

Secure Public Instant Messaging: A Survey[†]

Mohammad Mannan
 School of Computer Science
 Carleton University
 Ottawa, Ontario, K1S 5B6
 Email: mmannan@scs.carleton.ca

P.C. van Oorschot
 School of Computer Science
 Carleton University
 Ottawa, Ontario, K1S 5B6
 Email: paulv@scs.carleton.ca

Abstract

We provide a survey on security features and threats to existing Instant Messaging (IM) networks and discuss how currently available systems fail to provide adequate security in light of existing threats. Our discussion and analysis provide a starting point from which to advance academic research in the area of secure IM systems, enabling security improvement in the longer term.

I. INTRODUCTION AND OVERVIEW

Instant Messaging (IM) is a type of communications service over the Internet that enables individuals to exchange text messages and track availability of a list of users in real-time. IM systems have been around since the UNIX applications `talk` and `write`. IM usage increased with the early implementations of the MIT Project Athena Zephyr notification system [1], and IRC (started at University of Oulu in Finland; RFC1459 [2], RFC2810 [3]). However, the increasing popularity of consumer IM services in the recent past is phenomenal. Giga Information Group, a market research firm, estimates that about 325 million people worldwide will use consumer IM applications in the year 2003 [4]. Starting as a casual application, mainly used by teenagers and college students, IM systems now connect even naval operations (over 300 US Navy warships are connected via IM service) to various customer services [5].

There are many free public domain instant messaging services. The most popular are AOL Instant Messenger (AIM), ICQ, MSN Messenger (Windows Messenger in XP), and Yahoo! Instant Messenger (YIM). This paper focuses on these messaging networks and their default clients.¹ There are also many other third-party clients that interact on these networks. We discuss both the third party and default clients in terms of security risks associated with them. The basic protocols currently used in the public IM systems are open to many security threats (see §V). Security features (e.g. using SSL connections or digital certificates) in corporate IM systems are inadequate to address these threats. To our knowledge, there exists no security protocol suite in the literature specifically tailored for password-based IM systems.

The current Internet Threat Model [6, p.1] (including the SSL model) assumes a totally vulnerable communication link with trusted end-points. However, the assumption of “secure” end-points may undermine software security, as the present Internet environment is infested with malicious software compromising a large number of machines at any given point of time [7]. This is one of the reasons why it seems reasonable to us to conclude that an SSL based solution is not adequate for IM security. IM security is discussed in this paper with respect to an *extended* threat model which takes (not necessarily trusted) end-points into consideration.

Some of the security threats to instant messaging are similar to those for email, for example, misleading Web links (often used to “phish” [8] for passwords) and malcode execution from files. Anti-virus tools to protect email from such threats are quite mature and relatively effective. For email, such tools can be implemented at the gateway level, as monitoring email traffic is straightforward. For instant messaging, the use of unpublished, non-standard proprietary protocols and non-centralized peer-to-peer file transfer makes it difficult to monitor IM traffic. Hence, incorporating similar protection mechanisms as used for email appears to be more difficult, and to provide at best a limited shield against IM threats.

Motivation. IM differs from many other Internet applications because of its real-time nature of user interaction. Consequently many security mechanisms designed for other Internet applications (e.g. Web browser, email) are inadequate for IM. We highlight threats to IM to create greater public awareness of the danger of using present IM systems, and to improve security in the long term (although in the short term this may increase the risk of these threats becoming reality). We seek to lay the foundation to advance research in the area of secure IM, as a first step towards improving security in IM systems.

Scope. Some vendors provide IM for mobile devices. The Short Messaging System (SMS) was created as part of the GSM (Global System for Mobile Communications) Phase 1 standard. These systems, and Internet Relay Chat (IRC) and chat rooms (see also §III) are beyond the scope of this paper. Our main focus is (one-to-one) PC-to-PC messaging, which is the dominating feature of all instant messengers. Instant messaging systems that mainly target corporate users, such as Yahoo!

[†]Version: September 17, 2004. To appear in the Proceedings of the Second Annual Conference on Privacy, Security and Trust, Fredericton, NB, October 13–15, 2004.

¹By *default clients* we mean the IM clients provided by IM service providers (e.g. MSN Messenger). By *third-party* clients we refer to clients (e.g. Gaim, Trillian, IMSecure) which interact with the existing major IM networks, and security-enhanced IM products (e.g. Yahoo! Business Messenger).

Business Messenger,² are not fully analyzed in this paper (mainly because complete documentation of security features in these products is not publicly available). As the default IM clients discussed here are mainly Windows based, Windows is generally implied to be the underlying operating system when another is not explicitly mentioned.

Outline. The sequel is organized as follows. §II lists related work regarding IM. §III summarizes the basic protocols used for the mainstream IM systems. Privacy and security features of current IM services and third-party solutions are briefly described in §IV, along with weaknesses related to third-party solutions. §V discusses the most significant security threats to instant messaging. §VI discusses future work and conclusions.

II. RELATED WORK

Much work (albeit most unrelated to security) has been done on instant messaging and presence awareness systems in academia, mainly by Human Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW) research groups. Several IM applications – e.g. Hubbub [9] (a sound-enhanced IM), KIM (Kinetic Typography-Based Instant Messaging) [10], IMVis [11] (which uses pictures and video snapshots to visualize contacts), and Threaded Chat [12] – have been designed to augment functionalities and to analyze usage. The Unified Messaging System [13] emerged from the pervasive computing idea that combines email, IM, newsgroups, SMS, paging etc. into one system. Many researchers have explored the effects of instant messaging in the workplace. A study by Issacs *et al.* [14] found that 62% of IM conversations in the workplace were work related. Handel *et al.* [15] reported similar results (69% of recorded instant messages were work related). These results suggest positive contributions of IM in the workplace, although other researchers [16], [17] have expressed concerns of IM being used as a tool for “gossiping” or “goofing off”.

Related to IM security, a modified Diffie-Hellman protocol suitable to instant messaging has been designed by Kikuchi *et al.* [18], primarily intended to secure message confidentiality against IM servers. It does not ensure authentication and also has problems similar to the IMSecure³ solutions as discussed in §IV. Hindocha [19] discusses popular IM protocols, worms, threats and firewall issues in a 2003 white paper. A Web resource on security analysis of Cerulean Studios’ Trillian application is also available [20]. Informal discussions of security problems related to public instant messaging in the enterprise environment are available (e.g. see Frase [21] re: some solutions using well-defined security policies and anti-virus tools).

IM protocol standardization efforts are ongoing in the Internet Engineering Task Force (IETF) community in three main working groups: Instant Messaging and Presence Protocol (IMPP),⁴ SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE),⁵ and Extensible Messaging and Presence Protocol (XMPP, based on Jabber Instant Messaging and Presence).⁶ Several Internet-Drafts and Request For Comments (RFC) have been produced by these groups. RFC 2779 [22, §5] lists the security considerations for IMPP. A good comparative study on IM protocols including SIMPLE and XMPP has recently been done by Debbabi *et al.* [23]. Also, a Jabber Inc. whitepaper [24] compares the SIMPLE and XMPP protocols.

XMPP [25] includes a method to protect an XML stream⁷ from tampering and eavesdropping. XMPP uses the Transport Layer Security (TLS) protocol for stream encryption, along with a “STARTTLS” extension modeled after similar extensions for the IMAP, POP3, and ACAP protocols as described in RFC 2595 [26]. The Simple Authentication and Security Layer (SASL, RFC 2222 [27]) is proposed as a method for adding pluggable authentication support in XMPP.

Security protocols and mechanisms for SIP (RFC 3261 [28]) are quite standardized. However, no specific security protocols have been developed focusing on SIMPLE (RFC 3428 [29]). Mechanisms for authentication, end-to-end protection, replay and denial of service attack prevention for SIMPLE heavily rely on TLS and S/MIME protocols. Details of these security mechanisms are described in RFC 3428 [29], RFC 3261 [28] and RFC 3265 [30].

Our work is a survey of existing literature and working products regarding IM. However, instead of HCI or feature related issues, we focus on security risks of IM.

III. BASICS OF IM PROTOCOLS AND FEATURES

To facilitate discussion, we first provide a few definitions, mainly related to instant messaging:

²<http://messenger.yahoo.com/messenger/business/>

³<http://www.imsecure.com>

⁴<http://www.ietf.org/proceedings/03mar/111.htm>

⁵<http://www.ietf.org/html.charters/simple-charter.html>

⁶<http://www.ietf.org/html.charters/xmpp-charter.html>

⁷i.e., a container for the exchange of XML elements between any two entities over a network.

<i>online user</i>	A user successfully logged in to an IM server.
<i>presence</i>	Presence information reveals whether a user is logged in to an IM server or not.
<i>availability/user mode</i>	Availability information reveals a user's willingness (e.g. "busy", "do not disturb") to send/receive messages, or status (e.g. "away", "on the phone").
<i>contact/buddy list</i>	The list of user IDs whose presence and availability a user has currently subscribed to.
<i>block list</i>	The list of user IDs explicitly barred from getting the current user's presence and availability information; listed users cannot send any messages to the current user.
<i>allow list</i>	The list of user IDs allowed to send messages to the current user and which can track the user's presence and availability information.
<i>one-to-one chat</i>	When a user sends or receives messages from another user, generally through the IM server.
<i>group chat</i>	When more than two users are exchanging messages. Users form a virtual "group", generally which is short-lived. Users in a group chat are usually closely related.
<i>chat room</i>	A virtual room, generally consisting of many users who exchange instant messages on some closely related topics.
<i>IRC</i> ⁸	A client-server chat system of large (often worldwide) networks. IRC is structured as networks of Internet servers, each accepting connections from client programs, one per user.

Descriptions of most of the protocols for the major IM networks are available on the Internet. Software makers have released the protocols or the protocols have been reverse-engineered. The remainder of this section contains a brief architectural overview for popular IM networks. Details of protocols for AOL [31], Yahoo! [32] and MSN [33] Instant Messengers are publicly available.

Common features supported in most IM clients include: contact lists; block lists; presence information; availability (available, away, busy etc.); sending and receiving instant messages to online/offline users (one-to-one, multi-user); email; sending and receiving files, URLs; audio and video chat; sharing external applications (e.g. Internet browser); launching online games; setting permission levels for different types of users (e.g. contact list, everyone); and message archiving.

Most communications in IM systems are client-server based, where each user shares a secret, user-chosen (often "weak") password with the IM server. A password hash is generally exchanged between a client and a server for authentication. Messages among users are also typically relayed through the server (mainly to avoid firewall issues). However, purely peer-to-peer communications also occur in some situations (e.g. audio/video chat, file transfer). Communications occur mostly over TCP; however, UDP is sometimes used in peer-to-peer connections. Also, SSL is used in some corporate IM services (e.g. Reuters Messaging⁹) and in the authentication phase of the currently available MSN protocol. While the IM server appears to be a single entity to a client, it may be a group of servers controlled by a single IM service provider, or a collection of servers from independent IM service providers. If user *A* wants to communicate instantly with user *B*, both must log into the same IM service. Messages from *A* to *B* will be delivered by the server depending on *B*'s privacy settings. For direct communications between *A* and *B*, the server provides necessary information (e.g. network address) to each party. Figure 1 shows the standard IM communications model for single and multiple servers.

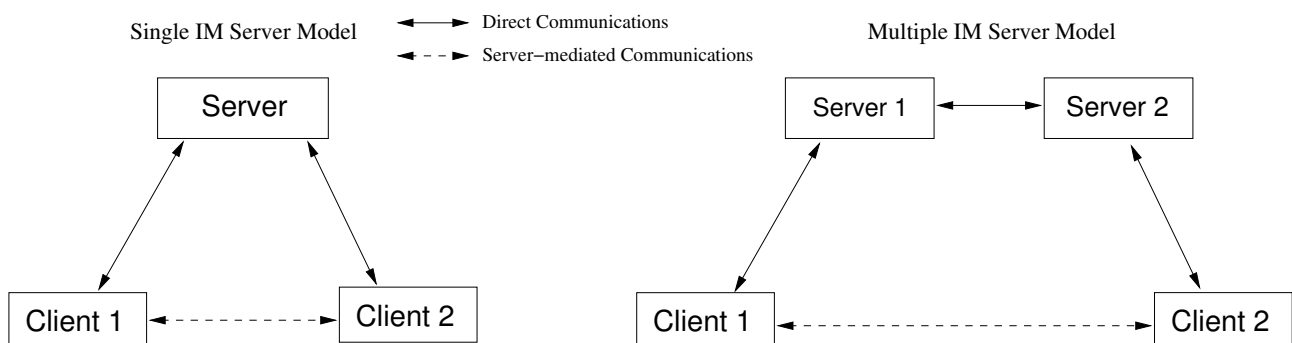


Fig. 1. IM Communications Model

IV. EXISTING SECURITY MECHANISMS

In this section, we list the available security and privacy mechanisms in popular default IM clients, and summarize the security features of popular third-party IM solutions. Problems of third-party and corporate IM solutions in the public domain are also briefly discussed.

⁸Definition from HyperDictionary, <http://www.hyperdictionary.com/>

⁹<http://about.reuters.com/productinfo/messaging/>

Security and Privacy Features in Default Clients. The latest versions of all major IM clients include an option to employ anti-virus software which can be launched automatically after every IM file download. An authorization option can be turned on so that explicit consent (non-cryptographically protected) of user *A* is required before *A* can be added to another user *B*'s contact list. The same option is there to select who (users from a contact list or everyone) can see user *A*'s online status, and who can send messages and files to *A*. However, "add-contact request" and "response" messages are transferred without any cryptographic protection, making them easy to spoof. In ICQ,¹⁰ a user can choose an option to select specific contacts who can see his/her online status even when the user is in *invisible* (logged in, but appearing offline to others) mode. ICQ and YIM clients are equipped with word filtering mechanisms which can be used to replace or remove unwanted words from incoming text messages.

An ICQ client may ask the user to enter the login password every time he/she wants to modify user-details, security and privacy permissions, and preferences settings. Users need to turn on this feature explicitly after installation. ICQ has an option to accept/decline URLs from everyone or only from those in a user's contact list.

IM clients are generally notified when a new version of client software is available with new features or new fixes, but users may choose not to upgrade, and most of the time, software vendors allow older versions for backward compatibility. However, since October 2003, MSN Messenger prevents login unless a user updates to the MSN protocol version preferred by Microsoft when there was a major change in its authentication mechanism.¹¹ Another useful feature that all the major IM vendors now provide is the protection against automated account creation [34]. This prevents software bots from signing up for an unlimited number of accounts that can be used for sending unwanted messages to legitimate IM users.

Third-party Solutions. Several IM products that claim to be "secure" are discussed below. There is a lack of documentation about what is protected and what is not in these products. Our discussion is mainly based on available Web resources, users' guides and help files.

Norton Anti Virus 2003 comes with an IM plug-in for automatic scanning of incoming files. Also in common anti-virus software, there is an option to check all executables before launching. However, anti-virus protections for IM currently check for malware only in the file transfers. They cannot provide message confidentiality or integrity, or protect against URL exploitations (see item 8 in §V).

Currently, AIM clients can use a personal digital certificate to enhance authentication, integrity and confidentiality of text messaging.¹² To enable this, both users buy Class 2 digital certificates from VeriSign. Although AIM's solution using personal digital certificates is a powerful mechanism, it is expensive for public domain IM users and puts the burden of certificate distribution, verification, expiry, renewal, revocation etc. on end users.

The ZoneAlarm¹³ personal firewall from Zonelabs has a feature called *ID Lock*. It is limited to protect user-configurable sensitive information like bank and credit card numbers, home address, SIN (Social Insurance Number) etc. from being divulged in IM or email texts. Zonelabs' IM security solutions IMSecure and IMSecure Pro provide seamless encryption for popular IM clients (AOL Instant Messenger, MSN Messenger and Yahoo! Messenger). IMSecure and IMSecure Pro have similar security properties. IMSecure purportedly works in the following manner (excerpt from the Readme file of IMSecure installation):

IMsecure Pro relies on the OpenSSL library for cryptographic services. The text of each message in a secure session is encrypted with the DES 56-bit cipher in the CBC mode.

IMsecure Pro automatically and transparently creates a self-signed X.509v3 digital certificate for each of the user's IM accounts upon the first login. At the beginning of the first IM conversation between two IMsecure Pro users after installing IMsecure Pro, the certificates are transparently exchanged between the users and stored on their computers. The public key from one of the certificates is used to encrypt the session key to be used for the duration of the session. Upon receiving the encrypted session key, the other user's IMsecure Pro decrypts it with that user's private key and completes the secure session initialization. IMsecure Pro never reuses session keys.

Trillian¹⁴ provides text message encryption for AIM accounts when both peers use Trillian's software to communicate. Murphy [20] provides an analysis of this approach. Security services (integrity and confidentiality, but not authentication) provided by IMSecure and Trillian's solutions are useful to some degree. In these systems, instant messages are confidential between two users — in the sense that decrypting messages intercepted during transmission is computationally infeasible. However, such systems provide no protection against malware implanted in users' systems, and communications from a client to a server are not encrypted. An attack scenario is depicted in Figure 2. Another disadvantage of IMSecure is that it needs to be installed in each system that a user wants to use for secure messaging (and also by the intended recipients). Furthermore, solutions such as IMSecure that use locally stored information (e.g. private keys) restrict users' mobility.

¹⁰<http://web.icq.com/help/faq/1,,1709,00.html>

¹¹<http://paulotaylor.com/palmsn/>

¹²<http://www.verisign.com/support/class1/secureaol.html>

¹³<http://www.zonealarm.com>

¹⁴<http://www.trillian.cc>

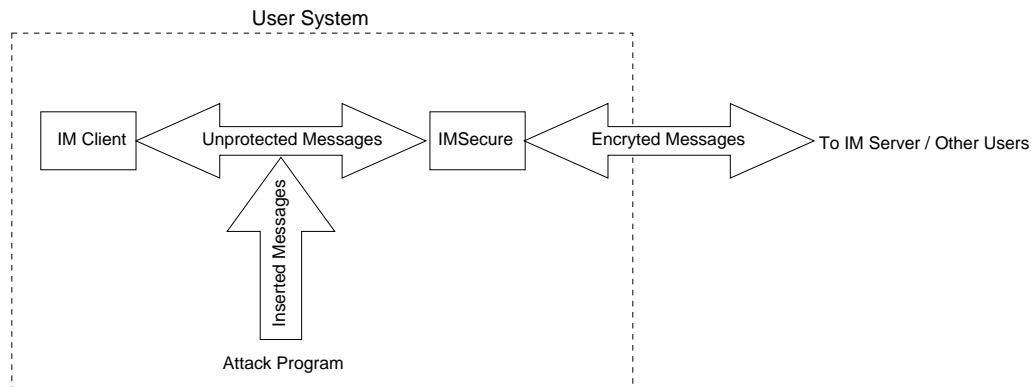


Fig. 2. Weakness of IMSecure model

Yahoo! Business Messenger¹⁵ and Reuters Messaging protect instant messages using 128-bit SSL encryption. These products mainly target corporate users. Using SSL-based solutions for public IM service has three major drawbacks: (1) limited threat model (see §I); (2) overhead for deployment at server-side (protocol is resource hungry and slow); and (3) messages may not be “private” when they go through a server, i.e. the server may view any encrypted message [18]. However, this last characteristic is desirable for message logging, albeit not when users value privacy to the extent that they prefer not to disclose their conversations to service providers.

iGo Incognito¹⁶ asserts to be an IM facility built from the ground up on cryptographically strong security. During login, a message encrypted with the server’s public key is sent from a client to the server that contains a user’s identification parameters. Upon verification, a random 128-bit session key is established between a client and the server. To send a text message to a user, the sender generates a random one-time key. The sender encrypts the message with this key using AES [35]. The encrypted message is forwarded to the recipient along with the key encrypted under the recipient’s public key. As the messages are relayed through the server, they are also encrypted using the shared key with the server, causing messages to be doubly encrypted. A client’s private key is stored encrypted on the local hard drive under a user-chosen pass-phrase. Also, all user messages are digitally signed, corroborating who the sender is. We have not attempted to independently confirm or refute these security properties.

V. SECURITY THREATS TO IM

The evolution of IM systems suggests that security and privacy issues have received little consideration from the major IM vendors. This section lists the most significant threats to IM systems. This list is constructed from known attack forms, and system design and implementation flaws that may allow future attacks. Detailed descriptions of IM exploits to date can be found in many Web resources (e.g. [19], Instant Messaging Planet,¹⁷ Symantec Security Response¹⁸). One objective of compiling this list is to acquire insight to aid in designing a robust security protocol suite for IM systems.

1. Insecure Connections. Perhaps the greatest threat to current popular instant messaging networks lies in their open, insecure connections. Most services use a client-server model for communication among users with a few exceptions like file transfer, voice and video services where a peer-to-peer connection is used. Connections are susceptible to being taken over during client-to-server, server-to-client, client-to-client and intra-server (in the same messaging network) communications. Once authenticated during the login time, all these connections deploy little (sequence number or transaction identifier, which can be easily spoofed) or no security measures at all. Hence almost all popular IM connections lack authentication (except in the login message), confidentiality and integrity. This opens the door to many other security vulnerabilities including impersonation, denial of service, man-in-the-middle attacks, replay, etc. Even if a user chooses not to receive any IM outside his/her contact list, it is possible to assault the user with unwanted messages. All the attacker needs to accomplish this is to capture an open connection with one of that user’s contacts.

2. Denial of Service (DoS). DoS attacks can be launched in many different ways. Some may simply crash the messaging client repeatedly. Attackers may use the client to process CPU and/or memory intensive work that will lead to an unresponsive or crashed system.

Flooding with unwanted messages is particularly easy when users choose to receive messages from everyone. In this case, attackers may also send spam messages such as advertisements. However, all the common IM clients support user blocking. A

¹⁵<http://messenger.yahoo.com/messenger/business/products/msg/faq.php>

¹⁶<http://www.igo-incognito.com/>

¹⁷<http://www.instantmessagingplanet.com/security/>

¹⁸<http://securityresponse.symantec.com/avcenter/>

victim can block the attacker's account ID easily; however, attackers may get through this barrier by using many compromised accounts simultaneously.

Attackers may also change the password of compromised accounts using automated scripts. This will cause the victims to lose access to their accounts whose account names they have distributed to many contacts.

3. Impersonation. Attackers may impersonate valid users in at least two different ways. If a user's password is captured, attackers can use automated scripts to impersonate the victim to users in his/her contact list. Alternatively, attackers can seize client-to-server connections (e.g. by spoofing sequence numbers). A connection may be taken over right after a user logs in, when a user initiates a connection with a peer or when a user gets disconnected unexpectedly (e.g. by DoS attacks). The server will keep the connection(s) open for some time until the keep-alive rate¹⁹ is violated. Attackers can take advantage of this time out to capture the connection to the server.

As none of the popular instant messaging service protects their connections with encryption, it is quite easy to impersonate any connection via man-in-the-middle attacks [19].

4. IM as a Worm Propagation Vector. Here we use a broad definition of worms by Kienzle *et al.* [36]: "A worm is malicious code (standalone or file-infecting) that propagates over a network, with or without human assistance". Worms can easily propagate through instant messaging networks using the file transfer feature. Generally, users are unsuspecting when receiving a file from a known contact. Worms successfully use this behavior by impersonating the sender. This is becoming a serious problem, as common anti-virus tools do not generally monitor IM traffic. Also, IM file transfers carrying malware penetrate firewalls more easily than email attachments. This is due to the difficulty in distinguishing IM traffic at gateway levels and vendors' use of proprietary protocols [19]. Like email address books, IM worms can use a user's online contact list as a propagation vector; however, unlike "offline" and slow email propagation, IM contacts provide "instant" victims for fast spreading. According to Keizer [37], a simulation by Symantec found that half a million systems could be infected in 30 to 40 seconds through an IM worm.

YIM has an option to open a file automatically after downloading. This can help spread malcode with less user intervention. In ICQ, users can choose to automatically accept all incoming file transfer requests.

With the latest IM clients, users can set up automatic virus scanning for incoming files. However, anti-virus tools generally scan only a small subset of all possible file types. For example, a media file (e.g. an MPEG file) may contain a specially crafted data sequence that may crash a user's media player or do something more harmful. In fact, for Real Media [38] and JPEG [39] files, these threats are already reality. As most anti-virus tools are not generally used to scan data files (e.g. media or image files), widespread use of software such as Windows Media Player may become a potential source of attacks that use malcode in data files.

5. Plaintext Registry and Message Archiving. There are many security related settings in IM clients. Knowledgeable users can set privacy and security settings for their needs. IM clients save these settings in the Windows registry. Any technically inclined Windows user can read registry values and users with administrative power can modify those as well. Some security related IM settings saved in the registry are: encrypted password, user name, whether to scan incoming files for viruses and the anti-virus software path, whether permission is required to be added in someone's contact list, who may contact the user (only from contacts or everyone), whether to share files with others, shared directory path, and whether to ask for a password when changing security related settings. MSN Messenger even stores a user's contact list, block list and allow list in the registry²⁰ in a human-readable format. Attackers can use Trojan horses to modify or collect these settings with little effort. Modifying the registry may help the intruder bypass some security options like add contact authorization, file transfer permission etc. By collecting user names and password hashes, attackers can take control of user accounts. Also, the plaintext password can be extracted from the encrypted password stored in the registry using tools such as Elcomsoft's Advanced Instant Messengers Password Recovery.²¹

IM clients allow message archiving. User conversations are saved in a plaintext format in a predictable system location. Revelation of these messages can be potentially very significant loss of message confidentiality for both corporate and home users.

6. Insecure Default Settings. As is common in many software products, default privacy and security settings in IM clients are often not appropriate. Most IM clients allow anyone from the same IM service to contact (send text message, files etc. to) a user by default. Allowing message reception from all opens the door to a new vector of nuisance — *spim*, the unsolicited commercial messages sent via an IM system. This option may be restrictive to allow the people only from a user's contact list, because IM users do not communicate with strangers often [40]. Also, the default file download location in a user's machine may be misused e.g. as in the ICQ scm file vulnerability.²²

¹⁹i.e. messages that are being transferred in a certain interval to notify that a connection is active.

²⁰HKEY_CURRENT_USER\Software\Microsoft\MessengerService\ListCache*.NET Messenger Service

²¹<http://www.elcomsoft.com/aimpr.html>

²²<http://www.securityfocus.com/archive/1/282631>

In ICQ, the default setting for contact list authorization is “All users may add me to their Contact List and see my Online / Offline status”. Clearly this is not acceptable for many users. Permission for viewing the user’s shared directory is set to “Only users from my Contact List” by default in ICQ. However this does not provide meaningful security when “add contact” authorization is not required.

7. Sharing IM Features with Other Applications. The MSN Messenger contact list and other features are available from multiple other software applications, such as the Microsoft Outlook Express email client and Hotmail Web email service (when launched from an Internet Explorer browser). Microsoft has also published IM APIs for application developers for custom integration.²³ Yahoo! also provides developers with programmable objects like Yahoo! Audio Conferencing and Yahoo! Webcam Upload/Viewer. AIM Express²⁴ is implemented as an applet for the Java platform that runs in Web browsers to support better user mobility. As IM capabilities are being integrated with many different applications, security risks are increasing for both the IM services and the host applications as a security bug in an IM service can affect other applications that implement the IM features and vice versa. This significantly increases attack opportunities for malware writers.

8. Malicious Hyperlinks. Links to Web pages containing malicious content can be sent as normal instant messages. ICQ has an option to accept or reject messages with hyperlinks. In AIM, a user can create hyperlinks where the visible text can be completely unrelated to the underlying web link. This can easily dupe any user receiving a hyperlink having an innocent visible text with a deceitful link.

9. Exploitable URI (Universal Resource Identifier) Handlers (*aim*, *ymgr*). YIM and AIM clients install custom URL handlers *ymgr* and *aim* respectively. These URIs can help in writing useful scripts to be processed by applications such as Microsoft Internet Explorer, Netscape Navigator, Mozilla Firefox, Microsoft Outlook, or the Windows command shell. A URI can be sent by another YIM or AIM user in a message, embedded in a Web site, or sent in an HTML email message. Web browsers and command shells can be used to launch AIM or YIM to process these URIs.

The lack of bounds checking in parameters of these protocols has allowed malicious hackers to launch various buffer overflow attacks (e.g. [41]). By changing the registry value (e.g. for YIM, <program path>\YPAGER.EXE %1) to any rogue program, attackers can guarantee to launch that program when these protocols are invoked. Also, scripts written using these URIs open a new front for automated attack.

10. DNS Spoofing to Setup Rogue IM Server. Trojans like QHosts-1²⁵ can be used to modify the TCP/IP settings in a victim’s system to point to a different DNS server. Malicious hackers can set up an IM server and use DNS spoofing so that victims’ systems connect to the rogue server instead of a legitimate one. IM clients presently have no way to verify whether they are talking to legitimate servers. Servers verify a client’s identity by checking the user name and password hash. This server-side only authentication mechanism can be targeted for IM man-in-the-middle attacks where a rogue server may pose as a legitimate server (e.g. [42]). Account-related information collection, eavesdropping, impersonation and many other attacks are possible if this attack is successful.

VI. CONCLUDING REMARKS AND FUTURE WORK

Public IM systems are widely popular, and yet the vast majority of users are largely ignorant of the dangers associated with IM. Although more enterprise users are aware of IM threats, current public and enterprise IM systems fail to provide sufficient security. In this survey, a number of risks of using IM systems are discussed. Our goal is to initiate more substantial discussions in the academic community on how to alleviate the IM threats, so that secure IM protocols and necessary security mechanisms are designed to better mitigate presently known and probable future threats.

The idea of *mobile* instant messaging, as introduced by Issacs *et al.* [43] in a modified version of Hubbub, is establishing a foothold on major messaging systems. The current version of AIM supports login from multiple devices at the same time to enhance user mobility. An IM security protocol should ideally accommodate the multiple-login as well as group chat and chat room features.

Designing a secure instant messaging system requires serious consideration of human-computer interface issues. A restrictive model (i.e. one which imposes high security at the expense of usability) may deter IM users, most of whom use IM as a casual system without being aware of the underlying threats. Restrictive systems may have adverse affects; users may move to competing products which are less secure but more convenient or it may destroy the spontaneity of IM. Nonetheless, we strongly believe that security and privacy issues in IM should get more emphasis and additional measures should be put in place before IM becomes as big a security problem as email, which remains the number one breeding ground for computer worms, despite ubiquitous security measures [36]. A widely deployed vulnerable system like IM is destined to increasingly attract malware writers.

Acknowledgements. We thank all members of Carleton’s Digital Security Group for their enthusiastic discussions on this topic, especially Anil Somayaji, Julie Thorpe and Glenn Wurster. The second author is Canada Research Chair in Network and

²³http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winmessenger/winmessenger/messenger_entry.asp

²⁴An AIM client with minimal features, http://www.aim.com/get_aim/express/aim_expr.adp

²⁵<http://securityresponse.symantec.com/avcenter/venc/data/trojan.qhosts.html>

Software Security, and is supported in part by an NSERC Discovery Grant, the Canada Research Chairs Program, MITACS and Alcatel Canada.

REFERENCES

- [1] C. A. DellaFera, M. W. Eichin, R. S. French, D. C. Jedlinsky, J. T. Kohl, and W. E. Sommerfeld, "The Zephyr notification service," in *Proceedings of the USENIX Technical Conference*, Dallas, TX, Feb. 1988, pp. 213–220.
- [2] J. Oikarinen and D. Reed, "RFC 1459: Internet Relay Chat Protocol," May 1993, Status: Experimental. <http://www.faqs.org/rfcs/rfc1459.html> [Accessed: June 10, 2004].
- [3] C. Kalt, "RFC 2810: Internet Relay Chat: Architecture," Apr. 2000, Status: Informational. <http://www.faqs.org/rfcs/rfc2810.html> [Accessed: June 22, 2004].
- [4] G. Lawton, "Instant messaging puts on a business suit," *IEEE Computer Society: Computer Magazine*, Mar. 2003, <http://www.computer.org/computer/homepage/0303/Lawton/> [Accessed: Dec. 8, 2003].
- [5] S. M. Cherry, "IM means business," *IEEE Spectrum Online*, vol. 39, pp. 28–32, Nov. 2002, <http://www.spectrum.ieee.org/WEBONLY/publicfeature/nov02/im.html> [Accessed: Dec. 7, 2003].
- [6] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*. Addison Wesley, 2001, http://www.iang.org/ssl/rescorla_1.html [Accessed: June 28, 2004].
- [7] S. Savage, Ed., *Proceedings of the 2003 ACM Workshop on Rapid Malcode (WORM'03)*. Washington, D.C., USA: ACM Press, Oct. 2003.
- [8] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-side defense against Web-based identity theft," in *Network and Distributed System Security Symposium Conference Proceedings: 2004*, San Diego, CA, Feb. 2004.
- [9] E. Isaacs, A. Walendowski, and D. Ranganathan, "Hubbub: A sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, 2002, pp. 179–186.
- [10] K. Bodine and M. Pignol, "Kinetic typography-based instant messaging," in *CHI '03 extended abstracts on Human Factors in Computer Systems*. ACM Press, 2003, pp. 914–915.
- [11] C. Neustaedter, S. Greenberg, and S. Carpendale, "IMVis: Instant messenger visualization," in *Proceedings of the 2002 ACM on Computer Supported Cooperative Work Video Program*. New Orleans, Louisiana: ACM Press, 2002, pp. 6–6.
- [12] M. Smith, J. J. Cadiz, and B. Burkhalter, "Conversation trees and threaded chats," in *Computer Supported Cooperative Work*, 2000, pp. 97–105, <http://citeseer.nj.nec.com/smith00conversation.html> [Accessed: Dec. 7, 2003].
- [13] J.-M. Wams and M. van Steen, "Pervasive messaging," Vrije Universiteit, Amsterdam, <http://citeseer.nj.nec.com/560838.html> [Accessed: Dec. 7, 2003].
- [14] E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano, and C. Kamm, "The character, functions, and styles of instant messaging in the workplace," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. ACM Press, 2002, pp. 11–20.
- [15] M. Handel and J. D. Herbsleb, "What is chat doing in the workplace?" in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. ACM Press, 2002, pp. 1–10.
- [16] J. D. Herbsleb, D. L. Atkins, D. G. Boyer, M. Handel, and T. A. Finholt, "Introducing instant messaging and chat in the workplace," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, 2002, pp. 171–178.
- [17] S. Whittaker, D. Frohlich, and O. Daly-Jones, "Informal workplace communication: What is it like and how might we support it?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, 1994, pp. 131–137.
- [18] H. Kikuchi, M. Tada, and S. Nakanishi, "Secure instant messaging protocol preserving confidentiality against administrator," in *18th International Conference on Advanced Information Networking and Applications, AINA 2004*, vol. 2, Fukuoka, Japan, Mar. 2004, pp. 27–30.
- [19] N. Hindocha, "Threats to instant messaging," Symantec Security Response, 2003, <http://securityresponse.symantec.com/avcenter/reference/threats.to.instant.messaging.pdf> [Accessed: Dec. 7, 2003].
- [20] M. D. Murphy, "Instant message security - Analysis of Cerulean Studios' Trillian application," SANS Institute, June 2003, http://www.giac.org/practical/GSEC/Michael_Murphy_GSEC.pdf [Accessed: Dec. 7, 2003].
- [21] D. Frase, "The instant message menace: Security problems in the enterprise and some solutions," SANS Institute, Nov. 2001, <http://www.sans.org/rr/papers/60/479.pdf> [Accessed: Dec. 7, 2003].
- [22] M. Day, S. Aggarwal, G. Mohr, and J. Vincent, "RFC 2779: Instant messaging / presence protocol requirements," Feb. 2000, Status: Informational. <http://www.faqs.org/rfcs/rfc2779.html> [Accessed: May 29, 2004].
- [23] M. Debbabi and M. Rahman, "The war of presence and instant messaging: Right protocols and APIs," in *1st IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, Jan. 2004, pp. 341–346.
- [24] Jabber Inc., "Architectural considerations for presence and instant messaging infrastructure: Comparing XMPP and SIP/SIMPLE," May 2004, http://www.jabber.com/index.cgi?CONTENT_ID=55 [Accessed: June 9, 2004].
- [25] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core draft-ietf-xmpp-core-24," Jabber Software Foundation, May 2004, <http://www.ietf.org/internet-drafts/draft-ietf-xmpp-core-24.txt> [Accessed: May 29, 2004].
- [26] C. Newman, "RFC 2595 - Using TLS with IMAP, POP3 and ACAP," June 1999, Status: Proposed Standard. <http://www.faqs.org/rfcs/rfc2595.html> [Accessed: June 22, 2004].
- [27] ———, "RFC 2444: The one-time-password SASL mechanism," Oct. 1998, Updates RFC2222. Status: Proposed Standard. <http://www.faqs.org/rfcs/rfc2444.html> [Accessed: June 18, 2004].
- [28] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261 - SIP: Session Initiation Protocol," June 2002, Status: Proposed Standard. <http://www.faqs.org/rfcs/rfc3261.html> [Accessed: June 22, 2004].
- [29] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle, "RFC 3428: Session Initiation Protocol (SIP) extension for instant messaging," Dec. 2002, Status: Proposed Standard. <http://www.faqs.org/rfcs/rfc3428.html> [Accessed: June 22, 2004].
- [30] A. B. Roach, "RFC 3265 - Session Initiation Protocol (SIP)-specific event notification," June 2002, Status: Proposed Standard. <http://www.faqs.org/rfcs/rfc3265.html> [Accessed: June 22, 2004].
- [31] A. Fritzler, "AIM/Oscar protocol specification," 2000, <http://aimdoc.sourceforge.net/OSCARdoc/> [Accessed: Feb. 19, 2004].
- [32] Venkat, "Yahoo Messenger Protocol (ver 11)," <http://www.venkydude.com/articles/yahoo.htm> [Accessed: Dec. 7, 2003].
- [33] M. Mintz, "MSN messaging protocol description," hypothetic.org, <http://www.hypothetic.org/docs/msn/index.php> [Accessed: Dec. 7, 2003].
- [34] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM Press, 2002, pp. 161–170.
- [35] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Nov. 2001, FIPS PUB 197 <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf> [Accessed: Aug. 6, 2004].
- [36] D. M. Kienzle and M. C. Elder, "Recent worms: a survey and trends," in *Proceedings of the 2003 ACM Workshop on Rapid Malcode*. Washington, D.C., USA: ACM Press, Oct. 2003, pp. 1–10, <http://pisa.ucsd.edu/worm03/worm2003-program.html> [Accessed: Dec. 7, 2003].
- [37] Gregg Keizer (TechWeb.com), "Symantec: IM worms could spread in seconds," June 2004, <http://www.techweb.com/wire/story/TWB20040618S0007> [Accessed: June 27, 2004].

- [38] Kieren McCarthy (Techworld.com), "Real Player struck by massive security hole," Feb. 2004, <http://www.techworld.com/news/index.cfm?fuseaction=displaynews&NewsID=986> [Accessed: Sep. 15, 2004].
- [39] Joris Evers (Macworld), "Microsoft warns of JPEG threat," Sept. 2004, <http://www.macworld.co.uk/news/index.cfm?NewsID=9635&Page=1&pagePos=2> [Accessed: Sep. 15, 2004].
- [40] R. E. Grinter and L. Palen, "Instant messaging in teen life," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. ACM Press, 2002, pp. 21–30.
- [41] US-CERT, "AOL Instant Messenger contains buffer overflows in parsing of AIM URI handler requests," Jan. 2002, <http://www.kb.cert.org/vuls/id/474592> [Accessed: July 8, 2004].
- [42] D. Petropoulos, "An empirical analysis of RVP-based IM (MSN Messenger Service 3.6)," Encode Security Labs, Nov. 2001, <http://www.encode-sec.com/esp0202.pdf> [Accessed: Dec. 7, 2003].
- [43] E. Isaacs, A. Walendowski, and D. Ranganathan, "Mobile instant messaging through Hubbub," *Commun. ACM*, vol. 45, no. 9, pp. 68–72, 2002.