INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

MOBSHIELD: ANALYZING MOBILE APPLICATIONS SECURITY

MS. PUJA S.TEKADE[1], PROF. A. S. KAPSE[2]

1. M. E. Second Year CSE, P. R. Patil COE&T, SGBAU, Amravati, Maharashtra ,INDIA
2. Assistant Professor, Dept. of CSE, P. R. Patil COE&T, SGBAU, Amravati, Maharashtra, INDIA

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: With day to day increase in the number of mobile applications there is an analogous increment in the mobile threats. For such kinds of threats to mobile devices there should be some security mechanism to be implemented. In the proposed system in order to improve the security to the mobile apps one methodology is proposed which will evaluate the mobile applications based on the cloud computing platform and data mining. Here also a prototype system named MobShield is presented to identify the mobile app's virulence or benignancy. Compared with traditional method, such as permission pattern based method, MobShield combines the dynamic and static analysis methods to comprehensively evaluate an Android app. In the implementation, Android Security Evaluation Framework (ASEF) and Static Android Analysis Framework (SAAF) are adopted , the two representative dynamic and static analysis methods, to evaluate the Android apps and estimate the total time needed to evaluate all the apps stored in one mobile app market. As mobile app market serves as the main line of defense against mobile malwares, the evaluation results show that it is practical to use cloud computing platform and data mining to verify all stored apps routinely to filter out malware apps from mobile app markets. In this proposed system the concept will be extended with the implementation of K-means algorithm.

Keywords: Android Platform; Mobile Malware Detection; Cloud Computing; Forensic Analysis; Machine Learning; Data Mining.

Corresponding Author: MS. PUJA S.TEKADE

Access Online On:

www.ijpret.com

How to Cite This Article:

Puja S. Tekade, IJPRET, 2015; Volume 3 (9): 1377-1383

PAPER-QR CODE

INTRODUCTION

Mobile operating system and the corresponding mobile applications are increasing rapidly in quality and quantity. Today businesses serve the customers through all sort of media that include email, sms, social media and corporate web sites. Now the fact is that the use of mobiles, tablets is increasing and hence the businesses turned their attention toward them. About 35 percent of U.S. adults now own Smartphone's, and Frost & Sullivan predicts that number will rise to more than 80 percent by 2015[1].Worldwide, more than 1 billion people will possess Smartphone's by 2013.To support the explosive popularity of mobile devices the cellular companies are providing vast range of resources and facilities .With today's high speed, high bandwidth cellular network resource-intensive features like real-time multimedia streaming and videoconferencing are now readily available to mobile device users the world over.

Including all these brings the new generation of mobile devices with tremendous capability and explosive amount of mobile applications. For example, more than 500,000 mobile apps are available for Apple's iOS mobile devices alone[1]. Many of these applications are developed to help businesses, education, society and entertainment.

These days as the use of mobile applications are increasing, similarly the threats towards them are increasing. Mobile security is now became an important in mobile computing. It is essential as it relates to the businesses and personal data of the mobiles of particular user [2]. So the various strategies are introduced in order to prevent from the threats. Different security counter-measures are being developed and applied to Smartphone's, from security in different layers of software to the dissemination of information to end users [3].

Attacks are based on mainly communication aspects. First category includes 'attack based on SMS and MMS', those attacks derive from flaws in management of SMS and MMS. Next category includes 'attacks based on communication networks', 'attacks based on GSM networks' where the attacker may try to break the encryption of mobile networks. Once the encryption algorithm of GSM is broken, the attacker can interpret all unencrypted communications made by the victims Smartphone [4]. The mostly attacks are based on vulnerabilities in software applications by which the attacker can gain access to the local data of mobile and misuse it. The proposed work focuses on this category of mobile attacks.

2. LITERATURE REVIEW

Jianlin Xu, Yifan Yu, Zhen Chen have proposed a system which has combination of dynamic and static frameworks to provide security to commercial mobile applications [5]. They have analyzed the amount of time needed to indicate and find out the non-secure applications among the installed apps. Based on home-brewed cloud computing platform and data mining, they proposed a methodology to evaluate mobile apps for improving current security status of mobile apps, MobSafe, a demo and prototype system, is also proposed to identify the mobile app's virulence or benignancy.

David Barrera, H. Güne¸s Kayacık contributed on novel methodology for exploring and empirically analyzing permission-based models [6]. They employed their methodology for the analysis of 1,100 applications written for the Android OS. Using the Self-Organizing Map (SOM) algorithm , they identified trends in how developers of these applications use the Android permissions model. They found that while Android has a large number of permissions restricting access to advanced functionality on devices, only a small number of these permissions are actively used by developers. Our analysis identifies permissions that are overly broad (i.e., controlling access to a large set of features). Furthermore they identify application clusters based on requested permissions, and extract the prominent permissions within each cluster. Our empirical observations provide a basis for possible enhancements to the Android permission model.

Dai-Fei Guo analyzed a mobile malware behavior analysis method based on behavior classification and self-learning data mining is proposed to detect unknown or metamorphic mobile malware [7]. A behavior classification based mobile malware detection method is proposed to analyze the network behavior of the new or metamorphic mobile malware which is improved gradually with an incremental self-learning method.

Wenhui Hu, Damien Octeau proposed Duet: a library integrity verification tool for Android applications at application stores [8]. Duet first collects the original library files from their providers. With the observation that reverse-engineered library files go through a build process and a reverse-engineering process, Duet takes a novel mirroring approach in which original files also go through a build process and a reverse-engineering process in order to create reference files.

3. Proposed work

Basically the concept is to verify the particular application before it is installed into the mobile so that the possible threat will be avoided. By that the apk is taken and analysed by compared with the signatures of threats. If it is found then the apk is declared with particular results. For doing the analysis the two representative techniques are used which are –ASEF, SAAF.

MobShield is a system to check whether an Android app is virulence or benignancy based on some customized tools in cloud platform. In this work a methodology is being proposed which provides security to your mobile phone from malwares that comes from faulty apps.

Here following techniques will be referred

- Cloud computing
- Data mining

In this work a methodology is being proposed which provides security to your mobile phone from malwares that comes from faulty apps.

It includes following:

1. It includes combination of dynamic and static methods to provide security.
2. It includes data mining techniques.
3. In this work, a methodology i.e. MobShield is proposed to evaluate the security of Android mobile apps based on cloud computing platform.
4. In this work ASEF and SAAF are adopted, the two representative dynamic analysis method and static analysis method, to evaluate the Android apps
5. Estimate the total time needed to evaluate all the apps stored in a mobile app market.

MobShield is an automatize system which can be used to analyze Android apps. When you submit an unknown apk file to MobShield for analysis, it will check the key value store whether the apk is already analyzed and its result is stored in storage.

Following flowchart shows the working of the project:

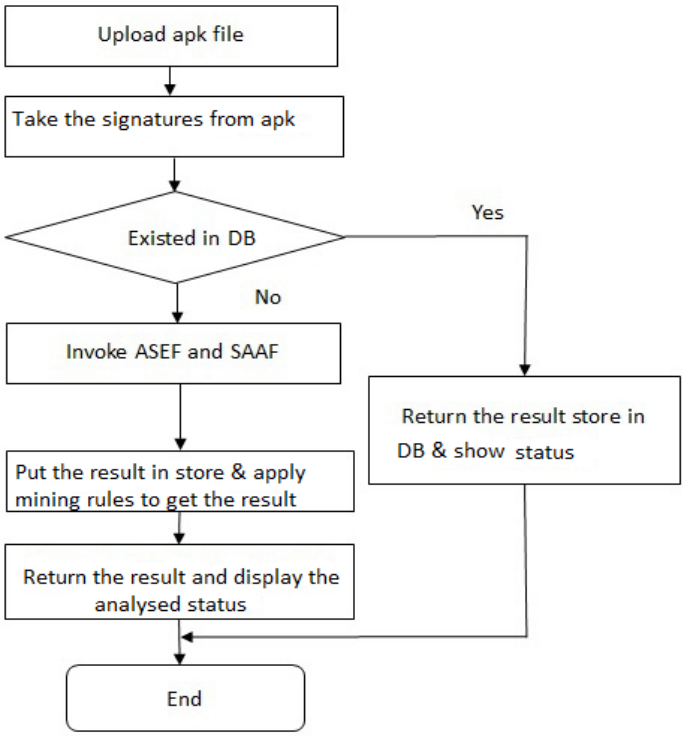

Fig.1:- Working of project

As shown in the above flowchart, the unknown apk will be submitted to the system, then the application is parsed and its signatures are taken and stored in database, then the signatures of app are compared with the signatures stored in the database which are of threats. Here the mining technique is used by which signatures are compared and if they are matched the result will be displayed. The time needed to do the analysis is also displayed.

As the possibility exist that even though we know that there might be presence of threat in a particular app still we need to install it in case of emergency , in that case one additional facility will be provided in this work. Here the backups are taken periodically, so in case of mobile attack the data will be recovered and most of data will get prevented from loss.

4. Result Analysis

In order to do the result analysis of the project three major outcomes are considered. Based on the three outcomes the whole work is analyzed and evaluation is done. Those are discussed as follows:

- **Security:** As above discussed that in order to prevent the mobile attacks the analysis of application is done. Firstly the signatures of threats are stored in database. The user will

then upload the apk from his mobile onto the cloud system, where the signatures are compared and the security status of the apk is displayed. By that the user came to know about the fraudulent apk in advance and thus prevented from the mobile attack. Thus the security to mobile applications is provided by the system and user is notified about the threat in advance.

- **Time analysis:** By this project the amount of time required to compare, analyze and evaluate the security status of application is provided. Thus the time constraints are displayed in addition to the result of analysis.

- **Facility of Backups:** In some cases even though we know that the particular application may harm the mobile still because of emergency and requirement we need to install it. In such situation the project provide an alternative solution of backups. The backups are taken from the user mobiles and uploaded over cloud in a periodic manner. In between those points if an attack happens, then maximum data can be updated through cloud. Therefore the project provides the facility of backups and thus provides security to the user data in addition to mobile applications.

5. CONCLUSION

- By this work massive data can be protected from malwares and the database can be protected

- Here ASEF and SAAF can be used to evaluate the Android apps and estimate the total time needed to evaluate all the apps stored in a mobile app market.

- Also here data mining technique such as Top k rules will be used in order to get optimized outcomes

6. FUTURE SCOPE

- As the future perspective some web mining and more advanced data mining techniques will be implemented to get more optimized outputs

- Machine learning is the issue which will be in the future work of this system

- The system can be extended to include the facility of providing the user security by which the user will be notified about the safe and unsafe contacts and numbers and generate the report of one user.

REFERENCES

1. R. Lawler, Mary Meeker's 2013 Internet Trends report, http://techcrunch.com/2013/05/29/mary-meeker-2013- internet-trends/, September 2, 2014.

2. Asaf Shabtai Yuval Fledel, "Automated Static Code Analysis for Classifying Android Applications Using Machine Learning", IEEE 2010 International Conference on Computational Intelligence and Security, pp.321-332.

3. A. D. Schmidt, R. Bye, H. G. Schmidt, J. Clausen, O. Kiraz, K. A. Yuksel, S. A. Camtepe, and S. Albayrak, "Static analysis of executables for collaborative malware detection on Android in Communications", ICC'09, IEEE International Conference on, Dresden, Germany, 2009, pp. 1-7.

4. T. Li, F. Han, S. Ding, and Z. Chen, "LARX: Large-scale Anti-phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform", in Proc. 20th International Conference on. IEEE. Computer Communications and Networks (ICCCN), Maui, Hawaii, USA, 2011, pp. 1-5.

5. Jianlin Xu , Yifan Yu, Zhen Chen_, Bin Cao, Wenyu Dong, Yu Guo, and Junwei Cao, "MobSafe: Cloud Computing Based Forensic Analysis for Massive Mobile Applications Using Data Mining", TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214ll10/10,Volume 18, Number 4, August 2013, pp. 418-427.

6. D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to Android, in Proc.", 17th ACM Conference on Computer and Communications Security, Chicago, USA, 2010, pp. 73-84.

7. Dai-Fei Guo1, Ai-Fen Sui1, Yi-Jie Shi, "Behavior Classification based Self-learning Mobile Malware Detection", JOURNAL OF COMPUTERS, VOL. 9, NO. 4, APRIL 2014, pp. 851-857.

8. Wenhui Hu, Damien Octeau, and Patrick McDaniel, "Duet: Library Integrity Verification for Android Applications", in Proc. 2nd ACM conference on Data and Application Security and Privacy, San Antonio,TX, USA, February, 2012, pp. 317-326.