International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 4 – MARCH 2017.

ENHANCED SECURE APP MODEL FOR THE PREDICTION AND PREVENTION OF MALWARE IN ANDROID SYSTEM

S.GRACIA ROSALIN SOPHIE [#1] and K.SHANMUGAPRIYA [*2]

[#] *B.E,Computer Science,V.R.S College of Engineering And Technology,Arasur ,Tamilnadu.*

[*] *B.E,Computer Science,V.R.S College of Engineering And Technology,Arasur ,Tamilnadu.*

***Abstract*— Android is an open source software developed by Google. It is a mobile application software used mainly in touch screen devices such as mobiles, tablets, etc.. It was found under the Google play store where the all android applications are available. The users and popularity of android application enables the attackers to cause threats in stealing the data and termination of services, etc. Previous work has been designed to improve the sensitivity data but fails to satisfy the user requirements. In this paper, we have proposed an enhanced machine learning model to prevent and predict the malware applications.In order to find the malware app from the normal app, data flow API features are extracted. The data flow analysis is used to check whether the sensible data leaves out the system.The sensible data flow path is used to predict the behavior of the others. The k nearest algorithm is used to differentiate the malicious apps from the normal apps. Experimental results have shown the effectiveness of the systems.**

***Index Terms*— Android, Google playstore, Machine learning model, k nearest neighbor algorithm, malicious app.**

I. INTRODUCTION

Since 2008, the rate of smartphone adoption has increased tremendously. Smartphones provide different connectivity options such as Wi-Fi, GSM, GPS, CDMA and Bluetooth etc. which make them a ubiquitous device. Google says, 1.3 million Android devices are being activated each day [1]. Android operating system left its competitors far behind by capturing more than 78% of total market share in 2013 [2]. Gartner report 2013 of smartphone sales shows that there is 42.3% increase in sales of smartphones in comparison with 2012. According to International data corporation IDC, Android OS dominates with 82.8% of total market shares in 2Q 2015 [3]. It could be observed that Android has become the most widely used operating system over the years.

The malware are termed as malicious software that is designed specifically to target a mobile device system, such as a tablet or smartphone to damage or disrupt the device. Most mobile malware is designed to disable a mobile device, allow a malicious user to remotely control the device or to steal personal information stored on the device [3]. 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].

People use smartphones for many of the same purposes as desktop computers: web browsing, social networking, online banking, and more. Smartphones also provide features that are unique to mobile phones, like SMS messaging, constantly-updated location data, and ubiquitous access. As a result of their popularity and functionality, smartphones are a burgeoning target for malicious activities [4]. In order to understand the motives of real mobile malware, we classify the malware in our data set by behavior. We find that the most common malicious activities are collecting user information (61%) and sending premium-rate SMS messages (52%), in addition to malware that was written for novelty or amusement, credential theft, SMS spam, search engine optimization fraud, and ransom. We describe the incentives that promote each type of malicious behavior and present defenses that disincentivize some of the behaviors [5]. In particular, malware that abuses SMS messages could be prevented with small changes to the Android and Symbian platforms. We also discuss incentives that we believe will motivate future mobile malware.

The rest of the paper is organized as follows: Section II describes the related work; Section III presents the proposed work; Section IV depicts the experimental analysis of the proposed work and concludes in Section V.

II. LITERATURE SURVEY

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 [6]. 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)[6].

The author in [9] evaluated Android malware to find out botnet behavior. The purpose of identifying botnet behavior was to find specific trends and characteristics. These trends had extracted from android code and structure. As a result, Author identified characteristics and explored them in terms of Android botnet invention process. This process includes infection, propagation and execution of malware. Finally this process lead to botnet maturity model for Android. The author in [10] Focused on the Android platform and characterize existing Android malware in 49 distinct malware families. An Android botnet is a network consisting of compromised Android smartphones controlled by a botmaster through a command and control (C&C) network [12]. The sophistication of Android botnets is increasing very rapidly. Traditional botnets communicate to the C&C server using IRC (Internet Relay Chat) protocol or using P2P (Peer-to-Peer) overlays. Popular botnet malwares like Geinimi, Pjapps, DroidDream and RootSmart exhibit traditional communication behaviour. The C&C server address is generally encrypted and stored so that it can surpass detection mechanism. A malware Geinimi applies DES encryption scheme to encrypt its communication to the server. Geng, Guining et al. [13] has proposed a new attack vector leveraging SMS structures for creating a heterogeneous mobile botnet model. Botnets can generally be used for various types of attacks such as spam delivery, DDoS attacks and for stealing personal data. A full blown botnet will have safety measures that cause bots to abort the mission and erase all traces of their existence if they are compromised. In this way, they can protect the botmaster and the C&C server.

The author in [12] presented a paper in proceedings ACM named Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints. In this paper, Apex provides framework for Android that allows a user to selectively grant permissions to applications as well as impose constraints on the usage of resources. They also describe an extended package installer that allows the user to set these constraints through an easy-to-use interface. The author in [13] presented a paper in proceedings ACM named Android Permissions: A Perspective Combining Risks and Benefits. In this paper, they investigate the feasibility of using both the permissions an app requests, the category of the app, and what permissions are requested by other apps in the same category to better inform users whether the risks of installing an app is commensurate with its expected benefit [14].

The author in [15] presented a paper in proceedings UBICOMM named Permission Tracking in Android. In this paper, they have a closer look at permissions that users grant to apps in Android, a wide-spread operating system for mobile devices like smart phones. They developed tool that allows users to administer permissions of their applications. They enable users to allow or deny permissions at any time.

III. PROPOSED WORK

This section explains about the general flow of proposed k nearest neighboring algorithm. It includes four steps which are explained as follows:

A. *Application Security*

Application security is the major part of android application. It is used to protect the mobile applications from the intruders. In this step, the user has to register and password verification has been done for security purposes. The application does not contain antivirus, where it is fault analysis application by injecting fault files into the applications.

B. *Fault Injection*

Fault injection is a technique used to improve the coverage of a test by injecting the fault files into the application. It is used for error handling.In this step the user have to login with E-MADAM applications, the app allows the user to inject the faulty files into the mobile application. It leads the user to inject the virus file into the mobile app,it contains malfunction.

C. *Fault detection and analysis*

As the faulty files are injected into the mobile, the virus file start to malfunction inside the mobile application. The virus can be detected and analyzed by using machine learning model. This application is used to predict the difference between the original application and malfunction application.

D. *Report generation*

In this step, the report can be generated using the machine learning model. This developed machine learning model is applied over the apps presented in the android systems. By using the results graph can be generated to understand the application results.

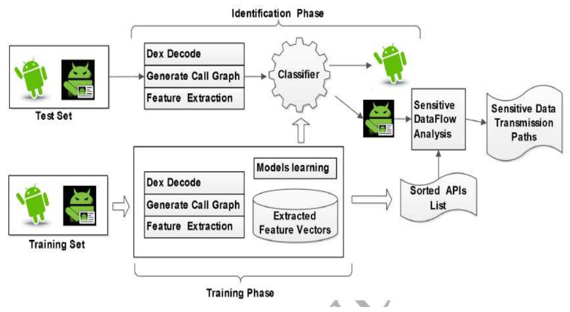


Fig.1. Proposed architecture

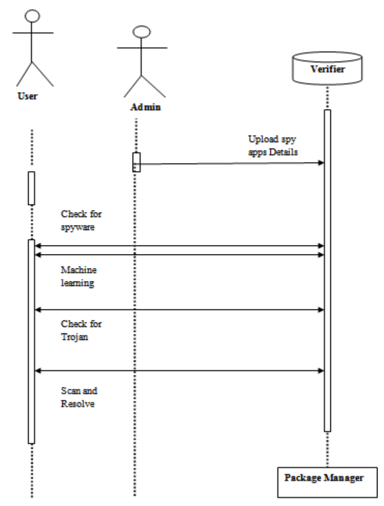


Fig.2. Activity diagram for detecting the malicious apps.

IV. EXPERIMENTAL RESULTS

This section presents the validation of the proposed machine learning model using Android Systems.

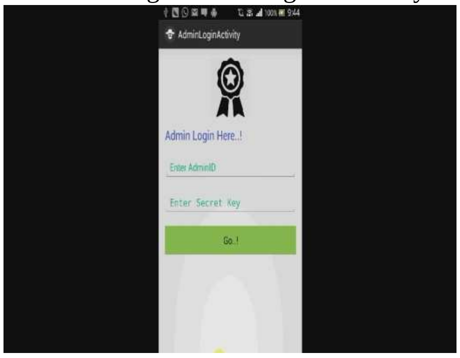


Fig.3. Admin's Login with their credentials

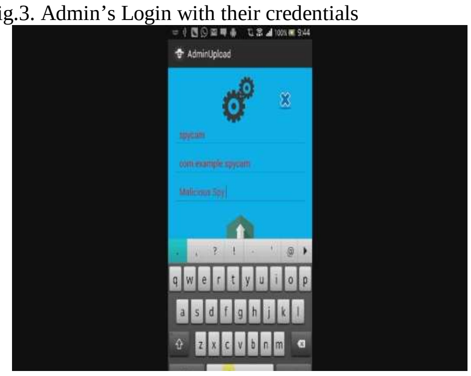


Fig.4. Admin uploading the mobile apps

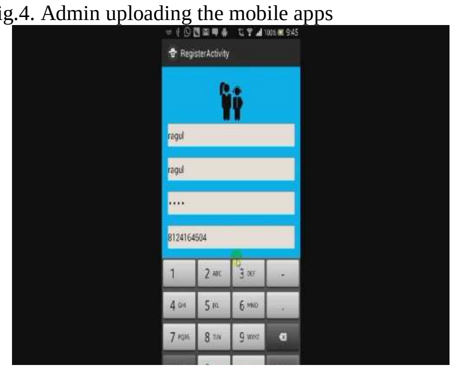


Fig.5. User's sign in with their android systems

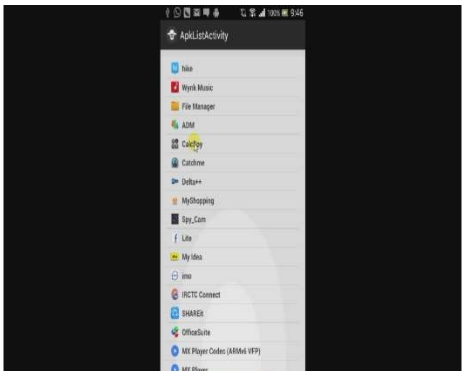
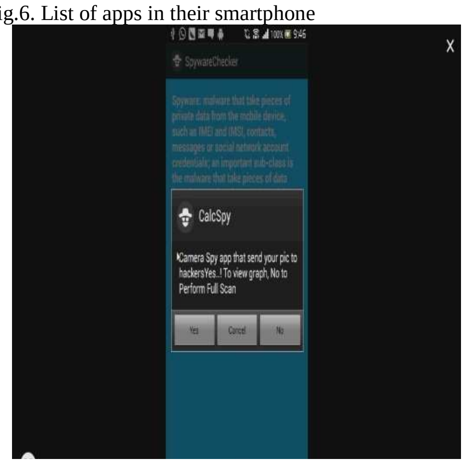


Fig.6. List of apps in their smartphone



Spyware: malware that take pieces of private data from the mobile device, such as IMEI and IMSI, contacts, messages or social network account credentials; an important sub-class is the malware that take pieces of data

Camera Spy app that send your pic to hackersYes..! To view graph, No to Perform Full Scan

Fig.7. Scanning operation performed over calculator app

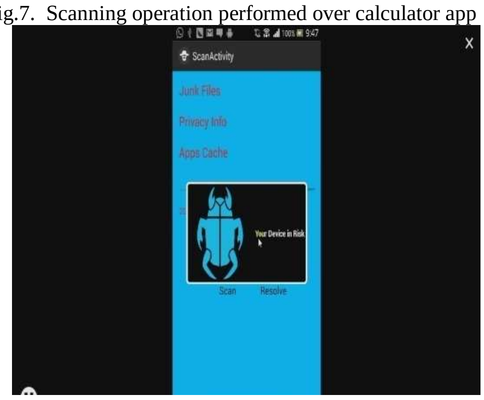


Fig.8. Outcome of the scanned apps

V. CONCLUSION

This study presented techniques to effectively detect the malicious apps which are easy to install and use on its Android based commercial mobile device environment. It also presented the algorithm to discriminate the malicious apps using the algorithm of frequency and similarity analysis of occurring events. The use of techniques presented in this study made it possible to analyze the characteristics of system call events occurring upon executing malicious apps, and can be applied for a way to discriminate whether the arbitrary mobile apps are malicious or not through this. We have implemented a static analysis framework to prevent the applications from malfunctions using machine learning model. One aim is to improve the framework by adding dead code detection to reduce the false positives. The other aim is to integrate it with dynamic analysis for verifying suspicious apps automatically. Experiemental results have shown the effectiveness of the proposed systems.

REFERENCES

[1] Andrea Saracino et al, "MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention", IEEE transactions on Dependable and Secure computing, 2016.

[2] More than 700,000 malicious Android apps wreak havoc on the web. [Online]. Available: http://www.neowin.net/news/more-than-700000-malicious-apps-wreak-havoc-in-the-play-store

[3] Android (operating system). [Online]. Available: http://en.wikipedia.org/wiki/Android_(operating_system)

[4] Uncovering android master key that makes 99% of devices vulnerable. [Online]. Available: http://bluebox.com/corporate-blog/bluebox-uncovers-android-masterkey/

[5] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A study of android application security," in Proc. the 20th USENIX Security Symposium, 2011.

[6] I. Burquera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for Android," in Proc. the 1st ACM workshop on Security and privacy in smartphones and mobile devices, 2011, pp.15-26.

[7] T. E. Wei, C. H. Mao, A. B. Jeng, H. M. Lee, H. T. Wang, and D. J. Wu, "Android malware detection via a latent network behavior analysis," presented at IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012.

[8] D. J. Wu, C. H. Mao, T. E. Wei, H. M. Lee, and K. P. Wu, "Droidmat: android malware detection through manifest and api calls tracing," presented at 2013 7th Asia Joint Conference on Information Security, 2013.

[9] Y. J. Ham, W. B. Choi, H. W. Lee, J. D. Lim, and J. N. Kim, "Vulnerability monitoring mechanism in Android based smartphone with correlation analysis on event-driven activities," in Proc. 2012 2nd International Conference on Computer Science and Network Technology, 2012, pp. 371-375.

[10] Strace. [Online]. Available: http://en.wikipedia.org/wiki/Strace

[11] Y. J. Zhou and X. X. Jiang, "Dissecting Android malware: characterization and evolution," in Proc. the 33rd IEEE Symposium on Security and Privacy, 2012, pp. 95-109.

[12] X. X. Jiang and Y. J. Zhou, Android Malware, NY, USA: Springer, 2013.

[13] M. Nauman, S. Khan, and X. Zhang, "Apex: extending android permission model and enforcement with user- defined runtime constraints," in Proc. the 5th ACM Symposium on Information, Computer and Communications Security, 2010, pp. 328-332.

[14] A. PorterFelt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in Proc. the 1st Workshop on Security and Privacy in Smartphones and Mobile Devices, 2011, pp. 3-14.

[15] W. Zhou, Y. Zhou, X. Jiang, and P. Ning, "Droidmoss: detecting repackaged smartphone applications in third- party android marketplaces," in Proc. the 2nd ACM Conference on Data and Application Security and Privacy, 2012, pp. 317-326.