Android App For Providing Security With GP-PP Model Using Decision Tree Algorithm

Rajeith Thanumalaya Perumal Pillai[1]**, Vimal kirubashankar**[2]

[1, 2]Anna university

Abstract- *The interface allows the user to search for an arbitrary application on the Play Store; the permissions list and the privacy policy are then automatically retrieved, whenever possible. The user has then the ability of selecting a specific permission and a list of relevant sentences are extracted by the privacy policy and presented to them, along with an accurate description of the permission itself. Such an interface allows the user to quickly evaluate the privacy-related risks of an Android application, by highlighting the relevant sections of the privacy policy and by providing useful information about sensible permissions. We presented a novel approach to the analysis of privacy policies in the context of Android applications. The tool we implemented greatly eases the process of understanding the privacy implications of installing third party apps and it has already been proven able to highlight worrisome instances of applications. The tool is developed with expandability in mind, and further developments in the approach can easily be integrated in order to increase the reliability and effectiveness. Refer to the "Personal and Sensitive Information" section if your app handles personal or sensitive user data. The applicable privacy or data protection laws are prescribed by google play requirements. We proposed, a user who wishes to install and use any third party app doesn't understand the significance and meaning of the permissions requested by an application, and thereby simply grants all the permissions as a result of which harmful apps also get installed and perform their malicious activity behind the scenes. The user's inability to analyse the risk of any application results in compromised security and privacy.*

Keywords- KNN Algorithm,GP-PP Model, Decision Tree Algorithm,Machine Learning, Android Apps

I. INTRODUCTION

The emerging paradigm of Internet of Things (IoT) facilitates the exchange of data and information among connected devices and systems. As an ongoing trend, mobile devices have become the major platforms of IoT applications for industrial enterprises. The dynamic nature of IoT applications demands reliable and consistent services. For instance, the Open Services Gateway Initiative (OSGi) platform based on a dynamic Service-Oriented Architecture (SOA) can enable and deploy smart services such as mobile apps in industrial contexts. With the emergence of a connected world, mobile phones are playing a major role in facilitating the applications for various settings of industrial IoT. As the mobile phones in the modern world are enabled with sensors, it is now possible for a common user to easily capture and share all kinds of information. Android apps have become increasingly popular among all the mobile platforms. In the first quarter of 2017, the market share of Android-based phones was 85%, whereas iOS phones captured 14.7% of the market, Windows Phones owned 0.1% of the market and 0.1% by all other platforms. The popularity of Android phones is due to a huge gamut of apps available on Google play. While some of the apps are freely downloadable, others are paid. While installing the apps, users are supposed to grant a huge list of permissions, out of which many permissions are not required and are 'dangerous'. Usually, users do not read the entire permission set requested before installing the app.

The GP-PP (Generic Permissions-Privacy invasive permissions) model is useful to classify the permissions into Generic and Privacy invasive permissions. The model proposes a simplistic way for users to decide which apps are dangerous to install. Based on the permission set that a particular app requests, the GP-PP model classifies an app as privacy-invasive if the majority of the permissions requested are privacy-invasive. So, users can decide which set of permissions can be harmful. Therefore, by looking at the set of permissions requested, a user can determine whether an app is privacy invasive or not. In other words, a user can determine whether it is safe to install the said app or not. Although the proposed GP-PP model seems valuable, its efficacy has not been fully assessed. To address this gap, we test the effectiveness of the GP-PP model using Naïve Bayes Classifier in this paper. In particular, we validate the GP-PP model in order to verify whether the model classifies an app on the basis of permission sets that the app requests. Our study confirms that the greater the number of Privacy-invasive permissions that an app requests, the more dangerous it is to install the app. In summary, the major contribution of the paper is the validation of the proposed GP-PP model using

Naïve Bayes Classifier and decision tree algorithm for popularity.

II. RELATED WORKS

The learning and detection stages are taken from Bayesian-based classifiers. The learning stage uses a training set of known malicious samples in the wild and another set of benign Android applications, collectively called the app corpus Empirical results and comparative analysis are presented offering useful insight towards development of effective static-analytic Bayesian classification-based solutions for detecting unknown Android malware[1].To achieve this, static analysis involves various binary forensic techniques, including decompilation, decryption, pattern matching and static system call analysis. Analysis of Permission Based Risk signals. In order to get a baseline, we first apply the only other mechanism that has been published to identify risk based on permissions, namely Kirin [2]. It adapts to effectively localize and detect possible changes from app repackaging using fuzzy hashing technique. The results call for the need of a rigorous vetting process for better regulation of third-party smartphone application marketplaces [3]. We extract the minimal path (using the Dijkstra's algorithm) as a chain of events, which are sequentially triggered in the symbolic execution. In this section, we present our evaluation results on the effectiveness and accuracy of AppIntent. In our evaluation, the event-space constraint guided symbolic execution uses an Intel Xeon machine with 2 eight core 2.0Ghz CPUs and 32 GB physical memory, which runs Debian Linux with kernel version 2.6.32. The controlled execution of App Intent is run on Android 2.3 [5]. The Android platform-based individual privacy information protection system architecture and the key execution techniques. The system could satisfy user functional and non-functional requirements, with stable operation and high task execution efficiency [6]. We developed tools to detect over privilege in Android applications. To find the permissions required to invoke each API method we applied automated testing techniques to Android 2.2. Our results show that applications generally are over privileged by only a few permissions, and extra permission can be attributed to developer confusion. This indicates that developers attempt to obtain least privilege for their applications but fall short due to API documentation errors and lack of developer understanding [7]. Our results suggest that our interface significantly increases users' privacy awareness and is easier to comprehend than Android's current permission interface [8]. A signature inference algorithm to generate the signatures of the methods and fields implemented in the application. The experimental results outline a frustrating phenomenon: a large fraction of mobile applications seem to abuse user's privacy by sending the data collected on a smartphone to advertising companies without notifying the smartphone's owner [9]. In this system, we are using Youmi Encoding techniques. We believe that a corresponding decoding algorithm could be created by reversing the encoding algorithm. We proposed potential solutions to several common ad library privacy vulnerabilities, including the failure to protect user data in ad requests, mishandling of UDIDs, and the lack of privilege separation between application and ad code on Android [10].

III. ALGORITHM

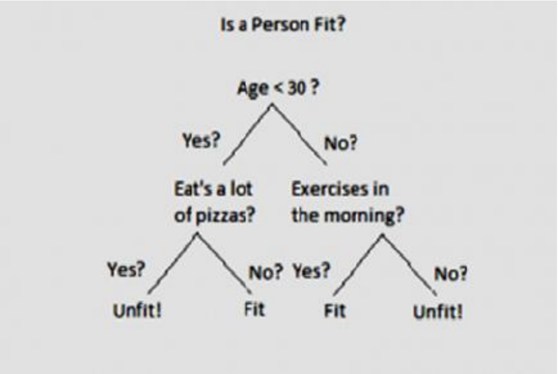
A Machine Learning Algorithm

Decision Trees for Classification

Decision Trees square measure a sort of supervised Machine Learning (that is you justify what the input is and what the corresponding output is within the coaching data) wherever the information is continuously split according to a certain parameter.

The trees are often explained by 2 entities, specifically call nodes and leaves. The leaves square measure the choices or the ultimate outcomes. And the call nodes square measure wherever the information is split.

In supervised learning, the algorithm decision tree is one of its parts. It is also useful for solving regression and classification problems. Training model has been created to predict class and value of target variables by learning decision rules inferred from training data. In classification and regression, the cost function is used to find the homogeneous branches or group of branches having the same response.



Machine Learning Based Classification of Android Apps

Machine learning and text mining methods are used to classify the Android mobile apps. Our approach consists in applying some machine learning methods on text characteristics that are extracted from app’s description on Google Play Store. Our proposed approach consists of two main phases. First we collect information about apps from the Google Play store using a web crawler. Then we extract some text information from this data. In our case, for each app we have extracted its description and its category

IV. EXISTING SYSTEM

The GP-PP (Generic Permissions-Privacy invasive permissions) model is useful to classify the permissions into Generic and Privacy invasive permissions. The model proposes a simplistic way for users to decide which apps are dangerous to install.

Based on the permission set that a particular app requests, the GP-PP model classifies an app as privacy-invasive if the majority of the permissions requested are privacy-invasive. So, users can decide which set of permissions can be harmful.

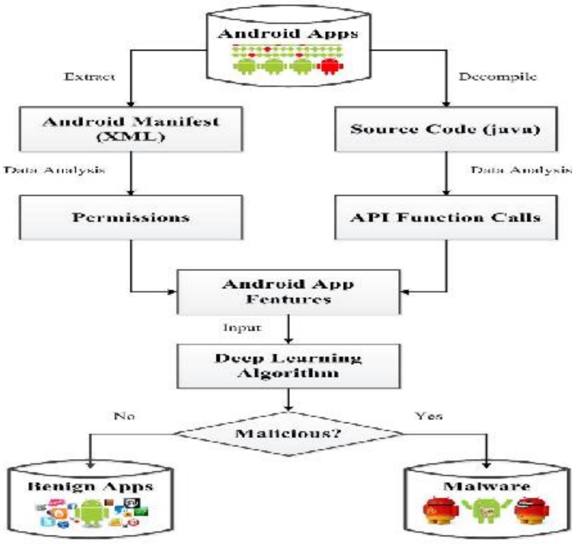
We validate the GP-PP model in order to verify whether the model classifies an app on the basis of permission sets that the app requests.

DISADVANTAGE

Using Naive Bayes algorithm alone the security and efficiency of detecting the malicious app is low compared to the proposed system.

Another problem happens due to **data scarcity**. For any possible value of a feature, you need to estimate a likelihood value by a frequentist approach. This can result in probabilities going towards 0 or 1, which in turn leads to numerical instabilities and worse results.

V. SYSTEM ARCHITECTURE



VI. PROPOSED SYSTEM

Identify the list of third party installed applications. Extract the complete list of permissions of each application. Identify android: protection level of each permission i.e., Normal or Dangerous of each app Take android app permissions dataset. To identify the harmful applications, apply classification algorithms. Note the accuracy of spam classification given by it and time required for execution Results as accuracy among different harmful and normal apps Classifiers are analysed

ADVANTAGES

- Security is high.
- We can easily find the less used applications.
- It is simple to uninstall the application which is affected.

VII. MODULE DESCRIPTION

Android Application Scan

Malware of this kind can’t be detected by victimization the quality signatures approach or by applying regular static or dynamic analysis strategies. The detection is performed on the basis of the application's network traffic patterns solely. For each application, a model representing its specific approach pattern is learned regionally (i.e., on the device). Semi-supervised machine-learning methods are used for learning the normal behavioural patterns and for detecting deviations from the application’s expected behaviour. These methods were implemented and evaluated on Android devices.

Malware Detection

The task of identifying malware will be categorised into analysis, classification, detection and ultimate containment of malware. Several classification techniques are utilized in order to classify malware in keeping with their instances and this has created its potential to acknowledge the sort and activities of a malware and new variants. Analysis of malware has got to do with distinguishing the instances of malware by completely different classification schemes by exploiting the attributes of well-known malware characteristics. Malware detection has to do with the quick detection and validation of any instance of malware in order to prevent further damage to the system. The last part of the work is containment of the malware, which involves effort at stopping escalation and preventing further damages to the system. A commercial antivirus uses signature based mostly technique wherever the information should be often updated so as to possess the newest virus information detection mechanisms. However, the zero-day malicious exploit malware can not be detected by antivirus, supported signature-based scanner, however the utilization of applied math binary content analysis of file to detect abnormous file segments.

Android Permission Check

A user who wishes to install and use any third-party app doesn't understand the significance and meaning of the permissions requested by an application, and thereby simply grants all the permissions as a result of which harmful apps also get installed and perform their malicious activity behind the scene.

Privacy Policy Database

We presented a novel approach to the analysis of privacy policies in the context of Android applications. The tool we implemented greatly eases the process of understanding the privacy implications of installing third party apps and it has already been proven able to highlight worrisome instances of applications. To find your apps and their permissions on mechanical man. open the settings and so faucet Apps & notifications, App info, and also the app you are inquisitive about. Select the Permissions entry to see all the privileges the app Uninstall.

VIII. CONCLUSION

Mobile devices have become the major platforms of IoT applications for industrial enterprises. The ever-increasing number of Android Phone users has created great interests regarding the privacy issues of Android Apps. Users typically do not either read the permissions that an app requests or are unable to judge the app on the basis of permissions requested. In this research we address the privacy issues by categorizing the app permissions into Privacyinvasive and Generic permissions and validating the classification using Naïve Bayes Classifier. Our results indicate that as per the classification done by the GP-PP model and validated through the Naïve Bayes Classifier, users can decide upon which apps are safe to install and which apps are not depending on the permission set that the app is requesting. The permissions that fall into Privacy Invasive Class are Your personal information, Camera, Microphone, Reading interaction info, Bluetooth and Your location.

REFERENCES

[1] Kesswani, N., & Lin, F. (2016, March). How privacy invasive Android apps are? In Computing for SustainableGlobal Development (INDIACom), 2016 3rd International Conference on (pp. 3731-3734). IEEE.

[2] Kelley, P. G., Consolvo, S., Cranor, L. F., Jung, J., Sadeh, N., &Wetherall, D. (2012, February). A conundrum of permissions: installing applications on an android smartphone. In International Conference on Financial Cryptography and Data Security (pp. 68- 79). Springer Berlin Heidelberg.

[3] Sarma, B. P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., & Molloy, I. (2012, June). Android permissions: a perspective combining risks and benefits. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies (pp. 13-22).ACM.

[4] Zhou, W., Zhou, Y., Jiang, X., & Ning, P. (2012, February). Detecting repackaged smartphone applications in third-party android marketplaces. In Proceedings of the second ACM conference on Data and Application Security and Privacy (pp. 317-326).ACM.

[5] Chia, P. H., Yamamoto, Y., &Asokan, N. (2012, April). Is this app safe? A large-scale study on application permissions and risk signals. In Proceedings of the 21st international conference on World Wide Web (pp. 311-320).ACM.

[6] Yang, Z., Yang, M., Zhang, Y., Gu, G., Ning, P., & Wang, X. S. (2013, November Appointment: Analysing sensitive data transmission in android for privacy leakage detection. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (pp. 1043-1054).ACM.

[7] Yerima, S. Y., Sezer, S., & McWilliams, G. (2014). Analysis Of Bayesian classification-based approaches for Android malware detection. IET Information Security, 8(1), 25-36.

[8] Zhang, W., Li, X., Xiong, N., &Vasilakos, A. V. (2016). Android platform-based individual privacy information

protection system. Personal and Ubiquitous Computing, 20(6), 875-884.

[9] Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011, October). Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (pp. 627-638). ACM.

[10] Lin, J., Amini, S., Hong, J. I., Sadeh, N., Lindqvist, J., & Zhang, J. (2012, September). Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing (pp. 501-510). ACM.

[11] Mann, C., &Starostin, A. (2012, March). A framework for static detection of privacy leaks in android applications. In Proceedings of the 27th annual ACM symposium on applied computing (pp. 14571462).ACM.

[12] Stevens, R., Gibler, C., Crussell, J.Erickson, J., & Chen, H. (2012, May). Investigating user privacy in android ad libraries. In Workshop on Mobile Security Technologies (MoST) (p. 10).