"MOBILE MALWARE DETECTION TECHNIQUES"

Vinit B. Mohata[1]

[1] 2nd SEM M.E, Department of Computer Science and Engineering
Sipna College of Engineering and Technology
Amravati, India
vinit.mohata@gmail.com

Dhananjay M. Dakhane[2]

[2] Associate Professor, Department of Computer Science and Engineering
Sipna College of Engineering and Technology
Amravati, India
ddakhane@gmail.com

Ravindra L.Pardhi[3]

[3] Assistant Professor, Department of Computer Science and Engineering
Sipna College of Engineering and Technology
Amravati, India
rlpardhi@gmail.com

Abstract— The malware threat for mobile phones is expected to increase with the functionality enhancement of mobile phones. This threat is exacerbated with the surge in population of smart phones instilled with stable Internet access which provides attractive targets for malware developers. Currently, in the smartphone market, Android is the platform with the highest share. Due to this popularity and also to its open source nature, Android-based smartphones are now an ideal target for attackers. Since the number of malware designed for Android devices is increasing fast, Android users are looking for security solutions aimed at preventing malicious actions from damaging their smartphones. Prior research on malware protection has focused on avoiding the negative impact of the functionality limitations of mobile phones to keep the performance cost within the limitations of mobile phones. From this perspective, we propose and analyze some potential limitation-oriented techniques for effective malware detection and prevention on mobile phones.

Keywords-Mobile Malware; Malware Detection; android malware;

I. INTRODUCTION

This Malware can also be termed as all kind of intrusions that is disastrous to the computer software and hardware system. Malware writer creates malware for different reasons and purposes ranging from challenges to economic gain, destruction to retaliation among others. Its growth is highly alarming in volume and its rate of expansion cannot be overlooked due to its damages[1]. Once malware gets itself into the system by different media like copying of files from external devices onto the system and mostly by downloading files from the internet, it checks the vulnerabilities of the system and infects the system if the system is highly vulnerable. The concern for the rate of spread of malware today is a global phenomenon, especially as it spreading double over the internet which is a means of global communication. Today's malware is capable of doing many things, such as: stealing and transmitting the contact list and other data, locking the device completely, giving remote access to criminals, sending SMS and MMS messages etc. Mobile malware causes serious public concern as the population of mobile phones is much larger than the population of PCs[2]. From year 2004 to 2008, the number of types of mobile malware has increased significantly. As of March 2008, FSecure has counted 401 different types of mobile malware in the world, and McAfee has counted 457 kinds of mobile malware [21]. Kaspersky Lab found that 99% of newly discovered mobile malicious programs target the Android platform, with a very small amount targeting Java- and Symbian-based smartphones. Kaspersky tracked a negligible eight new unique malicious programs in January 2011, after which the average monthly discovery rate for new Android malware in 2011 went up to more than 800 samples. In 2012, Kaspersky identified an average of 6300 new mobile malware samples every month. Overall, in 2012 the number of known malicious samples for Android increased more than eight times. Mobile malware has caused various harm such as leaking of user privacy, extra service charges by automatically sending expensive multimedia messages or making long-distance calls, and depletion of battery power. Even at least 15 variants of Cabir may be found spreading in over 35 countries [22], and 0.5~1.5% of MMS traffic in a Russian mobile network is made up of infected message, which is close to the fraction of malicious code in the email traffic [23]. At present, in response to this increasing threat, security

vendors such as avast, Quick Heal, Avg, McAfee, and Symantec have released mobile anti-virus, firewall, and encryption products. messages or data on handsets must be scanned for both mobile as well as regular malware—this will require searching against a *very* large database of known signatures. Due to limited CPU power, storage and memory, installing large signature databases is not an option for mobile devices. Therefore, there is a tremendous need for detecting malicious agents on handsets using alternative means. In this paper, we propose some malware detection and prevention techniques which will help to improve the current malware detection technique.

II. INCENTIVES FOR WRITING MOBILE MALWARE:-

Following are the incentives for writing mobile malware, they are as follows.

2.1 Novelty and Amusement:- Some malware causes mischief or damage in a way that appears to be intended to amuse the author. For example, Ikee. Changed the wallpaper of infected iPhone devices, and sent anti-religion text messages from Android phones. Many pieces of malware fall into this category and no other [14].

2.2 Selling User Information:- Mobile operating system APIs provide applications with large amounts of information about users. Applications can query mobile APIs for the user's location, list of contacts, browser and download history, list of installed applications, and IMEI (the unique device identifier). Although we cannot know for sure why malware collects this information, we hypothesize that this data is being sold by malware distributors for financial gain. Advertising or marketing companies might be willing to purchase users' locations, browsing histories, and lists of installed applications to improve behavioural profiling and product targeting. However, advertising libraries in legitimate applications already routinely collect user location, and web-based advertisements already track browsing habits.

2.3 Stealing User Credentials:- Credentials could be used directly by malware authors for greater financial gain, but financial fraud can be difficult to perpetrate and requires specialization. People use smartphones for shopping, banking, e-mail, and other activities that require passwords and payment information. Banks rely on cell phones for two-factor authentication. Users may also save authentication and payment credentials in text documents on their phones (for example, to use the phone as a mobile password manager). This makes cell phones a target for credential theft. Three pieces of malware in our data set target user credentials by intercepting SMS messages to capture bank account credentials[14].

2.4 Premium-Rate Calls and SMS:- Legitimate premium-rate phone calls and SMS messages deliver valuable content, such as stock quotes, technical support, or adult services. The cost of a premium-rate call or SMS is charged to the sender's phone bill. Premium rate calls can cost several dollars per minute, and premium-rate SMS messages can cost several dollars per message. In Android and Symbian, malware can completely hide premium-rate SMS messages from the user. Premium-rate SMS attacks could feasibly go unnoticed until the user's next phone bill.[14]

2.5 SMS Spam:- SMS spam is used for commercial advertising and spreading phishing links. Commercial spammers are incentivized to use malware to send SMS spam because sending SMS spam is illegal in most countries. Sending spam from a compromised machine reduces the risk to the spammer because it obscures the provenance of the spam. Furthermore, the use of SMS may lend more authenticity to spam than e-mail because phone contacts are often more intimately acquainted than e-mail contacts. 8 of the malicious Symbian and Android applications send SMS spam.[14]

2.6 Search Engine Optimization:- Many web sites rely on search engines for traffic, which makes web site owners desire high visibility in search engine results. Search engines rank web sites according to how relevant each web site is to a given search term. An engine's perception of relevance is inuenced by the rate at which users click on the web sites returned for a search term. A web site will rise in the results for a search term if many people search for that term and then click on that web site. Malware can be employed to improve a web site's ranking in search engine results. This type of malware sends web requests to the search engine for the target search term. The malware then fraudulently clicks" on the search result that corresponds to the target web site. As a result, the web site's rank for that search term will increase [14].

2.7 Ransom: Malware can be a tool for blackmail. For example, the desktop Trojan Kenzero stole the user's browser history, published it publicly on the Internet alongside the person's name, and then demanded 1500 yen to take down the person's browser history. There has not yet been any mobile malware that seriously threatens or publicly embarrasses the user for profit, but one piece of mobile malware has sought a ransom. A Dutch worm locked iPhone screens and demanded 5 euros to unlock the screens of infected phones [14].

III. Malware Attack Techniques:-

The infection strategies of malware include entry point obfuscation, code integration, code insertion, register renaming, memory access reordering and session hijacking. In entry point obfuscation, the virus hijacks the control of the program after the program has been launched, overwrite program import table addresses and function call instructions. During the code integration, a virus merges its code with legitimate program that

requires disassembly of target which is a very difficult operation (W95/Zmist). Malware can also either append virus code and thereby modify the entry point of a legitimate program or inject its code into unused sections of a program code. On the other hand, malware has two basic strategies adopted on a cell phone viz;

1) By creating a new process to launch its attack

2) By redirecting the program flow of a legitimate application in order to execute its malicious code within a legitimate security context (e.g messaging process) [24].

3.1 Malware First Attack Technique on Mobile Phone:- Malware, in this case created a new process to execute its malicious code and compromise the cell phone. This is a case where user operations are required, for example when a user downloads software on an internet or opens a received message from another user. The newly created process contains a program descriptor, which describes the address content, execution state and security context, which is different from that of the invoked parent process. This technique is widely adopted by the most existing malware one to its simplicity. In this technique, the cell phone malware launch an attack through legally installed application, having realized that the Symbian and windows programs register themselves within a platform and use their system services within their API framework. A good example is a cardblock Trojan, which is a cracked version of a legitimate Symbian application called instansis. It allows a user to create SIS archive. Cardblock blocks the MMC memory card and detect the subdirectories under system (SDI attack)[1].

3.2 Malware Second Attack Technique on Mobile Phone:- Malware, in this case redirects the program flow of a legitimate application (e.g. messaging activities) to execute its malicious code within a legitimate security context [24]. Open Source based OS and application a framework is the major target of this kind of malware attack. i.e Android smart phones. This type of attack is possible for malware by exploiting the stack buffer overflow in a Linux-based cell phone to "hijack "the normal program flow and launch its attacks.[1]

3.3 Malware method of Propagation:- The basic method of propagation of malware is either self-propagation or user interaction. A malware like worm does not require any user intervention before its execution occur. It is capable of copying itself and causing occasional execution without the intervention of host program or its user. Virus on the other hand is a user-interaction oriented malware that always looks for a host program for its execution and consequent infection. Other malware might not require any of these methods for its propagation, but may adopt internet medium for their spreading. Mobile malware on the other hand, adopt mobile phone network on the internet in order to propagate itself, but this action is usually curtailed by the internally built defence mechanism in the network mobile phone. Another opportunity for mobile malware to propagate is through the direct pair-wise communication resources i.e. Bluetooth, Wi-Fi, and Infrared.

IV. Malware Detection Techniques

The task of detecting malware can be categorized into analysis, classification, detection and eventual containment of malware. Several classification techniques have been used in order to classify malware according to their instances and this has made it possible to recognize the type and activities of a malware and new variant. Analysis of malware has to do with identifying the instances of malware by different classification schemes using the attributes of known malware characteristics. Malware detection has to do with the quick detection and validation of any instance of malware in order to prevent further damage to the system. The last part of the job is containment of the malware, which involves effort at stopping escalation and preventing further damages to the system. A commercial antivirus uses signature based technique where the database must be regularly updated in order to possess the latest virus data detection mechanisms. However, the zero-day malicious exploit malware cannot be detected by antivirus, based on signature-based scanner, but the use of statistical binary content analysis of file to detect anomalous file segments [11]. Toward this end, malware detection technique has been classified according to the following:

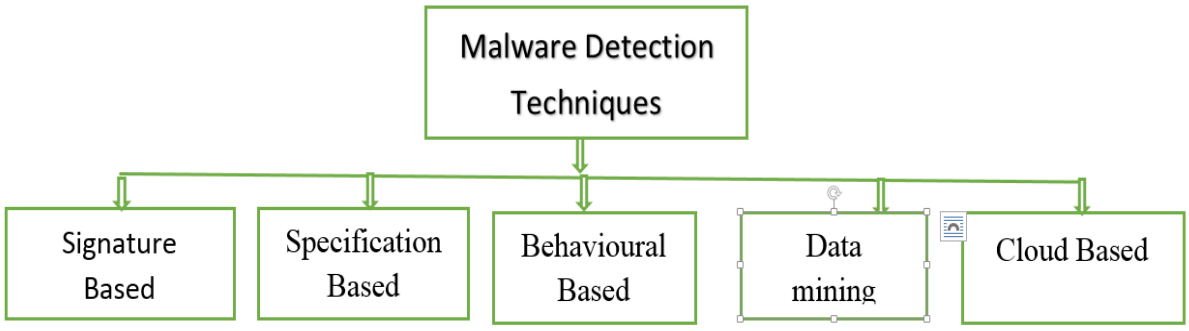


Fig. Malware Detection Techniques

4.1 Signature-based malware Detection:-

A pattern-marching approach commercial antivirus is an example of signature based malware detection where the scanner scans for a sequence of byte within a program code to identify and report a malicious code. This approach to malware detection adopts a syntactic level of code instructions in order to detect malware by analysing the code during program compilation. This technique usually covers complete program code and within a short period of time. However, this method has limitation by ignoring the semantics of instructions, which allows malware obfuscation during the program's run-time.

In smartphone operating systems, the behaviour of malware may occur in multiple locations, the occurrence of these acts combined according to certain timing in order to constitute malicious software behaviour, one or a few of these separate behaviour can't determine whether they are malicious behaviour's or not. This collection is then processed by temporal relations, after all the behaviours are abstracted and signed to software behaviour patterns. Code packing, simple scrambling does not change the behaviour of software, malware and its variants are generally in the same run-time behaviour patterns, the signature of these malware can be detected through the same behaviour. Compared with feather-based malware detection method, the signature database of behaviour signature based is becoming smaller, so the behaviour based detection of malicious software signature is ideal for resource-constrained mobile devices. New malicious software usually include new behaviour signature that is inconsistent with the previous known normal behaviour, so behaviour-based malware detection signatures can detect new and unknown malware.

a) True Negative: The file being processed is a clean file and no match is found. A true negative is generated in this case.

b) False positive:- A false positive occurs when a virus scanner erroneously detects a 'virus' in a non-infected file. False positives result when the signature used to detect a particular virus is not unique to the virus - i.e. the same signature appears in legitimate, non-infected software.

c) False negative:- A false negative occurs when a virus scanner fails to detect a virus in an infected file. The antivirus scanner may fail to detect the virus because the virus is new and no signature is yet available, or it may fail to detect because of configuration settings or even faulty signatures.

d) True Positive: The file being processed is a malware file and a match is found. A true positive is generated in this case.

Condition a and d do not trigger any update to the malware database. The update is required for condition b and c. In the case of a false positive, the subject file is a clean file, therefore the matching signature has to be removed from the malware signature database. Similarly in the false negative case, the signature has to be added to the signature database, as the subject file is a malware and the signature database do not carry a matching signature.

4.2 Specification-based malware detection:- Specification based detection makes use of certain rule set of what is considered as normal in order to decide the maliciousness of the program violating the predefined rule set. Thus programs violating the rule set are considered as malicious program. In specification-based malware detection, where a detection algorithm that addresses the deficiency of pattern-matching was developed. This algorithm incorporates instruction semantics to detect malware instances. The approach is highly resilience to common obfuscation techniques. It used template T to describe the malicious behaviours of a malware, which are sequence of instructions represented by variables and symbolic constants. The limitation of this approach is that the attribute of a program cannot be accurately specified. Specification-based detection is the derivate of anomaly based detection. Instead of approximating the implementation of a system or application, specification-based detection approximates the requirements of application or system. In specification-based system there exists a training phase which attempts to learn the all valid behaviour of a program or system which needs to inspect. The main limitation of specification based system is that it if very difficult to accurately specify the behaviour the system or program. One such tool is Panorama which captures the system wide information flow of the program under inspection over a system, and checks the behaviour against a valid set of rule to detect malicious activity.

4.3 Behavioural-based Detection:- Behaviour-based malware analysis and detection techniques was proposed by Forrest et a[25]l. This approach does not only perform surface scanning but also identify the malware's action.The approach generates database of a malicious behaviours by studying a distinct number of families of malware on a target operating system. [25] Develops a two stage mapping technique that constructs signatures at run-time from the monitored system event and API calls. The system trains a classifier using a support vector machines (SVMs) to distinguish a malicious program from normal application behaviours. This detection system is capable of detecting metamorphic malware which keep reproducing. The implementation of malware detection systems in mobile devices is a relatively new concept. Security tools and mechanisms used in computers are not feasible for applying on smartphones due to the excessive resource consumption and battery depletion. In behavioral-based detection techniques, the behavior of an application is monitored and compared

against a set of malicious and/or normal behavior profiles. The malicious behavior profiles can be specified as global rules that are applied to all applications, as well as fine-grained application-specific rules. Behavioural detection is more flexible to deal with polymorphic worms or code obfuscation, because it assesses the effects of an application based on more than just specific payload signatures. Considering the fact that a new malware variant is usually created by adding new functionality to existing malware or modifying obsolete modules with fresh ones, this abstraction is effective for detecting previously-unknown malware variants that share a common behaviour exhibited by previously-known malware. A typical database of behaviour profiles and rules should be smaller than that needed for storing specific payload signatures of many different classes of malware. This makes behavioural detection methods particularly suitable for handsets.[1]

One common problem with behavioural detection, however, is specification of what constitutes normal or malicious behaviour that covers a wide range of applications [4], while keeping false positives (incorrect identification of a benign activity as malicious) and false-negatives (failure to identify malicious activities) low. Another one is the reconstruction method of potentially suspicious behaviour from the applications, so that the observed signatures can be matched against a database of normal and malicious signatures. Heuristics on what behavioural should be monitored and how to monitor and collect behavioural vary. In the following, some of researchers' efforts to produce methodologies for malware detection and relevant behavioural measurements are conclude. Behaviour based detection differs from the surface scanning method in that it identifies the action performed malware rather than the binary pattern. The programs with dissimilar syntax's but having same behaviour are collected, thus this single behaviour signature can identify various samples of malware. This types of detection mechanisms helps in detecting the malwares which keeps on generating new mutants since they will always use the system resources and services in the similar manner.

4.4 Data Mining Technique of Detecting Malware:- In data mining methods for detecting malicious executables, a malicious executable as a program that performs function, such as compromising a system's security, damaging a system or obtaining sensitive information without the user's permission. Their data mining methods detect patterns in large amounts of data, such as byte code, and use these patterns to detect future instances in similar data. Their framework used classifiers to detect new malicious executables. A classifier is a rule set, or detection model, generated by the data mining algorithm that was trained over a given set of training data. They designed a framework that used data mining algorithms to train multiple classifiers on a set of malicious and benign executables to detect new examples. The binaries were first statically analysed to extract properties of the binary, and then the classifiers trained over a subset of the data. Their large sets of programs from public sources were separated into two classes: malicious and benign executables. Example of this data set is a Windows or MS-DOS format executable, which is also applicable to other formats. Since the virus scanner was updated and the viruses were obtained from public sources, it was assumed that the virus scanner has a signature for each malicious virus. They then split the dataset into two subsets: the training set and the test set. The data mining algorithms used the training set while generating the rule sets. The test set was then used to check the accuracy of the classifiers over unseen examples. This data mining method was able to detect previously undetectable malicious executables by comparing the results with traditional signature-based methods and with other learning algorithms. The Multi-Naive Bayes method had the highest accuracy and detection rate of any algorithm over unknown programs, 97.76%, over double the detection rates of signature-based methods. Its rule set was also more difficult to defeat than other methods because all lines of machine instructions would have to be changed to avoid detection [1].

4.5 Cloud Based Malware Detection:- Google Play applications are scanned for malware: Google uses a service named Bouncer to automatically scan applications on the Google Play Store for malware. As soon as an application is uploaded, Bouncer checks it and compares it to other known malware, Trojans, and spyware. Every application is run in a simulated environment to see if it will behave maliciously on an actual device. The applications behavior is compared to the behavior of previous malicious apps to look for red flags. New developer accounts are particularly scrutinized – this is to prevent repeat offenders from creating new accounts. Google Play can remotely uninstall applications: If you've installed an app that is later found to be malicious, Google has the ability to remotely uninstall this application from your phone when it's pulled from Google Play.Google announced an exciting security feature called the "application verification service" to protect against harmful Android applications. As stated in a recent Google+ post by a member of the Google Android team, "Now, with Jelly Bean Android 4.2 devices that have Google Play installed have the option of using Google as an application verifier. We will check for potentially harmful applications no matter where you are installing them from". Google to directly face Android malware threats and take such measures to better protect Android users. When you install the Android 4.2 update on your system, you'll be greeted with a pop-up notification asking you if you want to verify all installed apps. If you ignore that popup, you can also enable app verification from the Settings > Security menu.

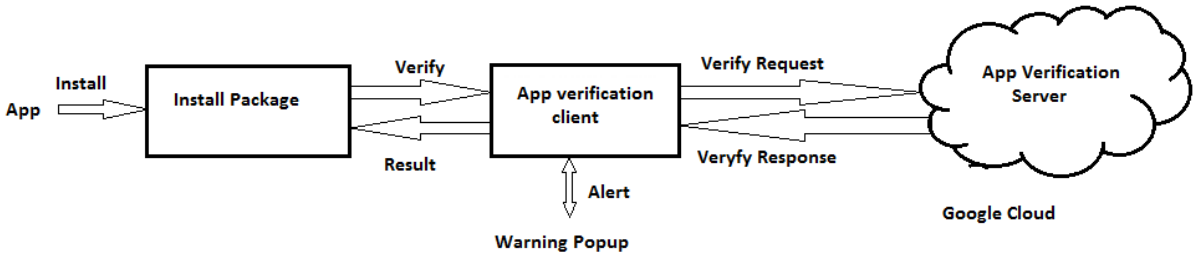


Figure. Cloud Based Detection

A user can turn the service on/off by going to "Settings," "Security," and then "Verify apps." When an app is being installed (Step 1), the service, if turned on, will be invoked (Step 2) to collect and send information about the app (e.g., the app name, size, SHA1 value, version, and the URL associated with it) as well as information about the device (e.g., the device ID and IP address) back to the Google cloud (Step 3). After that, the Google cloud will respond with a detection result (Step 4). If the app is not safe, the user is then shown a warning popup (Step 5) flagging the app as either dangerous or potentially dangerous. Dangerous apps are blocked from being installed, while potentially dangerous ones instead alert users and provide an option to either continue or abort the installation (Step 6) with a warning popup.

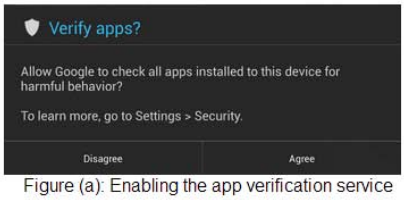


Figure (a): Enabling the app verification service

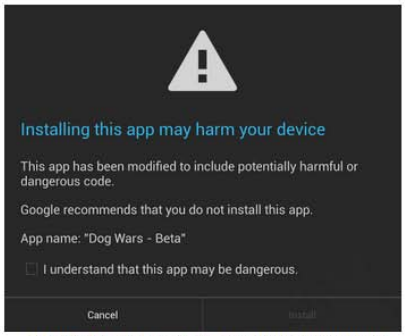
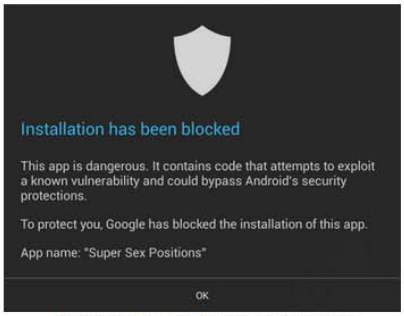


Figure (b): Detecting a *potentially dangerous* app



Figure(c): Detecting a *dangerous* app

Android 4.2 scans side loaded apps: While apps on Google Play are checked for malware, apps that are side loaded (installed from elsewhere) were not checked for malware. On Android 4.2, when you first try to sideload an app, you'll be asked whether you want to verify side loaded apps are safe. This ensures that all apps on your device are checked for malware.

V. Conclusion

Mobile handsets are glued victim to malwares due to their flexible communication and computation capabilities, and their resource constraints. To provide well-rounded protection, a security suite for mobile devices or smartphones especially open-source ones such as Android should include a collection of tools blending various capabilities that operate in synergic fashion. The presented detection techniques are viable, but large scale testing is required to determine real world performance. As Android malware evolves the effectiveness of these measures will decrease. However these techniques can still be valuable as they raise the bar of entry for repackaged and newly created malware and come at low overhead.

REFERENCES

[1] Adebayo, Olawale Surajudeen, Mabayoje, Amit Mishra, Osho Oluwafemi, "Malware Detection, Supportive Software Agents and Its Classification Schemes", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.6, November 2012.
[2] Marwa M. A. Elfattah, Aliaa A.A Youssif, Ebada Sarhan Ahmed," Handsets Malware Threats and Facing Techniques", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, 2011.
[3] Gianluca Dini, Fabio Martinelli, Andrea Saracino, and Daniele Sgandurra," MADAM: a Multi-Level Anomaly Detector for Android Malware".
[4] Hsiu-Sen Chiang, Woei-Jiunn Tsaur," Mobile Malware Behavioral Analysis and Preventive Strategy Using Ontology", IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust 978-0-7695-4211-9/10 $26.00 © 2010 IEEE DOI 10.1109/SocialCom.2010.160 1080.
[5] Asaf Shabtai," Malware Detection on Mobile Devices", Eleventh International Conference on Mobile Data Management", 978-0-7695-4048-1/10 $26.00 © 2010 IEEE DOI 10.1109/MDM.2010.28.
[6] Abhijit Bose," Propagation, Detection and Containment of Mobile Malware".
[7] Aubrey-Derrick Schmidt,Rainer Bye, Hans-Gunther Schmidt, Jan Clausen, Osman Kiraz, Kamer Ali Yuksely, Seyit Ahmet Camtepe, and Sahin Albayrak, "Static Analysis of Executables for Collaborative Malware Detection on Android". IEEE international conference on Communications, June 2009.
[8] Danny Iland, Alexander Pucher, Timm Schauble," Detecting Android Malware on Network Level", December 5, 2011.
[9] AV-TEST GmbH "Anti-Malware solutions for Android",March, 15th 2012.

[10] Ashkan Sharifi Shamili, Christian Bauckhage, Tansu Alpcan," Malware Detection on Mobile Devices using Distributed Machine Learning", 2010 International Conference on Pattern Recognition 2010 IEEE DOI 10.1109/ICPR.2010.1057.
[11] Markus Jakobsson, Karl-Anders Johansson," Retroactive Detection of Malware With Applications to Mobile Platforms", HotSec'10 Proceedings of the 5th, USENIX conference on Hot topics in security, Article No. 1-13, USENIX Association Berkeley, CA, USA ©2010.
[12] Jeffrey Bickford, H. Andrés Lagar-Cavilla, Alexander Varshavsky, Vinod Ganapathy, Liviu Iftode, "Security versus energy tradeoffs in host-based mobile malware detection", 9th international conference on Mobile systems, applications, and services, June 2011.
[13] Ken Dunham, "Mobile Malware Attacks and Defense", Syngress Publishing, October 2008.
[14] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steven Hanna, and David Wagner, "A Survey of Mobile Malware in the Wild", 1st ACM workshop on Security and privacy in smartphones and mobile devices, October 2011.
[15] Jeffrey Bickford , H. Andrés Lagar-Cavilla , Alexander Varshavsky , Vinod Ganapathy , and Liviu Iftode," Security versus Energy Tradeoffs in Host-Based Mobile Malware Detection", MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, Pages 225-238 ACM New York, NY, USA ©2011.
[16] Yajin Zhou, Xuxian Jiang, "Dissecting Android Malware: Characterization and Evolution", IEEE Symposium on Security and Privacy, 12 september 2012.
[17] Yacin Nadji, Jonathon Giffin, and Patrick Traynor, "Automated Remote Repair for Mobile Malware", 27th Annual Computer Security Applications Conference, December 2011.
[18] Abhijit Bose, Kang G. Shin, "On capturing malware dynamics in mobile power-law networks", 4th international conference on Security and privacy in communication netowrks, September 2008.
[19] Thomas Blasing, Leonid Batyuk, Aubrey-Derrick Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak, "An Android Application Sandbox System for Suspicious Software Detection".
[20] Yean Li Ho, Swee-Huay Heng, "Mobile and Ubiquitous Malware", 7th International Conference on Advances in Mobile Computing and Multimedia,ACM, December 2009.
[21] G. Lawton, "Is it finally time to worry about mobile malware," Computer, vol. 41, no. 5, pp. 12-14, 2008.
[22] Shane Coursen, "The future of mobile malware," Network Security, vol. 2007, issue 8, pp. 7-11, 2007.
[23] Yury, "Mobile threats – myth or reality," Kaspersky Lab's Report on Mobile Viruses. Nov. 2006. Available online at http://www.viruslist.com/en/weblog?weblogid=204924390 , 2010.
[24] Abhijit, B., Xin, H., Kang G. S. and Taejoon, P. (2008) " Behavioral detection of Malware on Mobile Handsets", June 17–20, 2008, Breckenridge, Colorado, USA. ACM 978-1- 60558-139-2/08/06.
[25] Stephanie, F., Steven, A., Hofmeyr, A. S. and Thomas, A. L. (1996) "A sense of self for Unix Processes", In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pages 120–128. IEEE Computer Society Press.
[26] http://developer.android.com/about/versions/jelly-bean.html