006). http://www.securityfocus.com/brief/375.

[46] D. J. Solove. 'I've got nothing to hide' and other misunderstandings of privacy. *San Diego Law Review*, 44, 2007.

[47] StopBadware.org. StopBadware.org identifies companies hosting large numbers of websites that can infect internet users with badware. Press release (May 3, 2007). http://www.stopbadware.org/home/pr_050307.

[48] K. Strater and H. Richter. Examining privacy and disclosure in a social networking community (poster). In *Symposium on Usable Privacy and Security (SOUPS)*, Pittsburgh, PA, USA, July 2007.

[49] Toronto Star. Social networking sites hacker targets. News article (Aug. 3, 2007). http://www.thestar.com/sciencetech/Technology/article/243096.

[50] Wired.com. Fraudsters target Facebook with phishing scam. News article (Jan. 3, 2008). http://www.wired.com/politics/security/news/2008/01/facebook_phish.

[51] Wired.com. Private Facebook pages are not so private. News article (June 28, 2007). http://www.wired.com/software/webservices/news/2007/06/facebookprivacysearch.