Mobile and Ubiquitous Malware

Yean Li Ho
Multimedia University
Jalan Ayer Keroh Lama
75450 Melaka, Malaysia
ylho@mmu.edu.my

Swee-Huay Heng
Multimedia University
Jalan Ayer Keroh Lama
75450 Melaka, Malaysia
shheng@mmu.edu.my

ABSTRACT

Mobile malware is an increasing threat to the world of handheld devices, which can prove to be costlier than PC viruses in the future. The current method used to combat mobile malware is virus signature matching which is based on the slow process of reverse engineering. This paper studies the growth, spread and generic behaviors of mobile and ubiquitous malware in mobile phones. We extend the works of Bose et al. and Schmidt et al. with an additional feature to reduce the effectiveness of mobile malware propagation. The objective of our work is to investigate the trends and generic behavioral patterns of mobile malware and suggest a generic proof-of-concept model which combines the works of Bose et al. and Schmidt et al. with a new feature to slowdown the spread of known and unknown mobile malware which may share similar behavioral patterns.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General - *Security and protection*

General Terms

Security.

Keywords

Mobile Security, Mobile Malware, Mobile Virus, Mobile Worm, Smartphone, Cellphone, PDA, Mobile Phone, Ubiquitous Malware.

1. INTRODUCTION

1.1 Background

A mobile phone, smart phone or personal digital assistant (PDA) infected by mobile malware can be a huge inconvenience to a mobile phone user. A compromised phone can cause its user service interruption, financial loss, privacy and confidentiality loss, slowdown of processing speed, unnecessarily huge consumption of memory and loss of battery power. In some cases, these mobile malware will even spread itself to other phones in the name of the owner of the compromised phone and subsequently ruin reputations at the expense of the owner by sending mass messages via SMS (Short Messaging Service), MMS (Multimedia Messaging Service), email and instant messenger or by spreading the data of the compromised mobile phone to all targeted mobile devices within range via Bluetooth, Infrared and Mobile Web. Data which is sent out through these very same channels without the knowledge of the smart phone owner might even be sensitive or confidential to the owner like personal information, phone numbers, credit card information used in m-commerce, photos, classified documents, e-contact cards, and etc.

1.2 Mobile Malware Research

Due to the proprietary nature and recency of this research area, available literature is limited. Most existing research in Mobile Smartphone Virology focus on detection and identification of smart phone viruses [12,13,16,3,1,11]. So far, not much research has been done to address the issue of finding the solution to these mobile viruses and worms with the exception of [15] and [2]. Xie et al. [15] proposed a method for cell-phone worm containment using the Graphic Turing Test (GTT) to block messages from automated worms. Bose and Shin [2] proposed an automated and proactive framework which quarantines mobile worms and viruses which spreads through Short Messaging Service (SMS) and Instant Messaging networks across platforms including from smart phone to desktop PC and vice versa.

Nevertheless, some research has been done to identify and extract behavioral features of mobile worm and viruses [16,1,11].Yap et al.[16] proposed a proof-of-concept malicious software detection model which monitors any application using message server service. Bose et al.[1] proposed a model to detect and construct mobile malware behavior signatures and classify them by using a machine learning algorithm. Their algorithm is based on the observation that the logical ordering of an application flow over time reveals the malicious intent even when each action alone may appear harmless. Bose et al. [1] studied 25 Symbian-based mobile malware families and generated a database of behavioral patterns using a two-stage mapping classifier technique based on Support Vector Machines (SVMs) which generates signatures from the monitored system events and API calls in Symbian OS at run-time. This algorithm was tested with 96% accuracy [1]. Schmidt et al [11] monitored anomalies in 10 of the most common applications used in smart phones, namely: SMS, game, camera, MMS pictures, PDA functions, Internet, WAP, Bluetooth, email and video camera. They recommended monitoring the following features in order of priority: amount of available RAM, created TCP/IP connections, user idle time in seconds, CPU usage in percent, battery charge level, boolean user idle indicator, amount of available hard disk space, amount of running threads, mobile phone network cell ID, number of installed applications, amount of opened Bluetooth connection, amount of sent SMS, amount of sent MMS and number of received MMS messages.

These detection methods use feature extraction methods to identify signature patterns based on malware behavior rather than specific virus signature matching which can be a rather tedious and slow process. This is a good direction to take in the face of the growing trend of zero-day attacks. These methods can be used

MoMM2009, December 14–16, 2009, Kuala Lumpur, Malaysia.
Copyright 2009 ACM 978-1-60558-659-5/09/0012...$10.00.

to identify potential worms or viruses that may share some similar behavior patterns as previously identified viruses or worms.

2. MOBILE AND UBIQUITOUS MALWARE

2.1 Trends

The only method in practice by antivirus vendors to counterattack mobile smart phone viruses to date is virus signature matching [12,1]. Unfortunately, this is only a stop-gap measure as these signatures require a lot of manual work in reverse engineering to discover virus pattern signatures for each virus that is produced and its variants. In the long run, the signature files will get larger and the existing mobile hardware which has very limited memory, storage space and processing power may not be able to support the increasingly large virus signature list which is used by personal computers today [1]. Furthermore, Morales et al. have done some work on evaluating some existing antivirus software for mobile devices and found that the false negative rates for detection of modified versions of known viruses in the tested software was as high as 47.5% [8].

The first mobile virus appeared in 2004. From about 200 viruses in 2006, the number of mobile viruses has steadily doubled in two years[5]. F-Secure reported that there are more than 400 mobile viruses in circulation as of November 2008 [14]. The growth rate for the virus signature list for mobile viruses in 2 years is equivalent to the growth rate of the virus signature list for personal computers in twenty years [5]. Based on the history of virus evolution in the internet and the speed of mobile smart phone virus evolution [5], this method will also soon prove to be insufficient against zero-day virus attacks which will be a reality in the world of mobile smart phones very soon. Additionally, current mobile devices are unable to support the existing antivirus technologies available for personal computers because of its limited processing power, storage space, memory and battery life.

Therefore, this paper attempts to identify generic behavioral patterns in mobile malware to be applied to a generic behavioral defense model. However, a complete solution to the containment of the spread of mobile and ubiquitous malware is not the objective of this paper although it is a work in progress for future enhancements. The aim of this paper is only to survey the current status of mobile malware and to introduce an extended model to slowdown the spread of mobile and ubiquitous malware and subsequently reduce the effects of its financial damage and cost to the phone user with the compromised mobile phone.

2.2 Categories of Mobile and Ubiquitous Malware

Some of the more popular mobile malware in existence were identified by [7] and [12]. These common mobile malware are briefly listed and described in Table 1.

Table 1. Some popular mobile and ubiquitous malware (Adapted from [3,7,9,10,12])

Mobile Malware	Features
Cabir	Replicates via a Symbian installation file (.SIS file) distributed through Bluetooth, scans for other Bluetooth-enabled devices, reduces battery life or reduces Bluetooth performance
Lasco	Similar to Cabir but can create its own .SIS installer file and infect all .SIS files in the infected Symbian mobile devices, changes the phone“s file directory.
Skulls	A Symbian-based trojan horse which replaces original Symbian binaries used in common applications with non-functional binaries, diables all applications and only allows the device to make and receive phone calls.
Mquito	Sends unauthorised SMS messages to phone numbers in UK, Germany, Switzerland and Holland
Duts	Infects all executables larger than 4 kB, appends itself to a file and disables the application file when it is executed
Metal Gear	A trojan horse which spreads via Bluetooth, disables the antivirus program in the phone and installs Cabir.G
Gavno	A trojan horse which removes critical data in the Symbian OS, causes errors in Nokia 6600 and 6630 phones and reboots the phone
Commwarrior	Similar to Lasco but it can also spread via MMS, sends MMS messages which includes the infected .SIS file to all the recipients in the phone address book
Mabir	Similar to Cabir but it can also spread via MMS, reads the phone address book, monitors received messages and sends fake replies which include a copy of the virus
Phage	Overwrites the beginning of Palm executable files and destroys all installed programs, spreads through infrared or when synchronizing the PalmOS device with a computer
RedBrowser	Java-based trojan horse, sends SMS messages to a phone number
Flexispy	A Symbian-based trojan horse, sends all mobile usage information to a server, FlexiSpy.
CardTrap	Cross-platform virus, disables the mobile system and all third-party applications on the mobile device, infects the memory card with Windows-based PC malware
Doomboot	A trojan horse that installsCommwarrior.B and some corrupted system binaries which causes the device to fail at the next reboot,spreads via Bluetooth
Crossover	A ubiquitous cross-platform virus (mobile and personal computer) which attacks .NET or .NET CF in Windows
Mobler	A ubiquitous trojan horse which spreads through available writable media, disables certain Windows features in a PC and launches a Denial-of-Service attack

Although viruses are usually grouped according to behavior, mobile operating system environment and family of variants [5], mobile viruses are difficult to classify because most mobile malware are hybrids which contain various overlapping features. Nevertheless, we would like to adopt the mode of classification introduced by Cheng et al. [3], which categorizes smart phone viruses according to infection vectors. Table 2 is our extended version of their categorization model.

Table 2. Mobile virus categorization based on infection vectors

Infection Vector	Mobile Virus Samples
Cellular Network (phone calls, SMS, MMS)	CommWarriors, Mabir
Bluetooth	Cabirs,CommWarrior
Infared	Phage
Email	MSIL.Letum
Instant Messaging	Opanki.d
Mobile Web (Internet over WiFi/GPRS/EDGE/ UMTS/3GPP)	Skulls,Doomboot
Ubiquitous Mobile-PC-Mobile (USB/Active Sync / Docking)	Crossover, Mobler
Peipherals (Memory card, SIM card)	Cardtrap

3. PROPOSED MODEL

3.1 General Issues with Current Technology

Existing antivirus solutions are limited because they are specific to a particular platform operating system and to particular phone models. Although this may be a good solution to address the immediate needs for the time being, it is inefficient in the long run because similar patterns of viral attacks in one platform may be replicated against another platform or phone model in future and require double work to reengineer individual solutions to address the same problem previously addressed in other mobile operating systems or phone models. Furthermore, most mobile antivirus engines only support certain versions and models of high-end smart phones. This neglects a large pool of regular phone users who are using phones which may have some of the features or applications of a smart phone but which are too primitive to be classified as a smart phone. Hence, this group of phone users who are also vulnerable to potential mobile phone virus attacks remain unprotected. We propose that a better defense model addresses the needs of this neglected group of users as well.

Since that is the case, it is much more cost-efficient to consider a generic base model which will benefit all platforms and all phone models in the long run. Thus, we are proposing to incorporate the works of Bose et al. [1] and Schmidt et al [11] into a base model which is platform-independent and model-independent. As most mobile operating systems support Java, we present a Java-based model engine which can be used on top of all, if not most mobile operating systems. Although Mobile Web is an important potential virus propagation channel, we will limit the scope of this paper and will not be addressing the issue of possible mobile virus propagation through Mobile Web as this channel involves many variables. Furthermore, the speed, connectivity and usage of the GPRS/EDGE/UMTS/3G networks are not yet as popular or as efficient as the wired networks to warrant the urgency of dealing with this channel at the moment.

Anyhow, most existing mobile phone viruses to date have been known to spread through Bluetooth, MMS and Infrared. These viruses spread by transferring from one phone to another by sending bogus messages via Bluetooth, MMS, SMS or Infrared. This process is usually automated by a virus program which either sends mass SMS messages to a phone number (i.e. Redbrowser for Sun OS or J2ME enabled devices) or mass MMS messages sometimes with a virus installation file (usually a SIS-file for Symbian OS) as an attachment (i.e. Commwarrior and Mabir for Symbian OS), or as attachments through infrared (i.e. Phage for Palm OS) to other infrared enabled devices in the surrounding radius or mass transmissions of the virus installation file (usually a SIS-file for Symbian OS) to neighbouring Bluetooth devices after scanning the area for Bluetooth enabled devices in discoverable mode (i.e. Cabir and Lasco for Symbian OS) [16].

In the cases of Bluetooth and Infrared, the only immediate costs is the spread of the viruses to other smart phones, the slowdown of the processing of legitimate applications on the phone and the increased draining of the phone battery due to the load or even the silent misuse of the compromised phone as a spam producing agent or an agent for Denial of Service (DoS) attacks. In the cases of MMS, SMS, Email and Instant Messaging, the user of the compromised phone will be slapped with a huge bill of unsolicited message charges on his or her phone bill as an added bonus.

3.2 Proposed Extended Feature Model

The algorithm described in Schmidt et al. [11] is an anomaly detection model which extracts features which are unusual or out of the ordinary from regular monitored behaviour of smartphones. Their work was based on Symbian OS and Windows Mobile smartphones. Similarly, the algorithm proposed by Bose et al. [1] conducts a run-time analysis to differentiate between normal program behavior and malware behavior. However, both algorithms require signatures to be generated in run-time and can be bypassed by obfuscating program behavior with behaviour reordering, file or directory renaming, normal behaviour insertion and equivalent behavior replacement.

We propose to combine and extend the models of Bose et al. [1] and Schmidt et al [11] with an additional feature of blocking the silent automated transmission attempts of virus installation files from a compromised mobile smart phone via MMS, Bluetooth, Infrared, Email and Instant Messaging and also block the automated sending of unsolicited MMS, SMS, Email and Instant Messaging messages from a compromised mobile phone. We will be addressing these 6 different propagation channels with one generic model. The process flow of our proposed feature model is as shown in Figure 2.

Our proposed filtering system basically detects if the triggering request to send messages or files is a legitimate manual request from the user or if it is an automated request from a program. The automated request for service is first filtered through a whitelist which is based on a predetermined set of rules quite similar to the concept of a firewall. If it is an automated request which is not found on the whitelist, it is highly likely that it is initiated by a virus or worm program. Thus, the system will assume that it has been caused by a virus bot and will immediately block the transmission of non-manual user initiated transmissions. This idea is inspired by the anomaly detection pseudocode proposed by [11] as shown in Figure 1.

```
GET UserInactivityTime
IF UserInactivityTime – 10s
RETURN User is inactive
   ELSE  RETURN User is active
```

Figure 1. Pseudocode for indicating user activity in [11]

The pseudocode presented by Schmidt et al. [11] detects if a button was pressed within the last 10 seconds. It is a Boolean function which returns "0" if a button was pressed (a user was active) and "1" if a button was not pressed (a user was inactive). This allows the system to track if the activities were caused directly by a user or if it occurred automatically and/or periodically in the background [11]. However, this pseudocode only detects keypad or button input. Our proposed feature model is not limited to button or keypad input, it also detects other forms of manual user input like touchscreen and touchpad input as well. Additionally, their model calculates the number of SMSs and MMSs in the Sent directory and compares the differences in volume. On the other hand, our feature model extends their model by monitoring the output flow of all messages and filters the messages based on the detection of user input. This differentiates the output between those caused by automated bot programs which in all likelihood are caused by malware and actual user output which requires manual action. Therefore, we extend their work and alter their pseudocode to fit our model as shown in Figure 3.

The advantage of this feature model is that it not only maintains the advantages of the anomaly detection models of Schmidt et al. [11] and Bose et al. [1], it strengthens the defense by preventing any unsolicited SMS or MMS from being sent out from a mobile phone without the knowledge and explicit manual action of the mobile phone user. This feature also prevents mobile malware from automatically and silently propagating as executable files through the 6 propagation channels and only allows legitimate

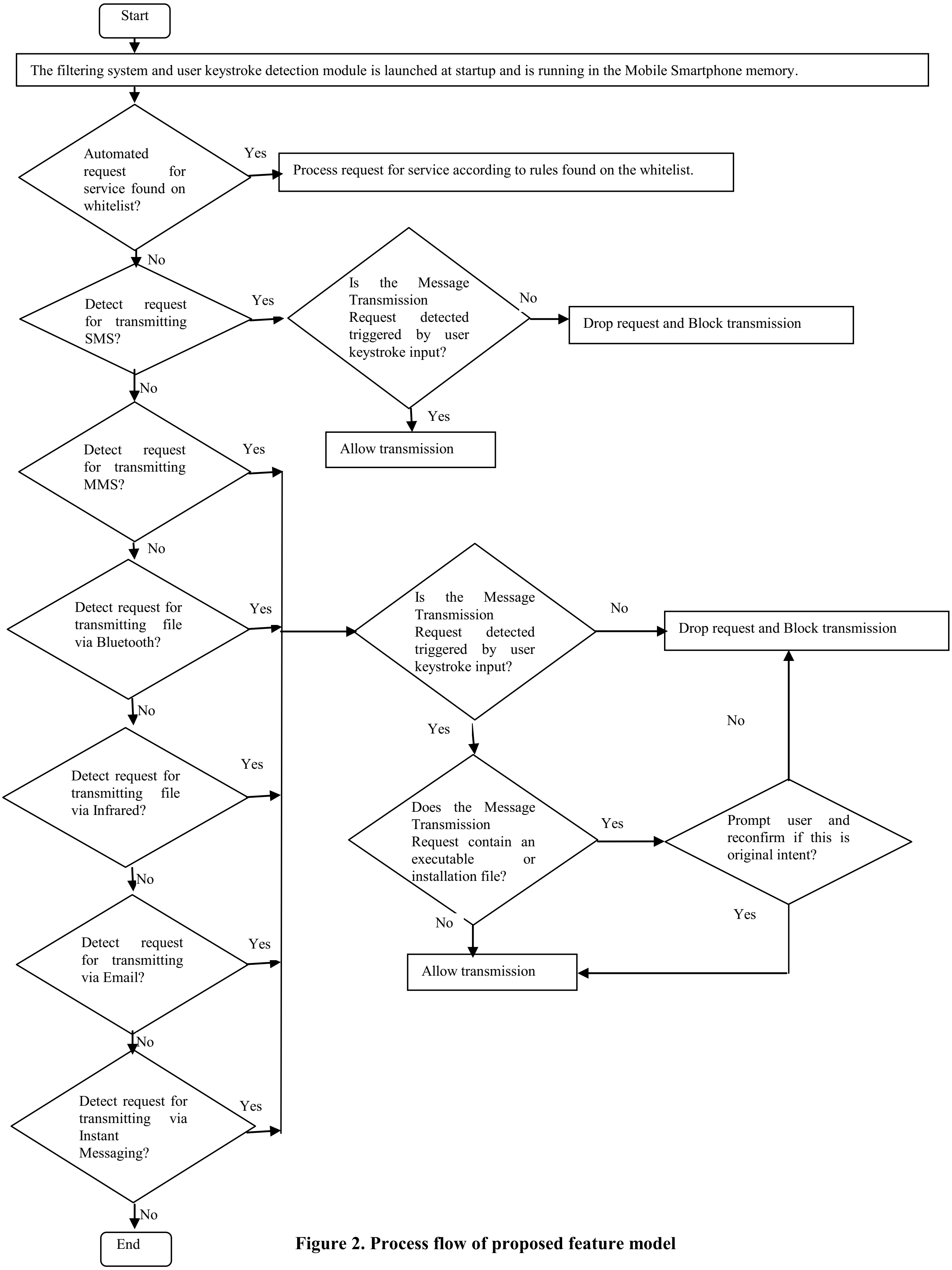
Start
The filtering system and user keystroke detection module is launched at startup and is running in the Mobile Smartphone memory.
Automated request for service found on whitelist?
Yes
Process request for service according to rules found on the whitelist.
No
Detect request for transmitting SMS?
Yes
Is the Message Transmission Request detected triggered by user keystroke input?
No
Drop request and Block transmission
Yes
Allow transmission
No
Detect request for transmitting MMS?
Yes
No
Detect request for transmitting file via Bluetooth?
Yes
Is the Message Transmission Request detected triggered by user keystroke input?
No
Drop request and Block transmission
Yes
No
No
Detect request for transmitting file via Infrared?
Yes
Does the Message Transmission Request contain an executable or installation file?
Yes
Prompt user and reconfirm if this is original intent?
No
Yes
Allow transmission
No
Detect request for transmitting via Email?
Yes
No
Detect request for transmitting via Instant Messaging?
Yes
No
End

Figure 2. Process flow of proposed feature model

```
IF MessageOutputTriggered
Check Whitelist
IF ServiceRequest not found on whitelist
   GET UserInput
   IF UserInput is FALSE
         Block and Delete message or File Transfer request
   ELSE   Transmit message
ELSE Process ServiceRequest according to rules on whitelist
```

Figure 3. Generic pseudocode for indicating user activity in our model

transmissions with explicit permission from the user. This model blocks all transmission of files and messages which have not been manually sent by a user or knowingly allowed by the whitelist. It detects if a mobile phone is sending any installation or executable files and prompts a user to confirm if the action was intended by a user or not. This is based on the assumption that legitimate applications using these 6 communication channels require user action to activate or launch the programs. Hence, this added feature not only helps contain the spread of mobile malware from mobile phone to smart phone, or PDAs, it also helps contain the spread of potential ubiquitous malware which spreads across multiple platforms through Infrared and Bluetooth channels to involve personal computers, and laptops i.e. Crossover, Opanki.d and Mobler. Another advantage of this model is that it also addresses the privacy and confidentiality issue of smart phone users.

While previously the only solution was to educate users to not install confidential files in a mobile devices [4], this model begins to address the solution to the issue of protecting the privacy and secrecy of confidential corporate files which may be stored on a smart phone or PDA. Although this model is limited against social engineering, it also prevents a mobile device from sending out any files from the mobile device through the 6 channels without the explicit permission and manual action of the mobile phone user. The side-effect of this is that this containment model helps reduce the effects of the financial damage and cost to the phone user with a compromised mobile phone.

4. CONCLUSION

We have proposed to extend the algorithms of Bose et al. [1] and Schmidt et al. [11] to include an additional feature model to hinder the effectiveness of mobile malware propagation and to slowdown the spread of mobile viruses or worms. While most of the work on mobile malware are specific to individual platforms, we have proposed a generic proof-of-concept countermeasure. Rather than just focusing on detection, our model complements and extends the detection models with a solutions model based on behavioral patterns of mobile malware. This feature model automatically contains and slows down the spread of known and unknown mobile and ubiquitous malware and is independent of the need to identify the existence of the particular malware in the mobile phone. This saves a lot of processing power in the mobile phone. Thus, a more complete solution to the containment of the spread of mobile viruses that includes propagation through the Mobile Web and peripherals is a work in progress for future enhancements to this model. We will also extend this work further to include performance testing as well.

5. REFERENCES

[1] Bose, A., Hu, X., Shin, K.G., and Park, T. 2008. Behavioral detection of malware on mobile handsets. MobiSys‘08. 225-238.

[2] Bose, A., and Shin, K.G. 2006. Proactive security for mobile messaging networks. Workshop on Wireless Security Proceedings of the 5th ACM workshop on Wireless security. 95-104.

[3] Cheng, J., Wong, S.H.Y., Yang, H., and Lu, S. 2007. SmartSiren:Virus detection and alert for smartphones. MobiSys’07. 258 – 271.

[4] Cheng, Z. Mobile malware: Threats and prevention. McAfee, Available at http://www.mcafee.com/us/local_content/white_ papers/threat_center/wpmalware_r2_en.pdf

[5] Gostev, A. 29 September 2006. Mobil emalware evolution. Viruslist.com.Available at http://www.viruslist.com/en/analysis?pubid=200119916

[6] Kentynet. 11 November 2008. Smartphone. Wikipedia. http://en.wikipedia.org/wiki/Smartphone

[7] Leavitt, N. 2005. Mobile phones: The next frontier for hackers? Technology News. Available at http://ieeexplore.ieee.org/stamp/ stamp.jsp?arnumber =01432639.

[8] Morales, J.A., Clarke, P.J., Deng, Y., and Kibria, B.M.G. 2006.Testing and evaluating virus detectors for handheld devices. Journal in Computer Virology 2(2).135 – 147. doi:10.1007/s11416-006-0024-y.

[9] Niemela, J. September 2005. F-Secure Virus Descriptions: Doomboot.A.Available at http://www.f-secure.com/v-descs/doomboot_a.shtml

[10] Peikari, C. 8 March 2006. Analyzing the Crossover Virus: The First PC to Windows Handheld Cross-infector. Available at http://www.informit.com/articles/article.aspx?p=458169&seqNum=5

[11] Schmidt, A-D, Peters, F., Lamour, F., Scheel, C., Camtepe, S.A., and Albayrak, S. November 2008. Monitoring smartphones for anomaly detection. ACM International Conference Proceeding Series; Vol. 278, Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Application. 92-106.

[12] Toyssy, S., and Helenius, M. 2006. About malicious software in smartphones. Journal in Computer Virology 2(2).109-119.

[13] Venugopal, D. 2006. An efficient signature representation and matching method for mobile devices. WICON’06.

[14] Williams, I. 13 November 2008. F-Secure warns of mobile malware growth. businessGreen.com

[15] Xie, L., Song, H., Jaeger, T., and Zhu, S. 2007. A systematic approach for cell-phone worm containment. Workshop On Rapid Malcode Proceedings of the 2007ACM workshop on Recurring malcode. 61 – 68.

[16] Yap, T.S., and Ewe, H.T. 2005. A mobile phone malicious software detection model with behavior checker. HSI2005. LNCS 3597. 57-65.