International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

A Review Paper on Effective Behavioral Based Malware Detection and Prevention Techniques for Android Platform

Mr. Sagar Vitthal Shinde[1]
M.Tech Comp. Department of Technology,
Shivaji University, Kolhapur,
Maharashtra, India.
Email id: sagarshinde440@gmail.com

Ms. Amrita A. Manjrekar[2]
Assistant Professor, Department of Technology,
Shivaji University, Kolhapur,
Maharashtra, India.
Email Id: aam_tech@unishivaji.ac.in

Abstract

Android is most popular platform for mobile devices. Smartphone's and mobile tablets are rapidly indispensable in daily life. Android has been the most popular open sources mobile operating system. On the one side android users are increasing, but other side malicious activity also simultaneously increasing. The risk of malware (Malicious apps) is sharply increasing in Android platform, Android mobile malware detection and prevention has become an important research topic. Some malware attacks can make the phone partially or fully unusable, cause unwanted SMS/MMS (short message service/multimedia messaging service) billing, money, or expose private information.

Various applications contain misbehavior code, but those are not actually malicious apps. The existing system categories such apps as a malware apps. This problem will be overcome in proposed system with the help of new feature extraction algorithm. In proposed system selected android features will be extracted for entire feature set to detect malware on four phases: package, user, application, and validation phase. The malware detection will be based on behavioral and classified according to their risk (High, Medium, and Low). This will be helpful for the user to handle the system (Application) very smoothly.

Keywords: *Android, malware detection and prevention, feature extraction, behavioral based*

INTRODUCTION

Android is Linux based open sources operation system developed by Google. Now a day it is very popular due to its several features. As increasing in number of Android phones there is simultaneous increase in mobile malware apps that performs malicious activities like misusing user's private information as sending messages. Malware is a type of malicious software which interrupt the different operation on mobile system, crashes the important information or private information of the mobile system. In other words malicious software, malware refers to software programs designed to damage or do other unwanted actions on a mobile system.

Furthermore, all these misbehaviors can be performed on Android devices without the user notice (or when it is too late). It has been recently reported that almost 60% of existing malware send stealthy premium rate SMS messages. Most of these behaviors are exhibited by a category of apps called Trojanized that can be found in online marketplaces not controlled by Google. However, also Google Play, the official market for Android apps, has hosted apps which have been found to be malicious [1] [21].

Existing system consist of some limited features of android app, malware detection is based on behavioral base. The malware detection and prevention process is also static which create some problems such as it increase false positive rate. Malicious apps (generically called malware) constitute the main vector for security attacks against mobile devices. Disguised as traditional and helpful apps, they hide treacherous code that performs actions within the background that threatens the user privacy, the device integrity, or maybe user's credit. Some common examples of attacks performed by Android malicious apps are stealing contacts, login credentials, text messages, or maliciously subscribing the user to expensive premium services.

Today's world is mobile (Smartphone) world. Due to low prices of Smartphone's are availability in 3G and 4G networks. Smartphone's and tablets have become extremely popular in the last years. At the end of 2014, the number of active mobile devices worldwide was almost 7 billion, and in developed nations the ratio between mobile devices and people is estimated as 120.8 % [22]. Most of these many are common peoples who don't know what is the android structure and how android system work. Due to well interface and openness android is very popular day by day. There are much more application available which increase performances of system as well as reduce time and cost.

The user is unaware of which application is good and which are harmful. Many apps are stealing user personal data. More than 1 millions of malicious apps are currently available in the world [23]. Many apps looks like as normal and useful apps, but they hide treacherous code which performs actions in the background that threatens the user privacy, the device integrity, or even user's credit.

All private companies and government organizations moved their work to android application. The chances of information leakage and theft of personal data is increased. In existing system focus on limited features of android application to

detect malware. Therefore here system is proposed to better malware detection in android Devices.

LITERATURE SURVEY

There are various methods available for android malware detection classification & prevention in literature it has been observed that mainly three approaches were considered which are as follows:

- **Signature-based detection:** may be a widespread technique supported looking for antecedently outlined virus signatures in input files [23]. Signature detection has the advantage of detecting malicious activity before the system is infected by the malicious code.
- **Behavior checking:** is another standard technique supported a behavior checker that resides within the memory longing for uncommon behavior. [23] During this case, the user is alerted. Behavior checker encompasses a disadvantage that by the time a malicious activity is detected, some changes have already been done to the system.
- **Integrity Checker**: is the technique that maintains a log of all the files that area unit gift within the system. The log could contain characteristics of files just like the file size, date/time stamp and substantiation. Whenever associate degree integrity checker is run, it'll check the files on the system and compares with the characteristics it had saved earlier. [23]

Depending upon types of malware detection method/technique The existing approaches are classified and listed below

Host based malware detection

Andrea Saracino at. [1] Presented MADAM, a novel host-based malware detection system for Android devices this simultaneously analyzes and correlates features at four levels: kernel, application, user and package, to detect and stop malicious behaviors. MADAM has been designed to take into account those behaviors characteristics of almost every real malware which can be found in the wild. MADAM detects and effectively blocks more than 96% of malicious apps, which come from three large datasets with about 2,800 apps, by exploiting the cooperation of parallel classifiers and a behavioral based detector.

Inter-App permission Leakage

Hamid Bagheri at [2], presents COVERT, a tool for compositional analysis of Android inter-app vulnerabilities. COVERT's analysis is modular to enable incremental analysis of applications as they are installed, updated, and removed. It statically analyzes the reverse engineered source code of each individual app, and extracts relevant security specifications in a format suitable for formal verification. Given a collection of specifications extracted in this way, a formal analysis engine (e.g., model checker) is then used to verify whether it is safe for a combination of applications holding certain permissions and potentially interacting with each other to be installed together.

ICC Detector trained model for malware detection

Author Proposed [4] a new malware detection method, named ICC Detector. ICC Detector outputs a detection model after training with a set of benign apps and a set of malwares, and employs the trained model for malware detection. The performance of ICC Detector is evaluated with 5264 malwares, and 12 026 benign apps. Compared with their benchmark, which is a permission-based method proposed by Peng et al. in 2012 with an accuracy up to 88.2%,

ALTERDROID, a dynamic analysis approach for detecting such hidden or obfuscated malware

In this paper [5] they had describe ALTERDROID, a dynamic analysis approach for detecting such hidden or obfuscated malware components distributed as parts of an app package. The key idea in ALTERDROID consists of analyzing the behavioral differences between the original app and a number of automatically generated versions of it, where a number of modifications (faults) have been carefully injected. Observable differences in terms of activities that appear or vanish in the modified app are recorded.

Exchanging data using two detection methods

The Author have aim [6] to spot malware covertly exchanging data using two detection methods based on artificial intelligence tools, such as neural networks and decision trees. To verify their effectiveness, seven covert channels have been implemented and tested over a measurement framework using Android devices. Experimental results show the feasibility and effectiveness to detect the hidden data exchange between colluding applications.

Privacy preserving data-leak detection

In this technique author presented [10] a privacy preserving data-leak detection (DLD) solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of our method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider. They describe how Internet service providers can offer their customers DLD as an add-on service with strong privacy guarantees.

Permission - induced risk in Android apps

In this paper [12], they explore the permission-induced risk in Android apps on three levels in a systematic manner. First, they thoroughly analyze the risk of an individual permission and the risk of a group of collaborative permissions. They employ three feature ranking methods, namely, mutual information, correlation coefficient, and T-test to rank Android individual permissions with respect to their risk. We then use sequential forward selection as well as principal component analysis to identify risky permission subsets.

VetDroid, a dynamic analysis platform

In This paper [13] they have presented VetDroid, a dynamic analysis platform for generally analyzing sensitive behaviors in Android apps from a novel permission use perspective. VetDroid proposes a systematic permission use analysis technique to effectively construct permission use behaviors,

i.e., how applications use permissions to access (sensitive) system resources, and how these acquired permission-sensitive resources are further utilized by the application. With permission use behaviors, security analysts can easily examine the internal sensitive behaviors of an app.

Solution that leverages a method to assign a risk score to each app

Christopher S. Gates [15] had proposed a solution that leverages a method to assign a risk score to each app and display a summary of that information to users. Results from four experiments are reported in which they examine the effects of introducing summary risk information and how best it is.

ANDROID MALWARE DETECTION OVERVIEW

Malware outbreaks in wireless networks represent associate rising analysis topic [24]. In November 2006 web poll of company IT directors by security seller's sophos reported that eighty one of respondent specific concern over malware and spyware targeting mobile devices can become a big threat. However, sixty fourth aforementioned they need nothing in situ to secure their sensible phones and PDAs [25]. Because the range of mobile devices within the world has swollen dramatically in recent years, the number of malware targeting the mobile devices conjointly accumulated [25]. This is brief examined the work by Dagon et.al [26], to grasp the kinds of attacks against mobile devices on the idea of securities attackers hope to attain. Table I depicts the classification established by the authors

The detailed description of security attacks is mentioned below

Security Goals	Types of Attacks
Confidentiality	Theft of data, bluebugging and bluesnarfing
Integrity	Phone hijacking
Availability	Denial-of-service attacks and battery draining

Table 1. Security Attacks

- **Theft of data:** Hackers usually attack mobile devices to get transient info and static info. Transient info includes the phones location, its power usage and alternative information, that the device doesn't usually record [26].They attack on static info that cellular devices store or send over the network. These attacks attempt to get information like contact info, phone numbers, and programs keep on sensible phones. The blue snarfing and blue bugging attack area unit samples of information larceny. the blue bugging attack permits unauthorized access to the phone and will embrace paying attention to calls made of and to a victim's phone.

- **Phone Hijacking:** Some malware may conceive to use the victim's phone resources. Prospects embrace putting long-distance or 900-number calls, causing valuable SMS messages, etc. The recent Mosquitoes virus is one example [26]. Pirated copies of a game were infected with an outbreak that sent valuable SMS messages once users complete the illicit copy of the sport. Hijacking phone resources isn't sudden – malware authors are mistreatment victims' resources for quite a where as.

- **Denial-of-Service (DoS):** consistent with Dagon et al. [26], DoS may well be done by flooding the device and exhausting power. At present, it's extraordinarily simple to crash or overwhelm most Bluetooth applications on mobile devices simply by causation perennial items of data, corrupted packets, and incorrect file formats. However, power demands forever constrain mobile devices , therefore this latter class is believed to be a lot of serious. DoS continues to be the dominant attack kind that may be exploited from the celebrated vulnerabilities [24]. Trying to the attack history, several Trojans, Worms, Viruses have entered the mobile world and have affected them. Consistent with F-secure, there have been quite 350 mobile malware in circulation by the tip of 2007 [9][29] .

Detection Methods of Android Malware

Presently, there are two Methods to detect Android Malware: static behavioral detection method and dynamic behavioral detection method.

Static Behavioral Detection Method:

Through analyzing and examination the instruction codes of the software package, Static activity detection methodology detects whether or not the software package contains API operate calls which may cause malicious behaviors. mistreatment this methodology to discover, it first of all acquires Java supply codes of humanoid application software package, analyzes whether or not the software package contains sensitive operate calls and whether or not there square measure security threats, so comes up with the conclusion whether or not the software package is malicious or not. Static activity detection methodology has to decompile the appliance by ways that of reverse engineering to accumulate supply codes. However the analysis is usually full of software package secret writing and implicit functions (virtual functions, etc.,) thus it always cannot draw the proper conclusion.

Dynamic behavioral Detection Method:

Dynamic behavioral detection methodology works throughout the running of the appliance. It detects and records the system's communications, short messages, network interfaces and therefore the network access of the relevant implicit data, therefore exploit the application's behavior model. Dynamic behavioral detection methodology will solve the issues that static detection methodology cannot do as a result of the appliance codes square measure encrypted or confused. Dynamic behavioral detection methodology constructs operation setting by victimization sandbox, virtual machine and alternative forms, and simulates the execution of the appliance to amass the application's behavior model. it's higher request to the period detection

Acquiring the Malicious Behaviors

Firstly, the author collects fifty malicious computer code samples as well as worm, spyware and worms, etc., decompiles them, and analyzes the operate calls of their APK supply codes. Within the method of decompilation, the author uses DEX2JAR to rework categories.dex file to Java codes, so the reworked categories.dex file contains APK implementation codes, acquires its resources file and sophistication file, and uses Java Decompiler to rework the category file to clear format. The binary AndroidManifest.XML file is reworked by AXMLPrinter2 [30].

Behaviors	**The name of information Collector**
Receives SMS/MMS	SMS/MMS , IMEI
Sends SMS/MMS, Access Device location	Phone Number, Contacts
Send Data over HTTP(s), Execute Commands	Email, Android Version
Uses WiFi, Encryption	Browser History, GPS Coordinates
Write to disk (internal or external flash card)	Data in Flash Card , Call logs
Obfuscation, Internet Logs	Phone Conversation
Send Data (cellular)	Root Level
Receive Data over HTTP(s)	

Table 2. The Statistic of the Sample`s malicious behavior

COMPARISONS BETWEEN DETECTION TECHNIQUES

Sr.No	**Malware Detection Technique**	**Pros**	**Cons**
1	Anomaly-based detection	• Can detect potentially a wide range of novel attacks	• May miss known attacks • May miss novel attacks if they don't stick out along the observed dimension • High false positive rate • Purity of training data (i.e., absence of attacks)
2	Behavior-based detection	• May detect a wide range of novel attacks • Low false positives • Can be cheap to deploy and monitor	• Post-facto, attack already occurred • Easy to evade once known

PROPOSED WORK

The propose System will be consist of new feature extraction algorithm to extract appropriate feature of android application. The malware detection is based on behavioral. On successful implementation of system it will classify android apps into two type's i.e. malicious app or benign app (not malware or harmful). The system will also give risk rank (risk involved) of that particular app. This will help user to handle various application in his android devices.

The proposed system will be consisting of three modules the details are as follows:

- **Pre-App Monitor**

The pre-app monitor will be complementary to the monitor activity it is aimed to detecting additional misbehaviors (Patterns) of unknown application. The pre-app monitor module deals with

Structure of an android application:

Components are basic logical building blocks of Android applications. Each component can be run individually, either by its embodying application or by system upon permitted requests from other applications. Android applications can comprise four types of components: (1) **Activity components** provide the basis of the Android user interface. Each Application may have multiple Activities representing different screens of the application to the user. (2) **Service components** provide background processing capabilities, and do not provide any user interface. Playing music and downloading a file while a user interacts with another application are examples of operations that may run as a Service. (3) **Broadcast Receiver** components respond asynchronously to system-wide message broadcasts. A receiver component typically acts as a gateway to other components, and passes on messages to Activities or Services to handle them. (4) **Content Provider components provide** database capabilities to other components. Such databases will be used for both intra-app data persistence as well as sharing data across applications.

Extraction of features of android app [2]

Algorithm 1. Model Extractor

Input: app: Android App
Output: A: App's Extracted Model

```
1 A ← < {}; {}; {}; {}; {}>
2 ICFG ← {}
3 summaries ← {}
//►Entity Extraction cf. Sec. 5.1
4 A:C ← extractManifestComponents (app)
5 A:P ← extractManifestPermissions(app)
6 A:F ← extractManifestFilters(app)
7 IFEntities ← {}
8 foreach method Є app do
9 IFEntities ← identifyIFEntity(method; summaries)
10 end
11 resolveIFEntityAttr(IFEntities)
12 A:I ← getIntents(IFEntities)
13 A:F ← getIntentFilters(IFEntities) Ư A:F
//►ICFG Augmentation - cf. Sec. 5.2
14 G ← constructICFG(app)
15 E ← extractImplicitCallBacks(app)
16 ICFG ← augmentICFG(G;E)
//►Vul. Paths Identification - cf. Sec. 5.3
17 A:S← findVulPaths(A:C; ICFG)
```

Storing the extracted feature in feature set.

The output of the model extractor algorithm will be stored in feature set as well as the pattern or behavior of known application will be stored for future use.

➢ **Risk Analysis**

In the risk analysis module calculate the risk of android application. The risk Analysis module deals with

Classifying the application according to their behavior malicious app or benign app:

The classification will be done on malicious behavioral patterns. which are as following

- "Text messages sent by a non-default message app".
- "Text messages sent to numbers not in the user contact list".
- "High number of outgoing message per period of time".
- "High number of process per app".
- "Unauthorized installation of new apps".
- "Unsolicited kernel level activity of background app".

Calculating the risk of Malware affected app:

The risk (High Medium Low) will be calculated on the basic of malware classes. Which are as following

- "Botnet"
- "Root kit"
- "Sms Trojan"
- "Spyware"
- "Installer"
- "Ransomware" (prevent user from interacting with the device)

Giving information to user about malicious app.

The risk information of the application will be given to user at the run time. The notification of pop-up message will be occurred on user screen.

➢ **User Interface**

The User interface module will be responsible for prevention of malware.

Taking Decision about Application:

In this stage proper decision about application will be taken, either user want to keep the application or uninstall the application.

Perform the appropriate action (uninstall app or keep app):

In this stage include prevention module which stop malicious action and in case a malware is found handle the procedure for removing the application.

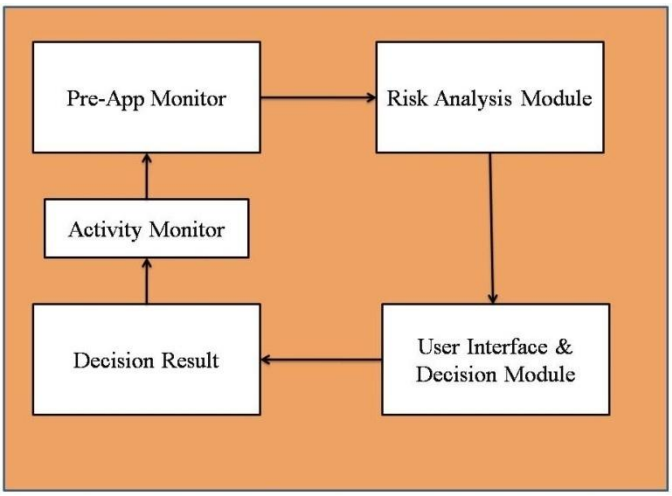

Figure 1. Proposed System Architecture

CONCLUSION

This paper reviews the existing malware detection and prevention techniques. Here existing malware detection & prevention approaches were discussed & overviewed malicious software are present in large numbers on android platform compare to others the harm of malicious behavior is increasing recently which brings greater challenges to detection & prevention of android malicious software. Here system is proposed with the help of new feature extraction algorithm for detection of android malware apps & risk analysis of it. Depending upon system result user will initiate appropriate action. This system will address android platform security problems will be useful in near future .

References

[01] Andrea Saracino, Daniele Sgandurra, Gianluca Dini and Fabio Martinelli, "MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention", IEEE Transactions on Dependable and Secure Computing , 2016.

[02] Hamid Bagheri, Member, IEEE, Alireza Sadeghi, Joshua Garcia, and Sam Malek, Member, IEEE, "COVERT: Compositional Analysis of Android Inter-App Permission Leakage" IEEE Transactiton on software engineering,2015

[03] Shancang Li, Theo Tryfonas, Gordon Russell, and Panagiotis Andriotis, "Risk Assessment for Mobile Systems Through a Multilayered Hierarchical Bayesian Network", IEEE TRANSACTIONS ON CYBERNETICS, VOL. 46, NO. 8, AUGUST 2016

[04] Ke Xu, Yingjiu Li, and Robert H. Deng "ICCDetector: ICC-Based Malware Detection on Android", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 6, JUNE 2016

[05] Guillermo Suarez-Tangil, Juan E. Tapiador, Flavio Lombardi, and Roberto Di Pietro, "ALTERDROID: Differential Fault Analysis of Obfuscated Smartphone Malware" IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 15, NO. 4, APRIL 2016.

[06] Luca Caviglione, Mauro Gaggero, Jean-François Lalande, Wojciech Mazurczyk, Senior Member, IEEE,and Marcin Urba´nski "Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 4, APRIL 2016

[07] Xiaokui Shu, Jing Zhang, Danfeng (Daphne) Yao, Senior Member, IEEE, and Wu-Chun Feng, Senior Member IEEE, "Fast Detection of Transformed Data Leaks", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 11, NO. 3, MARCH 2016

[08] Chunjie Zhou, Shuang Huang, Naixue Xiong, Senior Member, IEEE, Shuang-Hua Yang, Senior Member, IEEE,Huiyun Li, Yuanqing Qin, and Xuan Li, "Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 45, NO. 10, OCTOBER 2015

[09] Jemal Abawajy, Senior Member, IEEE, Morshed Chowdhury and Andrei Kelarev, "Hybrid Consensus Pruning of Ensemble Classifiers for Big Data Malware Detection", IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, OCTOBER 2015

[10] Xiaokui Shu, Danfeng Yao, *Member, IEEE*, and Elisa Bertino, *Fellow, IEEE* "Privacy-Preserving Detection of Sensitive Data Exposure" , IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 10, NO. 5, MAY 2015

[11] Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur,Mauro Conti, Senior Member, IEEE, and Muttukrishnan Rajarajan "Android Security: A Survey of Issues, Malware Penetration, and Defenses" IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 2, SECOND QUARTER 2015

[12] Wei Wang, Xing Wang, Dawei Feng, Jiqiang Liu, Zhen Han, and Xiangliang Zhang, *Member, IEEE*, "Exploring Permission-Induced Risk in Android Applications for Malicious application Detection" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 11, NOVEMBER 2014

[13] Yuan Zhang, Min Yang, Zhemin Yang, Guofei Gu, Peng Ning, and Binyu Zang, "Permission Use Analysis for Vetting Undesirable Behaviors in Android Apps" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 11, NOVEMBER 2014

[14] Silvio Cesare, Member, IEEE, Yang Xiang, Senior Member, IEEE , and Wanlei Zhou, Senior Member, IEEE, "Control Flow-Based Malware Variant Detection" IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 4, JULY/AUGUST 2014

[15] Christopher S. Gates, Jing Chen, Ninghui Li, Senior Member, IEEE, and Robert W. Proctor, "Effective Risk Communication for Android Apps" IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 3, MAY-JUNE 2014

[16] Zhiyong Shan and Xin Wang "Growing Grapes in Your Computer to DefendAgainst Malware" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 2, FEBRUARY 2014

[17] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang ,"Catch Me If You Can: Evaluating Android Anti-Malware Against Transformation Attacks" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 1, JANUARY 2014

[18] Wei Peng, Student Member, IEEE, Feng Li, Member, IEEE,Xukai Zou, Member, IEEE, and Jie Wu, Fellow, IEEE "Behavioral Malware Detection in Delay Tolerant Networks", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 1, JANUARY 2014

[19] Junghwan Rhee, *Member, IEEE*, Ryan Riley, *Member, IEEE*, Zhiqiang Lin, *Member, IEEE*, Xuxian Jiang, and Dongyan Xu, *Member, IEEE*, "Data-Centric OS Kernel Malware Characterization" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 1, JANUARY 2014

[20] Guillermo Suarez-Tangil, Juan E. Tapiador, Pedro Peris-Lopez, and Arturo Ribagorda, "Evolution, Detection and Analysis of Malware for Smart Devices", IEEE COMMUNICATIONS SURVEYS &

TUTORIALS, VOL. 16, NO. 2, SECOND QUARTER 2014

[21] http://www.symantec.com/connect/blogs/another-media-stealing-app-found-google-play

[22] "Global mobile statistics 2014 part a: Mobile subscribers;handset market share; mobile operators," http://mobiforge.com/research-analysis/global-mobile-statistics-2014-part-a-mobile subscribers-handset-market-share-mobile-operators, 2014.

[23] D. Venugopal, "An Efficient Signature Representation and Matching Method fo Mobile Devices," *Proc. 2nd Annual International workshop on Wireless Internet* (WICON '06), Boston, MA, United States, 2006. doi: 10.1145/1234161.1234177.

[24] Q. Yang, R. H . Deng, Y. Li, and T.Li, "On the Potential of Limitation-oreinted Malware Detection and Prevention Techniques on Mobile Phones," *International Journal of Security and its Applications*, vol. 4, no. 1, Jan. 2010.

[25] G. Lawton, "Is It Finally Time to Worry about Mobile Malware?," *Computer*, vol. 41, no. 5, May 2008, pp. 12-14, doi:10.1109/MC.2008.159.

[26] D. Dagon,T. Martin, and T. Starner, "Mobile Phones as Computing Devices,the Viruses are Coming!," *Pervasive Computing, IEEE*, vol. 3,no. 4, Oct-Dec. 2004, pp. 11-15. doi: 10.1109/MPRV.2004.21.

[27] M. Howell, S. Love, and M. Turner, "User Characteristics and Performance with Automated Mobile Phone Systems," *International Journal of Mobile Communications*, vol. 6, no. 1,2008, pp.1-15.

[28] A. Shmidt, F. Peters, F. Lamour, and S. Albayrak, "Monitoring Smartphones for Anomaly Detection," *Proc. 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, 2008.

[29] Lamia Ketari and Mohammadi Akheela Khanum, "A Review of Malicious Code Detection Techniques for Mobile Devices", *International Journal of Computer Theory and Engineering Vol. 4, No. 2, April 2012*

[30] K. Sharma, T. Dand, T. Oh, *et al.*, "Malware Analysis for Android Operating", 8th Annual Symposium on Information Assurance (ASIA'13), vol. 31, **(2013)**.