Computer Networks 56 (2012) 448–461

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Secure instant messaging in enterprise-like networks ☆

Mengjun Xie [a], Zhenyu Wu [b], Haining Wang [b,*]

[a] Computer Science Department, University of Arkansas at Little Rock, Little Rock, AR 72204, United States
[b] Computer Science Department, College of William and Mary, Williamsburg, VA 23185, United States

ARTICLE INFO

Article history:
Received 12 July 2010
Received in revised form 16 May 2011
Accepted 7 September 2011
Available online 17 October 2011

Keywords:
Instant messaging
Security
Enterprise networks

ABSTRACT

Instant messaging (IM) has been one of most frequently used malware attack vectors due to its popularity. However, previous solutions are ineffective to defend against IM malware in an enterprise-like network environment, mainly because of high false positive rate and the requirement of the IM server being inside the protected network. In this paper, we propose a novel IM malware detection and suppression mechanism, HoneyIM, which guarantees almost zero false positive on detecting and blocking IM malware in an enterprise-like network. The detection of HoneyIM is based on the concept of honeypot. HoneyIM uses decoy accounts to trap IM malware by leveraging malware spreading characteristics. Fed with accurate detection results, the suppression of HoneyIM can conduct a network-wide blocking. In addition, HoneyIM delivers attack information to network administrators in real-time so that system quarantine and recovery can be quickly performed. The core design of HoneyIM is generic, and can be applied to the scenarios that either enterprise IM services or public IM services are used in the protected network. Based on open-source IM client `Pidgin` and client honeypot `Capture`, we build a prototype of HoneyIM and validate its efficacy through both simulations and real experiments. Our results show that HoneyIM provides effective protection against IM malware in enterprise-like networks.

1. Introduction

Instant messaging (IM) has been widely used in enterprise environments. According to [2], the daily number of instant messages sent within enterprises around the world is 15 billion in 2009, and will be tripled in 2013, reaching 46 billion. However, large user base and communication immediacy also attract malware to land on IM, which is particularly ideal for malware propagation. By virtue of IM features and social engineering tricks, IM malware can spread quickly and stealthily, which poses a serious security threat not only to home IM users but also to organizations which allow the use of instant messaging in workplace. The IM malware studied in this paper refers to the malicious code that spreads through the Internet-based IM networks such as Windows Live Messenger (formerly named MSN Messenger) and AOL Instant Messenger (AIM), which have dedicated servers for account management and message relay. Bropia [3] that attacks MSN Messenger and Opanki [4] that attacks AIM are two examples of IM malware. Most of known IM malware spreads through public IM networks. Security breaches caused by IM malware not only result in individual system damage and financial losses, but also often seriously degrade the usability of IM service. For example, in November 2010, the spread of IM malware forced Microsoft to temporarily turn off active link functionality in Windows Live Messenger 2009 because the malware propagates through instant messages with malicious URL links [5]. IM malware can also penetrate enterprise IM systems such as IBM Lotus Sametime [6] and Microsoft Lync Server [7] as these corporate IM services usually provide connectivity and

☆ The preliminary version of this paper was appeared in the proceedings of ACSAC 2007 [1].

* Corresponding author.
E-mail addresses: mxxie@ualr.edu (M. Xie), adamwu@cs.wm.edu (Z. Wu), hnw@cs.wm.edu (H. Wang).

1389-1286/$ - see front matter © 2011 Published by Elsevier B.V.
doi:10.1016/j.comnet.2011.09.006

interoperability with public IM services. In 2005, the outbreak of a variant of Kelvir worm even forced Reuters to shut down its IM service [8].

File transfer and URL (Uniform Resource Locator)-embedded message are two major spreading vectors of IM malware. After compromising an IM client, the malware propagates itself by either making a malicious file transfer or sending a text message containing a malicious URL to the online users[1] in the victim's contact list. The contact list is also called buddy list. Once those invigilant contacts click the file or URL, malicious code will be triggered to execute or be downloaded from the URL and executed, and subsequently the malware propagation continues at an exponentially increasing speed.

Although the threat of IM malware, especially the outbreak of zero-day IM malware, is on the rise, network administrators still lack effective solutions to protect enterprise-like networks such as campus networks and corporate networks. Conventional protections using firewalls and anti-virus products are insufficient to defend against IM malware due to the unique propagation feature of IM malware. Most of popular IM protocols are able to circumvent firewalls if their default ports are blocked. Signature-based anti-virus products cannot detect zero-day IM malware. Meanwhile, anomaly detection techniques, such as Norman Sandbox technology [9], may also be ineffective in catching evasive malware which behaves differently in the sandbox environment. Compared to malicious file transfers, malicious-URL-embedded IM messages are even harder to be identified by firewalls and anti-virus programs. Although there exist many URL blacklists such as Google Safe Browsing API [10], SURBL [11], and URIBL [12], a recent IM threat characterization study shows that the majority of malicious URLs sent from IM malware slip through those blacklists [13].

IM providers may take quick responses, e.g., releasing patches and mandating client upgrade, to newly discovered vulnerabilities in their products. They may even proactively block potentially malicious file transfers. However, these filtering mechanisms still could be bypassed [14,15]. Moreover, it is extremely hard for IM providers to protect against malicious URLs that exploit the vulnerabilities of Web browsers or other related applications [16]. While some protection schemes, such as CAPTCHA and virus throttling for IM [17,18], can enhance IM security, the incurred overhead and usability degradation could be significant, and thus prohibit IM providers from using them in near future.

Motivated by the shortage of effective defense against IM malware, we propose HoneyIM, a framework for automating the process of IM malware detection and suppression in an enterprise-like network. Based on the concept of honeypot, HoneyIM detects IM malware by leveraging its inherent spreading characteristics. Specifically, HoneyIM uses decoy accounts in normal users' contact lists as sensors to capture malicious content sent by IM malware, which achieves almost zero false positive. With accurate detection, HoneyIM suppresses malware by performing network-wide blocking. In addition, HoneyIM delivers attack information to network administrators for system quarantine and recovery. The core design of HoneyIM is generic and can be applied to a network that uses either private (enterprise) or public IM services. We implement a prototype of HoneyIM for public IM services, based on open-source IM client `Pidgin` [19] and client honeypot `Capture` [20]. We validate the efficacy of HoneyIM through both simulations and real experiments. The simulations show that even only a small portion, e.g., 5%, of IM users in the network have decoys in their contact lists, HoneyIM can detect the IM malware as early as after 0.4% (on average) of IM users are infected. The experimental results demonstrate that the prototype system succeeds in detection, suppression, and notification of IM malware within seconds.

The remainder of the paper is structured as follows. Section 2 describes the major spreading mechanisms of IM malware and related work. Section 3 presents our measurement study on IM user communication. Section 4 details the framework of HoneyIM, followed by the implementation and evaluation of HoneyIM in Sections 5 and 6, respectively. Section 7 discusses possible evasion to HoneyIM and the countermeasures. Finally, we conclude the paper in Section 8.

2. Background and related work

2.1. IM malware

IM malware propagates mainly through two ways: malicious file transfer and malicious URL in text message. Usually the malware infection is triggered by the victim's action such as clicking the transferred file or the received URL. IM malware could also spread without victim's involvement, e.g., by exploiting the vulnerabilities in IM clients. However, this type of spreading vector is rare.

In the file transfer mechanism that has been used since early 2000s, IM malware propagates by initiating malicious file transfers to remote contacts. Malicious files are usually renamed to attract victims or to evade network filters. Once a victim clicks the file, the malware is invoked and will attempt to infect more victims in the contact list. To counter this type of malware spreading, some IMs such as MSN forbid IM clients to transfer certain types of files such as *.pif* files. While the actual file transfer is normally carried out directly between two IM clients, the messages for transfer establishment still go through IM server. Therefore, IM servers can easily detect the messages for establishing malicious file transfers and silently drop them to block malware propagation.

Nowadays malicious URL messages become much more popular than malicious file transfer for IM malware propagation. Instead of sending a file, IM malware sends a text message containing a malicious URL to remote contacts. Once a victim clicks the link, either a malware binary is downloaded and executed or some malicious web scripts run to exploit the vulnerabilities of the Web browser or other related applications. Compared to malicious file transfers, malicious URL messages have several advantages

[1] Offline contacts may also be attacked but this type of attack is rare.