HoneyIM: Fast Detection and Suppression of Instant Messaging Malware in Enterprise-like Networks

Mengjun Xie Zhenyu Wu Haining Wang
The College of William and Mary
{*mjxie, adamwu, hnw*}*@cs.wm.edu*

Abstract

Instant messaging (IM) has been one of most frequently used malware attack vectors due to its popularity. Distinct from other malware, it is straightforward for IM malware to find and hit the next victim by exploiting the current victim's contact list and playing social engineering tricks. Thus, the spread of IM malware is much harder to detect and suppress through conventional approaches. The 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 stepping into the workplace as well as people's daily life at remarkable speed. It is estimated that enterprise IM users will grow to 78 million by the end of 2008 [9]. 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 IM in workplace. The IM malware studied in the paper refers to any malicious code that spreads through Internet-based IM networks such as Windows Messenger series (MSN) and AOL Instant Messenger (AIM), which have dedicated servers for account management and message relay. Bropia [7] and Opanki [8] are typical examples of such malware. Although most of known IM malware spreads on popular public IM networks, enterprise IM systems such as [6] and [14] can also be penetrated as these corporate IM services usually provide connectivity and interoperability with public IM services. In 2005, the outbreak of a variant of Kelvir worm even forced Reuters to shut down its IM service [4].

File transfer and URL-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 mal-

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

ware. Meanwhile, anomaly detection techniques, such as Norman Sandbox technology [18], 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.

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 [22, 23]. Moreover, it is extremely hard for IM providers to protect against malicious URLs that exploit the vulnerabilities of Web browsers or other related applications [20]. While some protection schemes, such as CAPTCHA and virus throttling for IM [13, 33], 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` [1] and client honeypot `Capture` [29]. 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. We first describe the major spreading mechanisms of IM malware and related work in Section 2. Then we detail the framework of HoneyIM in Section 3, followed by the implementation and evaluation of HoneyIM in Sections 4 and 5, respectively. We discuss possible evasion to HoneyIM and the countermeasures in Section 6. Finally, we conclude the paper in Section 7.

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 in propagation. First, malicious URL messages have more means to compromise a system. File downloading is just one of its attacking vectors. Second, malicious URLs can be used to collect victims' information by exploiting Web functionality. For instance, the URL sent by Kelvir.k [21] points to a php script and contains the contact's email address. The email address is harvested as soon as the URL is clicked. Last but not least, IM malware can play more social engineering tricks on URLs. For example, a malicious URL can be crafted to mimic the link on a reputable Web site [3]. The IM clients supporting HTML scripts also provide a playground for IM malware to fake URLs at their will. Those forged URLs appear normal but in fact point to malicious webpages.

After infection, IM malware may take different actions for propagation. Many types of malware start spreading immediately after they compromise IM clients, while others wait until they receive instructions to spread. The latter usually install certain bot programs on compromised machines,

through which the malware is controlled by the remote bot herder.

2.2. Related Work

The security threats posed by IM malware have been studied in [5, 12]. In [5], the spreading speed of IM malware is estimated, showing that 500,000 machines could be infected within a minute.

Previous defense schemes against IM malware are closely related to IM network modeling and traffic measurement. Based on individual measurement and analysis, [15, 24, 33] all verify that IM social networks formed by IM contacts are scale-free, that is, the IM network connectivities follow power-law distributions. However, a recent measurement study [34] suggests that Weibull distributions may be more appropriate for describing the connectivity of IM social networks. For scale-free networks, a small portion of nodes that are highly connected have significant effect on mitigating malware spread. Based on this observation, Smith [24] proposed to delay the propagation of IM malware by disabling the accounts of most connected IM users on the network. This scheme needs to be deployed on IM servers. It only reduces the spread speed and may have significant side-effects. Williamson *et al.* [33] applied their virus throttling mechanism to IM and demonstrated its effectiveness by simulation. The throttling to IM is also conducted at servers. The throttling becomes blind blocking if its threshold is very restrictive, which degrades the usability. Mannan and van Oorschot [13] proposed two defense methods, namely limited throttling and CAPTCHA-based challenge-response. They also provided a usage study on per-user frequency of IM text messages and file transfers to support the applicability of their second scheme. Liu *et al.* modeled the spread of IM malware using multicast tree [11] and analogous branching process with varied lifetime [10]. HoneyIM is orthogonal to all the schemes mentioned above, and can achieve accurate detection and blocking without degrading usability.

Trivedi *et al.* studied the network and content characteristics of spim, the spam messages on IM networks, by using a proxy server as honeypot [31]. Their work is different from HoneyIM, since [31] is a measurement study and it targets spim but not IM malware. The honeypot used in [31] refers to a SOCKS proxy, which is exploited by spimmers to conceal their identities.

3. HoneyIM Framework

HoneyIM aims to assist network administrators in IM malware defense by automating the process of malware detection and suppression in an enterprise-like network. Utilizing the innate spreading characteristics of IM malware and applying the concept of honeypot, HoneyIM can detect and block unknown IM malware at its early stage of spreading, which greatly facilitates network filtration and system quarantine and recovery. In this section, we first give an overview of HoneyIM, how and why it can detect IM malware early. Then, we discuss several issues that need to be considered when using HoneyIM in practice. After that, we present the design of HoneyIM and the functionalities of its components. Finally, we describe the deployment of HoneyIM in an enterprise-like network.

3.1. Overview

HoneyIM is based on the concept of honeypot. As an effective intrusion detection technology, honeypot has been used widely. According to [30], *a honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource*. Not only can a honeypot be a physical machine or a specialized program, which is the common case, but it can also be an e-mail address, or even an IM decoy user. Since IM malware always attempts to infect other users on the victim's contact list, HoneyIM exploits decoy users to detect IM malware. Under normal circumstances, a client user will not initiate a conversation with a decoy user. Therefore, if the decoy user receives a file transfer request or a URL-embedded text message originated from a client user, it is highly probable that malware is spreading and the request/message sender is compromised. Thanks to decoy users, HoneyIM can achieve almost zero false positive in detection. This strong guarantee, which is rarely offered by other schemes, relieves network administrators from worrying about possible interruption to normal IM users caused by the protection technique. In addition, HoneyIM can block malicious content that has been detected and inform network administrators of the attack information, e.g., the IP address of the compromised machine, in real-time.

Figure 1 illustrates the working mechanism of HoneyIM. The IM user with an icon of honeypot is the one whose contact list contains a decoy user. The events happen in the following sequence. (1) Some IM malware compromises an IM client and (2) propagates. However, (3) when it tries to spread again, it hits a decoy user and (4) is detected by HoneyIM. (5) HoneyIM blocks the malicious content in IM traffic (either at the edge gateway or at the IM server if the IM service is provided within the network) and non-IM traffic[2] instantly, and notifies the attack information to the network administrator.

HoneyIM is designed to be independent, with no restriction on the type and location of IM servers. Therefore, the framework of HoneyIM can be flexibly realized under the

[2]Doing this is to block accesses to malicious contents, e.g., malicious URLs.