Mobile App Classification Based On Risk Score

[1]Revati Baviskar, [2]Aashay Raje, [3]Bhagyashri Bhole, [4]Akshay Aachat

[1,2,3,4]UG Student, Department Of Computer Engineering, Late. G.N. Sapkal Collage of Engineering, Nashik, Maharashtra, India.

[1]*revati22kar@gmail.com* , [2]*aashay.raje95@gmail.com* , [3]*bhagyashri.bhole96@gmail.com* , [4]*aachatakshay@gmail.com*

Abstract - The increasing growth of Android phones, due to its openness and popularity also calls for the increase in attacks. A very keen and intellectual study proves that Android users can easily be hacked with the existing single time permission approval system. Once approved, an application can access all the resources requested as permissions, at any time according to the code integrated into it. The permissions shown at installation time totally depends on the users privacy preference. An application rated riskier by a particular privacy rating software may not be riskier in the eyes of a user who is already decided to install this application. So in order to provide a clear cut idea about how an application will harm him or what are the risks involved in an application compared to the other top rated clean applications in the same category, we propose a security system, which involves a combination of static and dynamic risk analysis of Android applications. The proposed system allows its users to install a riskier application rated by static analyzer, and review it with a help of dynamic analyzer, which monitors application activity handling permissions.

Keywords: Android Security, Android Permissions, Malware Detection, Risk Analysis, Android Static and Dynamic Analyzer.

I. INTRODUCTION

Mobile devices are now a very part of our lives, with the evolution in them they now act as ubiquitous devices. With increasing in the use of mobiles there is large number of mobile apps coming in to the market, most of which provide same kind of functionality. Classification of these apps will play an important role not only to the user in order to search the required app easily but also we can have the analysis of the user references which can help the intellectual services like app recommendation, user segmentation, target advertising etc. But the information we get from the app directly is very limited and also ambiguous. Though in the app store the apps are associated with the predefined tags or Meta in- formation about them and this information is used for recognizing their latent semantic meanings. However this data may be difficult to obtain from the third party services when there are multiple app delivering channels and it is not able in identifying the source. Also the tags do many times provide the actual latent semantic meaning of the app, like in the app store of nokia the app called \safe 360" is placed in the category of \business" which too general[1]. Apart from providing the facility of communication mobiles are used for various other services like location searching, camera, storing the data, emails, SMS etc thus they include the personal and sensitive information. This makes their security as more important issue as many of these apps are coming from some unknown vendors as with the tools available it becomes easy to create the mobile apps with little knowledge. The security mechanism provided by the android is in stand-alone fashion [2], i.e. before the app is installed it just gives the list of permissions the app will be accessing and leaves the decision of trusting the app on the user. But to make this decision it is requires that the user should have the some technical knowledge and also as this request of permission comes for every app user's lose interest in this warning, they ignore it and mostly make their decisions based on the reviews and ratings. This error in decision making threatens the security of the personal and sensitive information present in mobiles. In order to effectively solve the above two issues and to have a more improved classification of the mobile apps we are proposing a system in which the classification of the apps will be done by exploiting all the contextual features like information from the labels (app name), information from the web search engine (snippets) and the contextual usage history of the app collected from the users usage record. By integrating all these features together will give us a more improved and accurate classification of the apps. Then in

order to improve the security mechanism, we will be exploiting the list of permissions the app requests the user and based on that will calculate their risk score, i.e. if the app is requesting some critical permissions that are rarely requested by the apps in the same category as the app, then the app will be labeled as risky. The risk score will be given in a simple manner to the user just like we have ratings of the apps i.e. in easy to understand manner. So that the user will have another metric to use while selecting the app and to protect their data from malicious apps.

II. LITERATURE SURVEY

We study how to effectively evaluate the risk of mobile applications, with a focus on the Android platform. The Android platform has emerged as one of the fastest growing operating systems. In May 2013 Google Inc. announced that 900 million Android devices have been activated. Additionally Google Play (formerly known as Android Market) crossed more than 48 billion downloads, and is now averaging about 2.5 billion downloads per month. Such a wide user base, coupled with ease of developing and distributing applications, makes Android an attractive target for malicious developers that seek personal gain while costing users' money and invading users' privacy. One of Android's main defense mechanisms against malicious apps is a risk communication mechanism which warns the user about the permissions an app requires before the app is installed by the user, trusting that the user will make the right decision. The specific approach used in Android has been shown to be ineffective at informing users about potential risks. The majority of Android apps request multiple permissions. When a user sees what appears to be the same warning message for almost every app, warnings quickly lose any effectiveness as the users are conditioned to ignore such warnings.

A.Z. Brooder et al [3],proposed to build a robust query classification system which will identify thousands of query classes with reasonable accuracy. Blind feedback technique is used i.e given a query its topic is determined by classifying the web search results retrieved by the query. Top related search results of the query are obtained from web search engine.

X. H. Phan et al.[5],proposed to leveraged the hidden topics to improve the representation of the short and sparse text for classification. Here semantic topics are the additional textual features integrated with the words to improve the classification.

M. Sahami and T.D. Heilman [4], proposed an approach for measuring the similarity between the short text snippets by exploiting the web search engine to provide greater context for the short texts.

H. Ma, H.Cao ,O.Yang ,E.Chen and J.Tian [6],proposed an approach which lever- ages search snippets to build vector space for both app usage and categories and classifies the app usage records using the cosine space distance.

W.Enack et al [8],proposed a tracking system for real time privacy monitoring on smartphones. here it informs the user when the application may be trying to send sensitive data o_ the phone. But it does not defend against the security and monetary focused malware which send out spam or create premium SMS messages without accessing private information.

E. Chin et al [7], conducted a user study to gain insight into user perceptions of smart phone security and installation habits. Where it found that users don't focus on the permissions during the app browsing and installation, they relay on the user rating and reviews while selecting the app.

A.P.Felt et al [9],uses static analysis to determine if an android app is over privileged, i.e if the is requesting for the permissions it never used .they found in the result that out of the 940 applications one third of the apps are Over privileged.

III. SYSTEM ARCHITECTURE

Classification of the apps but it is still not completely accurate like in the nokia store it is found that the app called \safe360" which is a app for security is classified in \business" category which is too general, as under this \business" category it can be further classified as \security" . Then again with large number of apps coming in the market from various unknown vendors, it is found that many of these apps are malicious, requesting for the access to personal and sensitive data. The defense mechanism of android is found to be in stand-alone fashion i.e. it only warns the user of the list of permissions the app will request and considers that the user will make the appropriate decision. As these warnings come for every app, user ignores them and this causes a threat to any personal or sensitive data present in the mobile. To overcome these problems mentioned above a system can be implemented which will not only classify the data more accurately but will also give the analysis of the risk the app will have in an easy manner like that of ratings which will be easy to understand and make appropriate decision.

We thus propose the concept of risk scoring functions. Such a function assigns to each app a numerical score, which indicates how risky the app is. This approach presents "comparative" risk information, i.e., each app's risk is presented in a way so that it can be easily compared to other

apps. Given a risk scoring function, one can construct a risk signal by choosing a threshold above which the signal is raised. However, we believe that it is better to use a risk scoring function for risk communication in the following way. Given this function, one can compute a risk ranking for each app, identifying the percentile of the app in terms of its risk score. This percentile number has a well defined and easy-to-understand meaning. Users can appreciate the difference between an app ranked in the top 1 percent group versus one in the bottom 50 percent. This ranking can be presented in a more user-friendly fashion, e.g., translated into categorical values such as high risk, medium risk, low risk, and very low risk. An important feature of the mobile app ecosystem is that users often have choices and alternatives when choosing a mobile application. If the user knows that one app is significantly more risky than another with similar functionality, then that may cause the user to choose the less risky one. Such an approach complements well other approaches that try to identify malicious apps. After malicious apps are removed, the remaining ones can be ranked according to their risks.

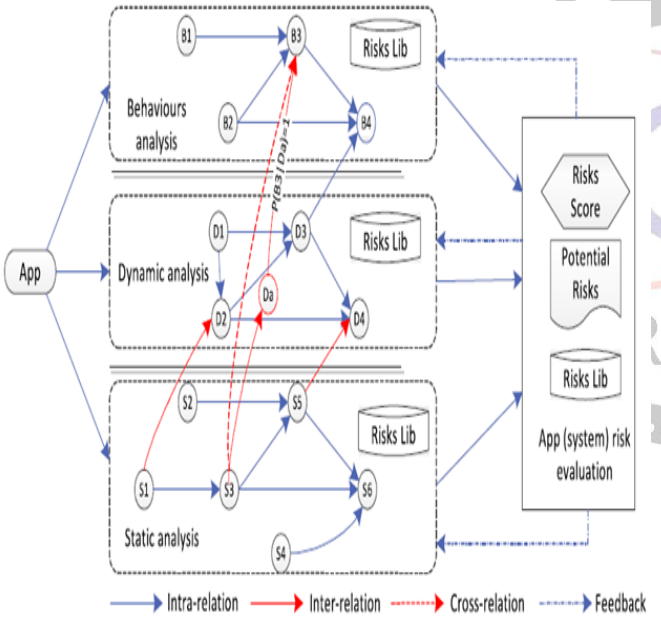

Fig. 1 System Architecture

A. Internal software data structure

APP processing & hierarchical feature extraction are been used for internal data structure.

B. Global data structure

Real time Apps are been used as global datastructure.

C. Temporary data structure

App layered Features.

D. ALGORITHM

Algorithm: To extract app permission.
Input: apk files
Output: List of permissions the app extracts

IV. RESULT ANALYSIS

A. Category

Fig 2 Categorise

This is the first page of application it contains all types of applications. There will be several number of application which we categorise in different category.

B. Apps:

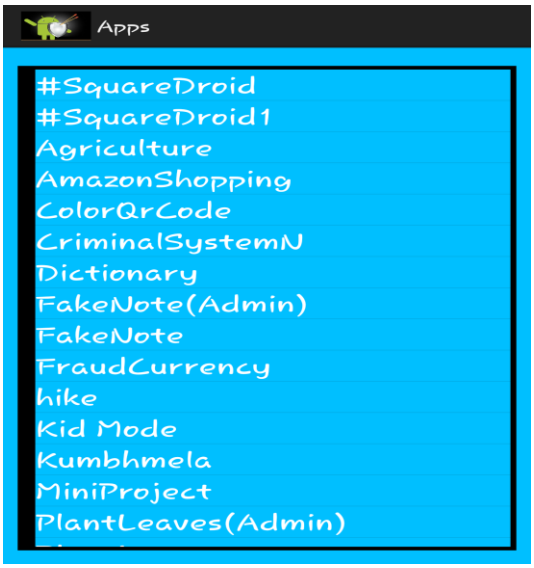

Fig 3 Apps

It contains applications which belongs to specific category.

C. Risk Calculate App

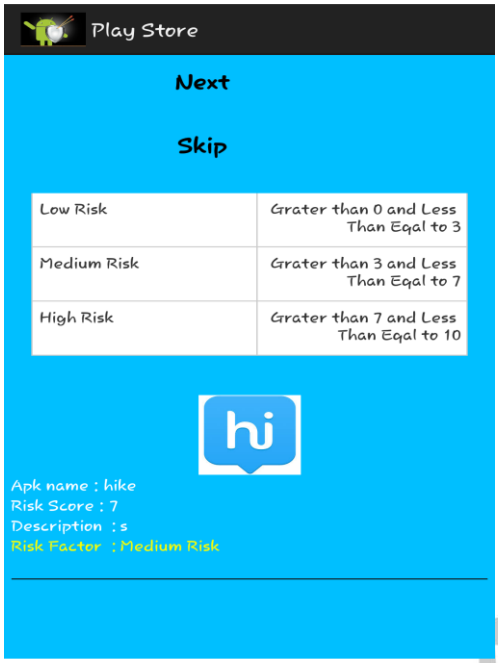

Fig 4. Risk Calculate App

Here we calculate the risk of the application and also we classified this risk between low, medium and high. And we also calculate the risk factor between 0 to 10 range.

D. Malwar Detection App

Fig. 5 Malwar Detection App

Here we shows the malware detection to the user if they are detected. On this basis user can decide whether he download the application or not.

E. Permission form

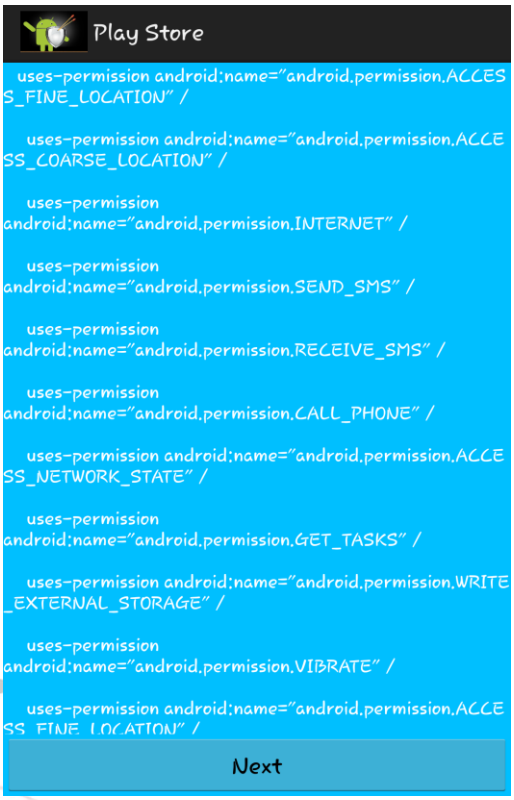

Fig. 6 Permission form

It's a accept process which will be done on user permission.

F. Display Information

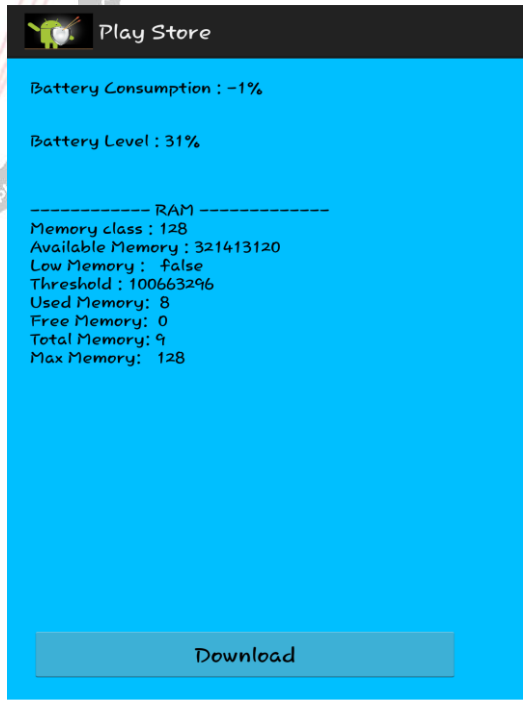

Fig. 7 Display Information

This is the last page or form. Here it display all information about the application. And finally user can download the application on its own decision.

G. Accuracy Graph

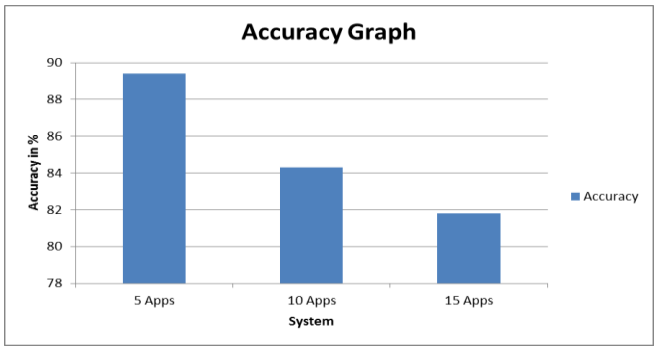

Fig. 8 Accuracy Graph

We had shown the bars of testing accuracy with respect to risk in apps of all above approaches in Fig. above, where training accuracy denotes the classification accuracy of the Apps on the training data. Similarly to the bars of log likelihood, we can observe that the training accuracy bars of all approaches converge quickly and Proposed can achieve a high testing accuracy than existing system.

V. CONCLUSION

We discuss the importance of effectively communicating the risk of an application to users, and propose several methods to rate this risk. We test these methods on large real-world data sets to understand each method's ability to assign risk to applications. One effective method is the RSS method which has several advantages. It is monotonic, and can provide feedback as to why risk is high for a specific app and how a developer could reduce that risk. It performs well in identifying most current malware apps as high risk. This method allows for highly critical permissions and less-critical permissions to affect the overall score in an easy to understand way, making it more intuitive as well as difficult to evade when compared with other models.

ACKNOWLEDGMENTS

It gives us great pleasure in presenting the Result Paper on 'Mobile App Classification based on risk Score.'

I would like to take this opportunity to thank my internal guide Prof. J. V. Shinde for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their Valuable suggestion were very helpful

REFERENCES

[1] Google Bouncer. http://goo.gl/QnC6G 2014.

[2] N. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes versus Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing, pp. 420-424, 2004.

[3] K.W.Y. Au, Y.F. Zhou, Z. Huang, and D. Lie, "PScout: Analyzing the Android Permission Specification," Proc. ACM Conf. Computer and Comm. Security (CCS '12), pp. 217-228, 2012.

[4] 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," Proc. 17th ACM Conf. Computer and Comm. Security, pp. 73-84, 2010.

[5] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, 2007.

[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," J. Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[7] S. Chakradeo, B. Reaves, P. Traynor, and W. Enck, "MAST: Triage for Market-Scale Mobile Malware Analysis," Proc. Sixth ACM Conf. Security and Privacy in Wireless and Mobile Networks (WISEC '13), pp. 13-24, 2013.

[8] E. Chin, A.P. Felt, V. Sekar, and D. Wagner, "Measuring User Confidence in Smartphone Security and Privacy," Proc. Eighth Symp. Usable Privacy and Security, (SOUPS '12), article 1, 2012.

