A Systematic Approach for Cell-phone Worm Containment

Liang Xie, Hui Song, Trent Jaeger, and Sencun Zhu
Department of Computer Science and Engineering, Penn State University
University Park, Pennsylvania, USA
{lxie, hsong, tjaeger, szhu}@cse.psu.edu

ABSTRACT

Cell phones are increasingly becoming attractive targets of various worms, which cause the leakage of user privacy, extra service charges and depletion of battery power. In this work, we study propagation of cell-phone worms, which exploit Multimedia Messaging Service (MMS) and/or Bluetooth for spreading. We then propose a systematic countermeasure against the worms. At the terminal level, we adopt Graphic Turing test and identity-based signature to block unauthorized messages from leaving compromised phones; at the network level, we propose a push-based automated patching scheme for cleansing compromised phones. Through experiments on phone devices and a wide variety of networks, we show that cellular systems taking advantage of our defense can achieve a low infection rate (e.g., less than 3% within 30 hours) even under severe attacks.

Categories and Subject Descriptors

C.2.m [**Computer-Communication Networks**]: Miscellaneous; C.2.0 [**General**]: Security and Protections

General Terms

Security, Design, Algorithms, Experimentation, Performance

Keywords

Cell phone, Graphic Turing test, Automated patching

1. INTRODUCTION

Mobile communication systems are becoming increasingly important and ubiquitous in people's daily lives. This popularity however comes with a price — mobile networks and terminal devices (e.g., smartphones, PDAs) becoming attractive targets to attackers. Especially, the popularity of mobile services (e.g., email, file transfer, and messaging) and the dependence on common software platforms (e.g., Symbian, Windows CE) make mobile phones ever more vulnerable. According to F-Secure [2], currently there are more than 200 mobile malware (or viruses) in circulation. Examples of the most notorious threats to cell phones include the Skull [8] and Mabir [7] worms, targeting at Symbian-based phone applications. We refer to these malware or viruses as *cell-phone worms*, which are malicious codes that exploit vulnerabilities in cell-phone software and propagate in networks through popular services such as Bluetooth and Multimedia Messaging Service (MMS). Worms are devastating to both users and network. A user can be unconsciously charged for numerous messages generated by the worm and the phone battery will be quickly drained. Other reported worm damages extend from stealing user data and privacy to destroying hardware. For example, a Trojan spy named Flexispy [1] monitors records on its victim's call history and contacts, and delivers these information to a remote server. Therefore, both phone designers and network service providers must employ appropriate defenses against such threats.

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

The problem of quarantining cell-phone worms has not been adequately addressed. Currently, the best defense mirrors the strategy against computer viruses with the inception of security patches for cell-phones. However, it is challenging for users to acquire worm signature files in a timely manner. Some recent work proposes more active solutions. At terminal-device level, Mulliner et al. [9] adopted a labeling technique to prevent cross-service attacks coming from a phone's PDA interface; at network level, Bose et al. [5] proposed to automatically identify compromised phones based on user interactions and suggested a proactive framework to quarantine these suspected devices. Cheng et al. [6] designed a collaborative virus detection and alert system named SmartSiren. These solutions, however, are still not complete because they do not leverage collaborations between the terminals and the network to throttle worm spreads in a systematic way. Moreover, some solutions require deploying external proxies to monitor cell-phone groups [6], which bring extra overhead and alter network architecture.

2. PROPOSED COUNTERMEASURE

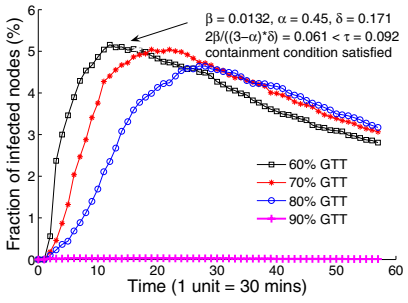
We propose a systematic countermeasure which consists of a terminal-level and a network-level defense.

Terminal-level Defense A smartphone, once infected, will typically generate worm messages, and then stealthily (in the background) deliver them to others. A cellphone user has strong motivation not to let his cell phone become the source of worm attacks to the people in his contact list. On the other hand, a recipient does not want to receive unauthorized messages either because of the potential threats and the related charges upon message receiving (e.g., Verizon and AT&T).

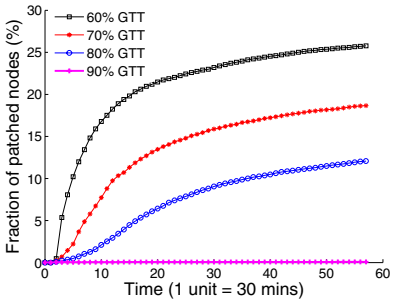
To detect cell-phone worms, we need to differentiate unauthorized worm messages from those authorized ones. The frustrating experience we have learned in combating email spams has already indicated that content-based filtering does not work well. Our idea here is to invoke a CAPTCHA (Completely Automated Public Turing test to Tell Computers and Human Apart) [3] visual test when delivering a newly composed message. A normal user will pass such a visual test easily; however, an automated worm will most probably fail the test. A typical realization of CAPTCHA could be GIMPY, which generates an arbitrary sequence of letters and renders a distorted image of the sequence (as shown in Figure 1). We notice that at the current stage user programs in Symbian cannot directly make system calls; instead, they have to call some APIs to access the resources of the system. This provides us a way to implement the test within an appropriate API (e.g., the *CActiveScheduler:start()* API) that must be called when sending a MMS message.

Although GTTs can be employed to block worms from senders, still the recipients and the network do not know which messages are authorized. We propose that right after a user passes the GTT, a digital signature is used to authenticate message. Both the recipient and the network can verify the signature to decide whether or not to block the message. In general, the system complexity and the cost for deploying a PKI are high. Therefore, we propose to use identity-based

Figure 1: Using graphic turing test (GTT) to block malicious messages from automated worms



(a) Fraction of infected phones vs. time



(b) Fraction of patched phones vs. time

Figure 2: A systematic defense blocks worm attacks and disinfects victims. N=20,000 nodes, MMS network I, detection threshold β_{th} = 25 msg/hour, Bluetooth: λ_b = 1%, v = 0.3 m/s, r = 10m

cryptography [11], more specifically, identity-based signature (IBS). In IBS, a users' identifier information such as email or IP addresses instead of digital certificates can be used as the public key for signature verification. In our context, we propose to use the sender's cellphone number as her public key. Since a receiver knows the sender's cellphone number in a received message, it can easily verify the digital signature in the message, and hence know if the message has passed the sender's GTT.

Network-level Defense The best strategy of network is to block worm messages before they enter the rest part of the network infrastructure. Otherwise, not only those message recipients could be threatened, but also network resources could be greatly wasted. We propose a *push-based automated patching* mechanism, in which the network disinfects origins of malicious messages by automatically pushing software patches to those compromised terminals. Specifically, once a cell phone is suspected being compromised by a cell-phone worm (using the terminal-level defense), its MMS Center (MMSC) immediately notifies a security vendor (e.g., F-Secure and Symantec) through a direct link or an Internet routing infrastructure. Note that the user's phone number is also included in the notification, so that the vendor knows which cell phone is suspicious and it may deliver the latest security patch(s) (worm-signature–based) to disinfect or immunize the cell phone. Specifically, a security patch contains worm information (e.g., worm type and severity level) and the vendor's signature. This patch should be delivered to the phone via a secure data connection such as HTTPS or incrementally using SMS messages. Upon receiving the patch, the user first authenticates the patch origin. If it is from a trusted vendor, the user decides whether to install and activate the patch. A security update usually involves charges. However, recipients have incentive to install them because delivering numerous worm messages to others costs much more.

3. EVALUATIONS

We conducted experiments on Symbian smartphones. We adopted Metrowerks CodeWarrior V3.1.1 as the integrated development environment (IDE). To test compatibility with major phone vendors, we implemented our terminal-level defense using Nokia S60 2rd Edition and Sony-Ericsson UIQ 2.1 (UIQ) for C++ SDKs. To study worm propagation and evaluate the systematic defense, we designed a network simulator in which two MMSCs provide messaging service to 20,000 smartphone users (each for 10,000 users). Both MMSCs are securely connected to a security vendor (e.g., F-secure). As we mentioned earlier, contacts in cell-phone address books form a logical social network among phone users. This network resembles an email network whose topology is typically *heavy-tailed distributed* [10]. We used Barabasi Graph Generator [4] to create power-law graphs and build messaging networks with various degree distributions.

Fig.2 illustrates the results of our systematic defense, which combines the terminal-level and the network-level scheme. We set different percentages of smartphones as GTT–enabled and also launched the network-level protection: worm detection and automated patching. Fig.2 demonstrates that our systematic defense effectively throttles worm spreads in the system. From Fig.2(a), we can see that initially when there are not many victims identified by the network, the worm spread reaches a certain level (around 6%). However, as worm detections and automated patching proceed, compromised phones gradually get disinfected and the infected population starts to decrease. On the other hand, when more phone users adopt GTT (α is higher), worm detection reacts slower as there are less compromised phones. This results in a slower worm extinction (i.e., a longer worm containment time). Fig.2(b) shows the patched population versus time. It suggests that a lower percentage of GTT-enabled phones in the system gets a compensation of faster worm detection and patching from the network. In this sense, components in our systematic defense are complementary to each other.

4. CONCLUSIONS

We predict that worms will become the most devastating threats to terminals and networks as more and more people switch to smartphones. In this work, we have proposed a systematic solution which include both terminal-level and network-level defenses. We showed through smartphone experiments and network simulations that our systematic solution is lightweight, effective and easy to deploy. our approach provides some interesting and practical ways for worm containment in real mobile environments.

Acknowledgement This research was supported by CAREER NSF-0643906.

5. REFERENCES

[1] http://www.f-secure.com/v-descs/flexispy_a.shtml.
[2] http://www.f-secure.com/wireless/threats.
[3] L. Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *EUROCRYPT'03*, 2003.
[4] A. Barabasi and R. Albert. Emergence of scaling in random networks. In *Science, pages 509-512*, Oct., 1999.
[5] A. Bose and K. Shin. Proactive security for mobile messaging networks. In *Proc. of WiSe'06*, 2006.
[6] J. Chen, S. Wongand, H. Yang, and S. Lu. Smartsiren: Virus detection and alert for smartphones. In *Proc. of MobiSys'07*, 2007.
[7] E. Chien. Security response: Symbos.mabir, symantec, 2005.
[8] E. Chien. Security response: Symbos.skull, symantec, 2004.
[9] C. Mulliner, G. Vigna, D. Dagon, and W. Lee. Using labeling to prevent cross-service attacks against smartphones. In *DIMVA'06*, 2006.
[10] M. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. In *Physical Review*, 2002.
[11] A. Shamir. Identity-base cryptosystems and signature schemes. In *Proc. of Crypto'84*, Springer-Verlag, 1984.