IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Research and Analysis of Energy Efficient Monet for Malware Variants Detection System for Android

Khandagale Swati M[1], Mr. Shyam S. Gupta[2], Prof Sanjay Jain[3]

Department of Computer Engineering, Siddhant College of Engineering, Sudumbare, Pune (MH) [1, 2, 3]

Abstract: The most popular mobile Operating System is Android. Due to its popularity it attracts many malware attacks. In fact people have uncover around one million new malware samples per quarter and it was reported that over 98 % of these new malware samples are in fact variants from existing malware families. In this paper we first show that runtime behaviors of malware's core functionalities are in fact similar within a malware family. We propose a framework to combine runtime behavior with static structures to detect malware variants. We present the design and implementation of MONET which has a client and backend server module. The client module is a light-weight in device app for behavior monitoring and signature generation and we expound these using two novel interception techniques. The backend server is responsible for excellence scale malware detection. Our analysis shows that MONET can achieve accuracy in detecting malware variants. This paper examines the problem of malware in android and recent progress made in detection techniques. We first present a detailed analysis on how malware has evolved over the last years for the most popular stage. We identify show behaviors pursued goals infection and distribution strategies and provide numerous examples through case studies of the most relevant specimens. We next survey classify and ventilate efforts made on detecting both malware and other suspicious software.

Keywords: Code Offload Malware Detection, Android, Runtime Behavior, Mobile Computing, Energy Management, Jade System, Distributed Computing.

I. INTRODUCTION

Android operating system has the maximum market in the year 2014.Which makes it the most extensively used operating system in the world. This makes the android users the biggest target group for malware developers. Permission based mechanism plays a vital role in android security which limit the accesses of third party android applications to effect resources. Piece of android malware can raise concession by sending short service messages by making calls to insurance numbers sharing local information through Global Positioning System (GPS) and finally collects very bulk of information without the user's knowledge. The requirement for android malware detection is increased as security mechanism in android does not set boundary on the system resource usage. This is a critical sensitivity point for malicious applications.

Mobile technology has extended dramatically spheric the world. These day smart mobile devices are used for different motive like personal mobile communication, data storage, and multimedia financial transactions and also for the entertainment. Android is a popular mobile operating system the reasons for being favoured by users are listed as Open source software being supported by Google applications being developed in the most popular programming language Java being open for the customization. Smartphone usages have brought new information exchange to everyday life. The number of mobile malware is growth speedily with distinct malicious activities such as stealing user personal data sending insurance messages and making calls without the user's acknowledgement. These malicious activities are hidden from the user and they run in the background. Many studies have developed methods to detect such attacks.

II. STUDY ON METHODS

In this section, various techniques of Malware Detection are discussed and analyzed.

H. Qian and D. Andresen (2015).
1) This author presenting Extending Mobile Device Battery Life by Offloading Computation Cloud. The requirement for growth performance of mobile device directly conflicts with the want for longer battery life. Offloading measurement to resourceful servers is an effective method to rundown energy consumption and enhances performance for mobile applications. Android provides mechanisms for creating mobile applications but lost a congenital scheduling system for determining where code should be executed. This paper presents Jade system that adds sophisticated energy aware measurement offloading capabilities to android applications. Jade monitors device and application status and automatically decides where code should be executed. Jade dynamically adjust offloading strategy by adapting to workload variation, communication costs and device status.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

K. Xu, Y. Li, and R. H. Deng. (2016)
2) Most existing mobile malware detection methods are designed based on the resources required by malwares (e.g. permissions, application programming interface (API) calls and system calls). These techniques capture the interactions during mobile apps and Android system but ignore the communications among components within application boundaries. As a consequence the majority of the existing techniques are small effective in identifying many typical malwares which need a little or no suspicious resources but leverage on inter component communication (ICC) mechanism. To address this challenge we propose a new malware detection method named ICC Detector. After manually analyzing false positives we discover 43 new malwares from the benign data set and rundown the number of false positives to seven. More importantly ICC Detector discovers 1708 more advanced malwares than the benchmark while it misses 220 obvious malwares which can be easily detected by the benchmark.

M. Lindorfer, M. Neugschwandtner, and C. Platzer, Marvin. (2015)
3) In contravention the considerable number of proposed malware analysis systems generic and practical malware analysis solutions is rare and often short-lived. Systems relying on static analysis oneself conflict with growth popular disturbance and dynamic code loading techniques while purely dynamic analysis systems are prone to analysis evasion we present MARVIN a system that bunch static with dynamic analysis and which transfer machine learning techniques to assess the risk associated with unknown android apps in the form of a hatred score. MARVIN performs static and dynamic analysis both of device to represent properties and behavioral aspects of an app through a rich and comprehensive feature set. In our valuation on the largest android malware distribution data set to date comprised of over 135,000 Android apps and 15,000 malware samples MARVIN correctly classifies 98.24% of malicious apps with less than 0.04% false positives. We another approximation the necessary retraining interval to maintain detection performance and demonstrate the long term practicality of our entreat.

H. Qian and D. Andresen. (2015)
4) Android supply mechanisms for creating mobile applications but deficiency a native scheduling system for determining where code should be executed. Jade a system that adds sophisticated energy aware measurement offloading capabilities to android applications. Jade monitors device and application status and automatically determine where code should be executed. Jade dynamically adjusts offloading strategy by adapting to workload difference communication costs and device status. Jade minimizes the burden on developers to construct applications with computation offloading ability by providing easy to use Jade API. Valuation shows that Jade can effectively reduce up to 35% of average power consumption for mobile device while improving application performance.

W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri. (2011)
5) The fluidity of application markets complex smart-phone security. Although current efforts have light on particular security issues there remains small insight into large security characteristics of smart-phone applications. We introduce the deddecom-piler which retrieval android application source code directly from its installation image. We design and exe-cute a horizontal study of smart-phone applications based on static analysis of 21 million lines of retrieval code. Our analysis uncovered generic use or misuse of personal phone identifiers and deep penetrate of advertising and analytics networks. However we did not find proof of malware or exploitable vulnerabilities in the studied applications. We conclude by considering the implications of these preliminary findings and offer directions for future analysis.

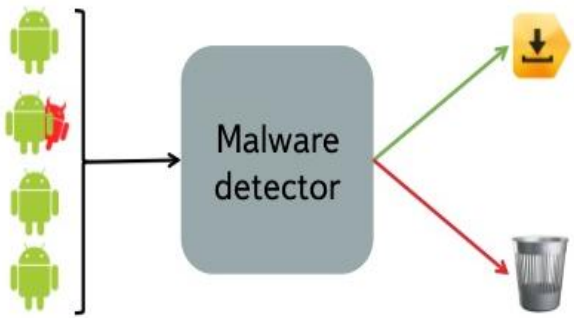

Figure 3: Malware Detector System

III.COMPARATIVE ANALYSIS

The methods studied above are compared in terms of advantages, disadvantages, techniques and accuracy performance. Table 1 is showing the comparative study among these methods.

Table 1: Malware Detection Techniques

Paper Title	Key Techniques and Methods	Advantages	Disadvantages
Extending Mobile Device Battery Life by Offloading Computation Cloud.	Offloading Computation Cloud.	Reduce energy consumption and enhance performance for mobile applications.	Shorten the response time

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

Mobile Malware Detection	Mobile Malware Detection Methods	Capture the Interactions between mobile apps and Android system.	Ignore the communications among components within or cross application boundaries.
Analysis of Malware Detection	Static Analysis Techniques	It is cheap fast not very resource consuming technique	Have to know Malware patterns Or signatures in patterns
Analysis of Malware Detection	Dynamic Analysis Techniques	Detection of unknown attacks	Highly resource consuming not Feasible for Battery Devices
Intelligent Multi-agent system based on Jade	Jade System	Jade can effectively reduce energy consumption of mobile devices, and dynamically change its offloading strategy according to device Status.	Decrease Low performance because of code offloading.

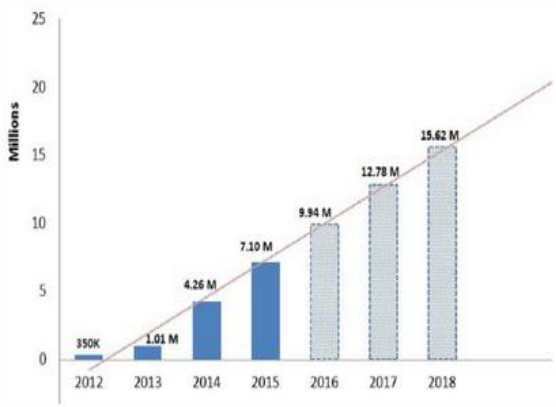

Figure 2. Future Trends of Android Malware Growth

IV. LIMITATIONS OF EXISTING WORK

- Existing HUI methods does not use down-ward closure property
- Existing HUI methods frequently present a large number of high utility item sets to users which makes difficult to end user to comprehend the results.
- Existing algorithms are inefficient in terms of time and memory requirement, or even run out of memory.

The recent efficient HUI representation approach is overcoming previous limitations but still time improvement is major research challenge.

V. RESEARCH PROBLEM

Later on research the fresh techniques and comparing their performances. In this portion the going drawbacks and research challenges are noted for future work. Working on EEMONET is very imperative now days hence its must that technique should be capability of all malware detection in all viewpoints. Recently many researchers did work to deliver the best solution to Detect Malware in Android but we had below observations through our study.

Two approaches of malware detection from users mobile such as static malware detection and dynamic intrusion detection systems. In literature under these two categories number of solutions introduced but suffered from the limitations in terms of efficiency and scalability. Such methods are not supporting to scan the runtime behaviours of malware in order to detect them successfully. Recently another hybrid solution introduced in which runtime behaviour with static structures is combined to detect malware variants efficiently this method called as Monet. The problem identified with Monet is power consumption or energy efficiency is not addressed while malware detection.

VI. CONCLUSION AND FUTURE WORK

The malware are categorized on the basis of detection methods they use. A detailed performance evaluation of these malware techniques is also supply and the advantages and limitations of these malware are deduced comprehensively. Concept of hybrid malware is presented which will address the limitations of existing static and dynamic approaches.

In future it is aimed to implement the proposed hybrid solution which will be a generic malware that will provide better security for android devices by firstly statically analyzing the Android applications on local device and then it will perform dynamic analysis on a remote malware server. This will consume very small amount of memory space on the device and the battery consumption will also be low as all dynamic analysis will be performed at the remote server.

REFERENCES

[1] H. Qian and D. Andresen. Extending Mobile Devices Battery Life by Offloading Computation to Cloud. In Proceedings of the 2nd ACM International Conference on Mobile Software Engineering and Systems (MOBILE Soft), 2015.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

[2] H. Qian and D. Andresen. An Energy-saving Task Scheduler for Mobile Devices. In Proceedings of the 14th IEEE/ACIS International Conference on Computer and Information Science (ICIS), 2015.
[3] H. Qian and D. Andresen. Emerald: Enhance Scientific Workflow Performance with Computation Offloading to the Cloud. In Proceedings of the 14th IEEE/ACIS International Conference on Computer and Information Science (ICIS), 2015.
[4] Q. Chen, H. Qian et al. BAVC: Classifying Benign Atomicity Violations via Machine Learning. In Advanced Materials Research, Vols 765-767, pp. 1576-1580, Sep, 2013.
[5] F. Wei, S. Roy and S. Ou. Am android: A Precise and General Inter component Data Flow Analysis Framework for Security Vetting of Android Apps. In proceedings of the 2014 ACM Conference on Computer and Communications Security. 2014.
[6] L. Peng, Y. Yang et al. Highly Accurate Video Object Identification Utilizing Hint Information. In proceedings of the International Conference on Computing, Networking and Communications (ICNC), 2014.
[7] S. Zhang, X. Zhang and X. Ou. After We Knew It: Empirical Study and Modeling of Cost-effectiveness of Exploiting Prevalent Known Vulnerabilities Across IaaS Cloud. In proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2014.
[8] D. Arp, M. Spreitzenbarth, M. H ubner, H. Gascon, K. Rieck, and C. Siemens, Drebin: Effective and explainable detection of android malware in your pocket, in Prof. of the Network and Distributed System Security Symposium, 2014.
[9] S. Roy, J. DeLoach, Y. Li, N. Herndon, D. Caragea, X. Ou, V. P.Ranganath, H. Li, and N. Guevara, Experimental study with real world data for android app security analysis using machine learning, in ACSAC. ACM, 2015.
[10] K. Xu, Y. Li, and R. H. Deng, Iccdetector: Icc-based malware detection on android, TIFS, 2016.