International Journal of Computer Science and Business Informatics

IJCSBI.ORG

A Survey on Mobile Malware: A War without End

Sonal Mohite
Sinhgad College of Engineering,
Vadgaon. Pune, India.

Prof. R. S. Sonar
Associate Professor
Sinhgad College of Engineering,
Vadgaon. Pune, India.

ABSTRACT
Nowadays, mobile devices have become an inseparable part of our everyday lives and its usage has grown up exponentially. With the functionality upgrade of mobile phones, the malware threat for mobile phones is expected to increase. This paper shades a light on when and how the mobile malware got evolved. Current scenario of mobile operating system shares' and number and types of mobile malware are also described. Mobile malware can be propagated via three communication media viz. SMS/MMS, Bluetooth/Wi-Fi and FM-RDS. Several mobile malware detection techniques are explained with implemented examples. When one uses the particular malware detection technique is clarified along with its pros & cons. At first, static analysis of application is done and then a dynamic analysis. If external ample resources are available then cloud-based analysis is chosen. Application permission analysis and battery life monitoring are novel approaches of malware detection. Along with malware detection, preventing mobile malware has become critical. Proactive and reactive techniques of mobile malware control are defined and explained. Few tips are provided to restrain malware propagation. Ultimately, Structured and comprehensive overview of the research on mobile malware is explored.

Keywords
Mobile malware, malware propagation, malware control, malware detection.

1. INTRODUCTION
Before decades, computers were the only traditional devices used for computing. Here and now, smart phones are used as supporting computing devices with computers. With the increasing capabilities of such phones, malware which was the biggest threat for computers is now become widespread for smart phones too. The damage made by mobile malwares includes theft of confidential data from device, eavesdropping of ongoing conversation by third party, incurring extra charges through sending SMS to premium rate numbers, and even location based tracking of user, which is too severe to overlook. So there is a judicious requirement of understanding the propagation means of mobile malware, various techniques to detect mobile malware, and malware restraint.

2. RELATED WORKS

Malware is a malicious piece of software which is designed to damage the computer system & interrupt its typical working. Fundamentally, malware is a short form of Malicious Software. Mobile malware is a malicious software aiming mobile phones instead of traditional computer system. With the evolution of mobile phones, mobile malware started its evolution too [1-4].

When propagation medium is taken into account, mobile viruses are of three types: Bluetooth-based virus, SMS-based virus, and FM RDS based virus [5-9]. A BT-based virus propagates through Bluetooth & Wi-Fi which has regional impact [5], [7], and [8]. On the contrary, SMS-based virus follows long-range spreading pattern & can be propagated through SMS & MMS [5], [6], [8]. FM RDS based virus uses RDS channel of mobile radio transmitter for virus propagation [9]. Our work addresses the effect of operational behavior of user & mobility of a device in virus propagation.

There are several methods of malware detection viz. static method, dynamic method, cloud-based detection method, battery life monitoring method, application permission analysis, enforcing hardware sandbox etc. [10-18]. In addition to work given in [10-18], our work addresses pros and cons of each malware detection method. Along with the study of virus propagation & detection mechanisms, methods of restraining virus propagation are also vital. A number of proactive & reactive malware control strategies are given in [5], [10].

3. EVOLUTION OF MOBILE MALWARE

Although, first mobile malware, 'Liberty Crack', was developed in year 2000, mobile malware evolved rapidly during years 2004 to 2006 [1]. Enormous varieties of malicious programs targeting mobile devices were evolved during this time period & are evolving till date. These programs were alike the malware that targeted traditional computer system: viruses, worms, and Trojans, the latter including spyware, backdoors, and adware. At the end of 2012, there were 46,445 modifications in mobile malware. However, by the end of June 2013, Kaspersky Lab had added an aggregate total of 100,386 mobile malware modifications to its system [2]. The total mobile malware samples at the end of December 2013 were 148,778 [4]. Moreover, Kaspersky labs [4] have collected 8,260,509 unique malware installation packs. This shows that there is a dramatic increase in mobile malware. Arrival of 'Cabir', the second most mobile malware (worm) developed in 2004 for Symbian OS, dyed-in-the-wool the basic rule of computer virus evolution. Three conditions are needed to be fulfilled for malicious programs to target any particular operating system or platform:

- The platform must be popular: During evolution of 'Cabir', Symbian was the most popular platform for smart phones. However, nowadays it is Android, that is most targeted by attackers. These days' malware authors continue to ponder on the Android platform as it holds 93.94% of the total market share in mobile phones and tablet devices.
- There must be a well-documented development tools for the application: Nowadays every mobile operating system developers provides a software development kit & precise documentation which helps in easy application development.
- The presence of vulnerabilities or coding errors: During the evolution of 'Cabir', Symbian had number of loopholes which was the reason for malware intrusion. In this day and age, same thing is applicable for Android [3].

Share of operating system plays a crucial role in mobile malware development. Higher the market share of operating system, higher is the possibility of malware infection. The pie chart below illustrates the operating system (platform) wise mobile malware distribution [4]:

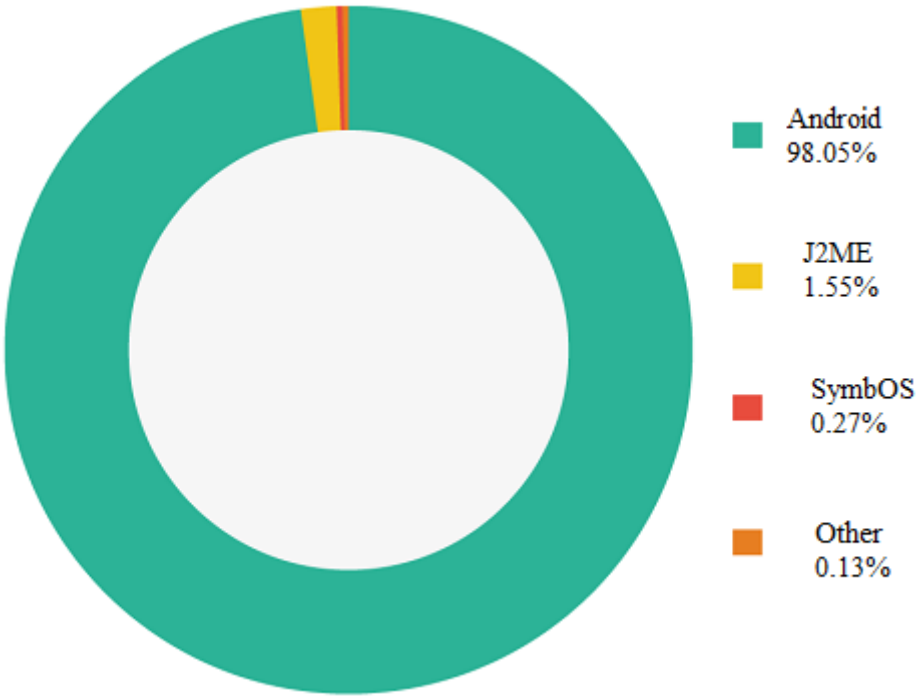

Figure 1. OS wise malware distribution

4. MOBILE MALWARE PROPAGATION

There are 3 communication channels through which malware can propagate. They are: SMS / MMS, Bluetooth / Wi-Fi, and FM Radio broadcasts.

4.1 SMS / MMS

Viruses that use SMS as a communication media can send copies of themselves to all phones that are recorded in victim's address book. Virus can be spread by means of forwarding photos, videos, and short text messages, etc. For propagation, a long-range spreading pattern is followed which is analogous to the spreading of computer viruses like worm propagation in e-mail networks [6]. For accurate study of SMS-based virus propagation, one needs to consider certain operational patterns, such as whether or not users open a virus attachment. Hence, the operational behavior of users plays a vital role in SMS-based virus propagation [8].

4.1.1 Process of malware propagation

If a phone is infected with SMS-based virus, the virus regularly sends its copies to other phones whose contact number is found in the contact list of the infected phone. After receiving such distrustful message from others, user may open or delete it as per his alertness. If user opens the message, he is infected. But, if a phone is immunized with antivirus, a newly arrived virus won't be propagated even if user opens an infected message. Therefore, the security awareness of mobile users plays a key role in SMS-based virus propagation.

Same process is applicable for MMS-based virus propagation whereas MMS carries sophisticated payload than that of SMS. It can carry videos, audios in addition to the simple text & picture payload of SMS.

4.2 Bluetooth/ Wi-Fi

Viruses that use Bluetooth as a communication channel are local-contact driven viruses since they infect other phones within its short radio range. BT-based virus infects individuals that are homogeneous to sender, and each of them has an equal probability of contact with others [7]. Mobility characteristics of user such as whether or not a user moves at a given hour, probability to return to visited places at the next time, traveling distances of a user at the next time etc. are need to be considered [8].

4.2.1 Process of malware propagation

Unlike SMS-based viruses, if a phone is infected by a BT-based virus, it spontaneously & atomically searches another phone through available Bluetooth services. Within a range of sender mobile device, a BT-based virus is replicated. For that reason, users' mobility patterns and contact

frequency among mobile phones play crucial roles in BT-based virus propagation.

Same process is followed for Wi-Fi where Wi-Fi is able to carry high payload in large range than that of BT.

4.3 FM-RDS

Several existing electronic devices do not support data connectivity facility but include an FM radio receiver. Such devices are low-end mobile phones, media players, vehicular audio systems etc. FM provides FM radio data system (RDS), a low-rate digital broadcast channel. It is proposed for delivering simple information about the station and current program, but it can also be used with other broad range of new applications and to enhance existing ones as well [9].

4.3.1 Process of malware propagation

The attacker can attack in two different ways. The first way is to create a seemingly benign app and upload it to popular app stores. Once the user downloads & installs the app, it will contact update server & update its functionality. This newly added malicious functionality decodes and assembles the payload. At the end, the assembled payload is executed by the Trojan app to uplift privileges of attacked device & use it for malicious purpose. Another way is, the attacker obtains a privilege escalation exploit for the desired target. As RDS protocol has a limited bandwidth, we need to packetize the exploit. Packetization is basically to break up a multi-kilobyte binary payload into several smaller Base64 encoded packets. Sequence numbers are attached for proper reception of data at receiver side. The received exploit is executed. In this way the device is infected with malware [9].

5. **MOBILE MALWARE DETECTION TECHNIQUE**

Once the malware is propagated, malware detection is needed to be carried out. In this section, various mobile malware detection techniques are explained.

5.1 Static Analysis Technique

As the name indicates, static analysis is to evaluate the application without execution [10-11]. It is an economical as well as fast approach to detect any malevolent characteristics in an application without executing it. Static analysis can be used to cover static pre-checks that are performed before the application gets an entry to online application markets. Such application markets are available for most major smartphone platforms e.g. ‘Play store’ for Android, ‘Store’ for windows operating system. . These extended pre-

checks enhance the malware detection probabilities and therefore further spreading of malware in the online application stores can be banned. In static analysis, the application is investigated for apparent security threats like memory corruption flaws, bad code segment etc. [10], [12].

5.1.1 Process of malware detection

If the source code of application is available, static analysis tools can be directly used for further examination of code.

But if the source code of the application is not available then executable app is converted back to its source code. This process is known as disassembling. Once the application is disassembled, feature extraction is done. Feature extraction is nothing but observing certain parameters viz. system calls, data flow, control flow etc. Depending on the observations, anomaly is detected. In this way, application is categorized as either benign or malicious.

***Pros*:** Economical and fast approach of malware detection.

***Cons*:** Source codes of applications are not readily available. And disassembling might not give exact source codes.

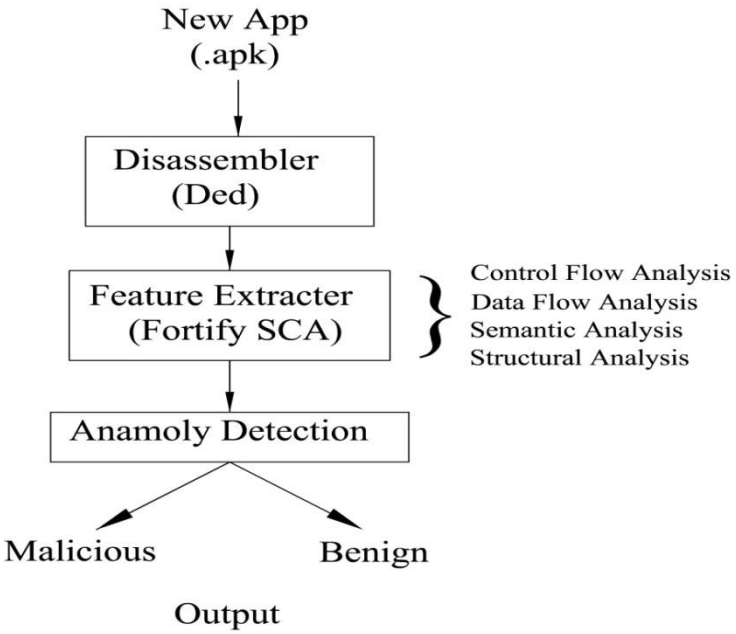

Figure 2. Static Analysis Technique

5.1.2 Example

Figure 2 shows the malware detection technique proposed by Enck et al. [12] for Android. Application's installation image (.apk) is used as an input to system. *Ded*, a Dalvik decompiler, is used to dissemble the code. It

generates Java source code from .apk image. Feature extraction is done by using Fortify SCA. It is a static code analysis suite that provides four types of analysis; control flow analysis, data flow analysis, structural analysis, and semantic analysis. It is used to evaluate the recovered source code & categorize the application as either benign or malicious.

5.2 Dynamic Analysis Technique

Dynamic analysis comprises of analyzing the actions performed by an application while it is being executed. In dynamic analysis, the mobile application is executed in an isolated environment such as virtual machine or emulator, and the dynamic behavior of the application is monitored [10], [11], [13]. There are various methodologies to perform dynamic analysis viz. function call monitoring, function parameter analysis, Information flow tracking, instruction trace etc. [13].

5.2.1 Process of malware detection

Dynamic analysis process is quite diverse than the static analysis. In this, the application is installed in the standard Emulator. After installation is done, the app is executed for a specific time and penetrated with random user inputs. Using various methodologies mentioned in [13], the application is examined. On the runtime behavior, the application is either classified as benign or malicious.

Pros: Comprehensive approach of malware detection. Most of the malwares is got detected in this technique.

Cons: Comparatively complex and requires more resources.

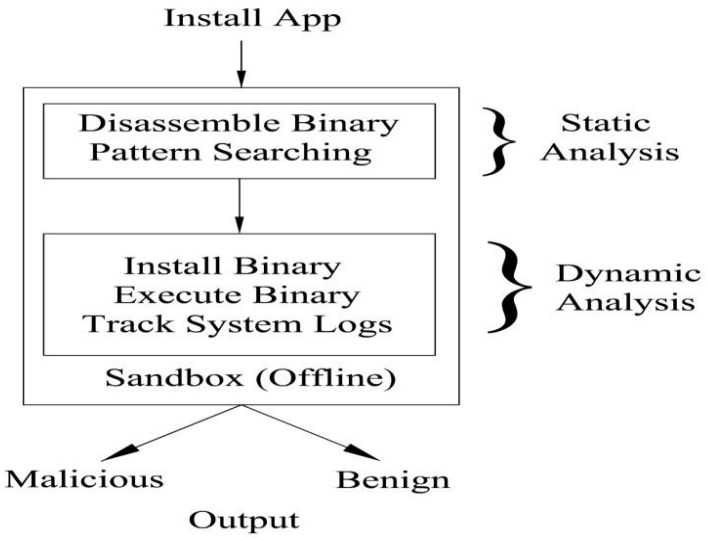

Figure 3. Dynamic Analysis Technique

5.2.2 Example

Figure 3 shows Android Application Sandbox (AASandbox) [14], the dynamic malware detection technique proposed by Blasing et al. for Android. It is a two-step analysis process comprising of both static & dynamic analysis. The AASandbox first implements a static pre-check, followed by a comprehensive dynamic analysis. In static analysis, the application image binary is disassembled. Now the disassembled code is used for feature extraction & to search for any distrustful patterns. After static analysis, dynamic analysis is performed. In dynamic analysis, the binary is installed and executed in an AASandbox. 'Android Monkey' is used to generate runtime inputs. System calls are logged & log files are generated. This generated log file will be then summarized and condensed to a mathematical vector for better analysis. In this way, application is classified as either benign or malicious.

5.3 Cloud-based Analysis Technique

Mobile devices possess limited battery and computation. With such constrained resource availability, it is quite problematic to deploy a full-fledged security mechanism in a smartphone. As data volume increases, it is efficient to move security mechanisms to some external server rather than increasing the working load of mobile device [10], [15].

5.3.1 Process of malware detection

In the cloud-based method of malware detection, all security computations are moved to the cloud that hosts several replicas of the mobile phones running on emulators & result is sent back to mobile device. This increases the performance of mobile devices.

Pros: Cloud holds ample resources of each type that helps in more comprehensive detection of malware.

Cons: Extra charges to maintain cloud and forward data to cloud server.

5.3.2 Example

Figure 4 shows Paranoid Android (PA), proposed by Portokalidis et al. [15]. Here, security analysis and computations are moved to a cloud (remote server). It consists of 2 different modules, a tracer & replayer. A tracer is located in each smart phone. It records all necessary information that is required to reiterate the execution of the mobile application on remote server. The information recorded by tracer is first filtered & encoded. Then it is stored properly and synchronized data is sent to replayer over an encrypted channel. Replayer is located in the cloud. It holds the replica of mobile phone running on emulator & records the information communicated by tracer. The replayer replays the same execution on the emulator, in the

cloud. Cloud, the remote server, owns abundant resources to perform multifarious analysis on the data collected from tracer. During the replay, numerous security analyses such as dynamic malware analysis, memory scanners, system call tracing, call graph analysis[15] etc. are performed rather there is no limit on the number of attack detection techniques that we can be applied in parallel.

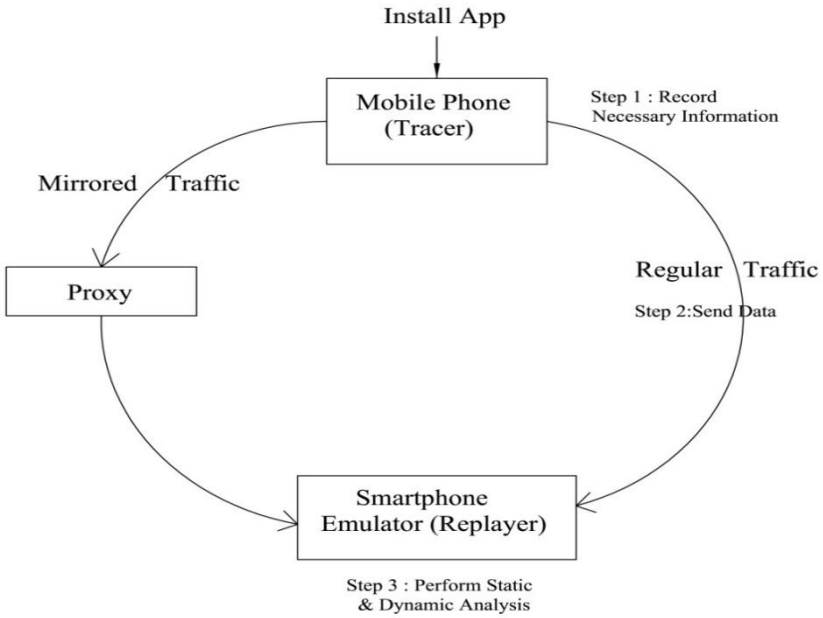

Figure 4. Cloud-based Detection Technique

5.4 Monitoring Battery Consumption

Monitoring battery life is a completely different approach of malware detection compared to other ones. Usually smartphones possess limited battery capacity and need to be used judiciously. The usual user behavior, existing battery state, signal strength and network traffic details of a mobile is recorded over time and this data can be effectively used to detect hidden malicious activities. By observing current energy consumption such malicious applications can indeed be detected as they are expected to take in more power than normal regular usage. Though, battery power consumption is one of the major limitations of mobile phones that limit the complexity of anti-malware solutions. A quite remarkable work is done in this field. The introductory exploration in this domain is done by Jacoby and Davis [16].

5.4.1 Process of malware detection

After malware infection, that greedy malware keeps on repeating itself. If the mean of propagation is Bluetooth then the device continuously scans for

adjacent Bluetooth-enabled devices which in turn consume a remarkable amount of power. This time-domain data of power consumption collected over a period of time is transformed into frequency-domain data & represented as dominant frequencies. The malwares are identified from these certain dominant frequencies.

Pros: Economical and novel approach of malware detection.

Cons: Because of multi-functionality of smart phones, power consumption model of smart phone could not be accurately defined.

5.4.2 Example

Recent work by Liu et al. [17] proposed another detection technique by comparing the compressed sequences of the power consumption value in each time interval. They defined a user-centric power model that relies on user actions. User actions such as duration & frequency of calls, number of SMS, network usage are taken into account. Their work uses machine learning techniques to generate rules for malware detection.

5.5 Application Permission Analysis

With the advancements in mobile phone technology, users have started downloading third party application. These applications are available in third party application stores. While developing any application, application developers need to take required permissions from device in order to make the application work on that device. Permissions hold a crucial role in mobile application development as they convey the intents and back-end activities of the application to the user. Permissions should be precisely defined & displayed to the user before the application is installed. Though, some application developers hide certain permissions from user & make the application vulnerable & malicious application.

5.5.1 Process of malware detection

Security configuration of an application is extracted. Permissions taken by an application are analyzed. If application has taken any unwanted applications then it is categorized as malicious.

Pros: Fewer resources are required compared to other techniques.

Cons: Analyzing only the permissions request is not adequate for mobile malware detection; it needs to be done in parallel with static and/or dynamic analysis.

5.5.2 Example

Kirin, proposed by Enck et al. (2009) [18] is an application certification system for Android. During installation, Kirin crisscrosses the application permissions. It extracts the security configurations of the application

&checks it against the templates i.e. security policy rules already defined by Kirin. If any application becomes unsuccessful to clear all the security policy rules, Kirin either deletes the application or alerts the user for assistance [18].

6. MOBILE MALWARE CONTROL STRATEGIES

Basically, there are two types of malware control strategies, viz. proactive & reactive control. In proactive malware control strategy, malware is mitigated before its propagation. Proper set of preventive measures is used for this purpose. While, in reactive malware control strategy, malware is first propagated and then a reaction is taken upon malware contamination.

6.1 Proactive Malware Control Strategy

Here are some of the proactive malware control techniques given in [10]; however, users' own security awareness plays a crucial role.

- Install a decent mobile security application i.e. antivirus.
- Always download apps from trusted official application markets. Before downloading any app, do read the reviews and ratings of the app. During installation, always remember to read the permissions requested by the app and if it appears doubtful don't install it. Always keep installed apps up-to-date.
- Turn-off Wi-Fi, Bluetooth, and other short range wireless communication media when not to be used. Stay more conscious when connecting to insecure public Wi-Fi networks & accepting Bluetooth data from unknown sender.
- When confidential data is to be stored in the mobile phone, encrypt it before storing and set a password for access. Do regular back-ups. Assure that the sensitive information is not cached locally in the mobile phone.
- Always keep an eye on the battery life, SMS and call charges, if found any few and far between behaviors, better go for an in-depth check on the recently installed applications.
- During internet access, don't click on links that seem suspicious or not trustworthy.
- Finally, in case of mobile phone theft, delete all contacts, applications, and confidential data remotely.

6.2Reactive Malware Control Strategy

When the malware is detected then the control strategy is implemented, is the working principle of reactive malware control strategy. Antivirus solution comes under proactive malware control, however when a new

malware is found, antivirus updates for that malware are implemented and forwarded to mobile phones, is a part of reactive malware control. This is known as adaptive patch dissemination.

Adaptive Patch Dissemination

A pre-immunization like antivirus is used to protect networks before virus propagation. However, in reality, we first detect certain viruses and then update antivirus, known as patches. These patches are forwarded into networks only after these viruses have already propagated. Network bandwidth limits the speed with which the security notifications or patches can be sent to all users simultaneously. Therefore, a new strategy namely adaptive dissemination strategy is developed. It is based on the Autonomy Oriented Computing (AOC) methodology which helps to send security notifications or patches to most of phones with a relatively lower communication cost. The AOC is used to search a set of the highly connected phones with large communication abilities in a mobile network [5].

7. CONCLUSION

Rapid growth in smart phone development resulted in evolution of mobile malware. Operating system shares' plays crucial role in malware evolution. SMS/MMS is the fastest way of mobile malware propagation as it has no geographical boundary like BT/Wi-Fi. FM-RDS is still evolving. Among all malware detection techniques, static malware detection is performed first during pre-checks. Later dynamic analysis is performed and can be combined with application permission analysis. Cloud-based analysis is more comprehensive approach as it uses external resources to perform malware detection and can perform more than one type of analysis simultaneously. Proactive control strategy is used to control malware before its propagation while reactive control strategy is used after malware is propagated.

REFERENCES

[1] La Polla, M., Martinelli, F., & Sgandurra, D. (2012). A survey on security for mobile devices. *IEEE Communications Surveys & Tutorials*, 15(1), 446 – 471.

[2] Kaspersky Lab IT Threat Evolution: Q2 2013. (2013). Retrieved from http://www.kaspersky.co.in/about/news/virus/2013/kaspersky_lab_it_threat_evolution_q2_2013.

[3] Kaspersky Security Bulletin 2013: Overall statistics for 2013. (2013 December). Retrieved from

http://www.securelist.com/en/analysis/204792318/Kaspersky_Security_Bulletin_2013_Overall_statistics_for_2013.

[4] Maslennikov, D. Mobile Malware Evolution: Part 6. (2013 February). Retrieved from http://www.securelist.com/en/analysis/ 204792283/Mobile_Malware_Evolution_Part_6.

[5] Gao, C., and Liu, J. (2013). Modeling and restraining mobile virus propagation. *IEEE transactions on mobile computing*, 12(3), 529-541.

[6] Gao, C. and Liu, J. (2011). Network immunization and virus propagation in Email networks: Experimental evaluation and analysis. *Knowledge and information systems*, 27(2), 253-279.

[7] Yan, G., and Eidenbenz, S. (2009, March). Modeling propagation dynamics of Bluetooth worms (extended version). *IEEE transactions on Mobile Computing*, 8(3), 353-368.

[8] Gonzalez, M., Hidalgo, C., and Barabasi, A. (2008). Understanding individual human mobility patterns. *Nature*, 453(7196), 779-782.

[9] Fernandes, E., Crispo, B., Conti, M. (2013, June). FM 99.9, Radio virus: Exploiting FM radio broadcasts for malware deployment. *Transactions on information forensics and security*, 8(6), 1027-1037.

[10] Chandramohan, M., and Tan, H. (2012). Detection of mobile malware in the wild. *IEEE computer society*, 45(9), 65-71.

[11] Yan, Q., Li, Y., Li, T., and Deng, R. (2009). Insights into malware detection and prevention on mobile phones. *Springer-Verlag Berlin Heidelberg, SecTech 2009*, 242–249.

[12] Enck, W., Octeau, D., Mcdaniel, P., and Chaudhuri, S. (2011 August). A study of android application security. *The 20th Usenix security symposium*.

[13] Egele, M., Scholte, T., Kirda, E., Kruegel, C. (2012 February). A survey on automated dynamic malware-analysis techniques and tools. *ACM-TRANSACTION*, 4402(06), 6-48.

[14] Blasing, T., Batyuk, L., Schmidt, A., Camtepe, S., and Albayrak, S. (2010). An android application sandbox system for suspicious software detection. *5th International Conference on Malicious and Unwanted Software.*

[15] Portokalidis, G., Homburg, P., Anagnostakis, K., Bos, H. (2010 December). Paranoid android: Versatile protection for smartphones. *ACSAC'10.*

[16] Jacoby, G. (2004). Battery-based intrusion detection. *The Global Telecommunications Conference.*

[17] Liu, L., Yan, G., Zhang, X., and Chen, S. (2009). Virusmeter: Preventing your cellphone from spies. *RAID*, 5758, 244-264.

[18] Enck, W., Ongtang, M., and Mcdaniel, P. (2009 November). On lightweight mobile phone application certification. *16th ACM Conference on Computer and Communications Security.*

This paper may be cited as:

Mohite, S. and Sonar, R. S., 2014. A Survey on Mobile Malware: A War without End. *International Journal of Computer Science and Business Informatics, Vol. 9, No. 1, pp. 23-35.*