A Supervised Machine Learning Approach for Malware Detection

Ankita Sharma*, Rohan Gupta

Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India
Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India

ABSTRACT

There is Explosive increase in mobile application more and more threat, viruses and benign are migrate from traditional PC to mobile devices. Existence of this information and access creates more importance which makes device attractive targets for malicious entities. For this we proposed a probabilistic discriminative model which has regularized logistic regression for android malware detection with decompiled source code. There are so many approaches for detection of android malware has been proposed by using permission or source code analysis or dynamic analysis. In this survey paper, we use a probabilistic discriminative model for detection of malware by using supervised method. It also shows that probabilistic model based on regularized regression also it works well with permission. These three tools will give us complete part of analysis for the existing system which will gives us a complete part of analysis for exiting system. From this tool we get source code patterns. These patterns are studied and network database is train for getting malware and normal app signature.

KEYWORDS: Android, malicious application, machine learning, discriminative model, dataset.

INTRODUCTION

The Number of Android mobile devices has been increased in recent year. There are so many approaches for detection of android malware has been proposed by using permission or source code analysis or dynamic analysis. In this survey paper, we use a probabilistic discriminative model for detection of malware by using supervised method .It also show that probabilistic model based on regularized regression also it works well with permission. Furthermore this first research model achieves the best detection result code and application permission. In the Previous work limited tool were used for analysis of the APK file, we will be using three of the most advanced tool for APK analysis namely 1.dex2jar, 2. Java Decompiler, 3.class file analyzer. These three tools will give us complete part of analysis for the existing system. In previous method utilizes training data to build probabilistic model under assumption that the testing data, which is the model will applied to drawn from the same. But these types of assumption may or may not be same in reality. Since malicious application may evolves. So model need to be updated with new training data and adding new benign application and new malicious application. To overcome these disadvantages we proposed system in this paper for malware detection as a supervised learning method.

LITERAURE SURVEY

In[1] paper, we propose a probabilistic discriminative model based on regularized logistic regression with decompile source code for android malware detection which can generate most accurate detection result than previous result with application permission or with source code. But some limitation has been accord. As a statistic technique it shares the weakness of all static technique on android application. Moreover dead code could cause trouble to our method, leading to unreliable feature extraction and representation. Android limitation is introduced by the nature of supervised learning. In this paper it utilizes training data to build probabilistic model by taking assumption that the testing data. So this assumption may not be true in reality since malicious application may revolve. So to overcome this limitation in proposed system. In [2] paper, it based on home-brewed cloud computing platform and data mining. It stated a methodology to evaluate mobile apps for improving the current status security of mobile apps, mob safe, a demo & prototype system. So there are not uses any permissions or source code for identify the apps are virulence or benignancy. We overcome these limitation by using machine learning to identify mobile apps based on data mining. In [3] paper, it approach toward android operating system and aim to detecting existing android malware. It chooses some malware from dataset that contains 104 malware samples and try to analyze them by understanding their installation method and activation. This model gives detail study of android architecture and its security model. It also uses many tools like APK tool, Dex2jar, java decompiler, android box. Finally it concludes that, all functionality of mobile device is very attractive target for attacker to gain information of user & used it for his/ her benefit. So, user has to aware and keep full responsibility to read and understand permission requested by the application before agree to grant access. In [4] paper, It proposed ADAM an automated and extensible system that can evaluate by large scale stress test. It is an automated generic and extensible platform that evaluates the detection of android malware detection system. ADAM uses

difference transformation techniques involving repackaging and code obfuscation which generate different variant. So ADAM mainly focus on generating malware threat &examine the effectiveness of malware detection in android Smartphone. But there are several issues that not addressed in this work. One is signature coverage that was not accurately inferring the underlying signature being use by the system. Second is defense solution that can defend against obfuscation and repacking techniques. In this signature is finding out and classify the pattern but not identify the pattern which is malware or spyware or safe application. This deficiency will implement in proposed system In [5] paper, the research presented is an attempt to analyze malware behavioral by using two phase as code review and live testing by suggest security techniques not currently found in Android-based devices. In first phase, malware sample are retrieved from security repository and classify into categories like Trojan, warm, spyware etc. This sorting help in the analysis of categorizing malware which may share possible behavior. In next phase malware run on device emulator to observe malware interaction with device and user. It is use for solution aim to increase confidentiality, integrity &availability of system by improving isolation of data and also protecting access to data. It is exploit the vulnerability of the mobile operating system.

PROBLEM DEFINITION AND RESEARCH METHODOLOGY

Problem Definition

The existing method utilizes training data to build probabilistic model, under the assumption that the testing data, which model will be applied to, is drawn from the same population as the training data. This assumption is generally not true in reality since malicious application may evolve. Hence the model needs to be updated with new training data including new benign applications and new malicious applications. So proposed system used training as well as testing data and after it gives to probabilistic discriminative model for malware detection with decompile source code. So our aim to improves delay for malware detection and accuracy of malware identification.

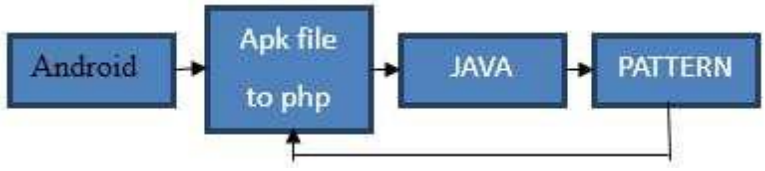
Research Methodology

It works on improved pattern analysis or in-depth analysis of signature patterns. In existing system used only two tools which not give complete analysis. So proposed system used training as well as testing data and after it gives to probabilistic discriminative model for malware detection with decompile source code.

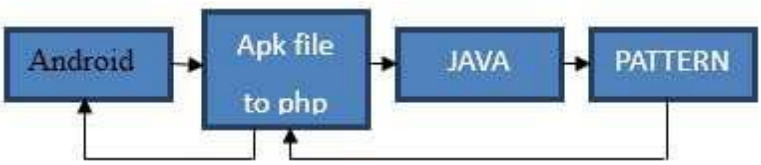
PROPOSED SYSTEM

In the existing system work has been done either on analysis of signature patterns of the apk file or on the permissions requested by the application. But none of these researches focus on the patterns along with the permissions. Secondly, none of the researches work on improved pattern analysis or in-depth analysis of signature patterns. This reduces the efficiency of the malware detection system, and also makes the APK files which have malwares to install easily on the Android phones. In the previous work limited tool were used for analysis of the APK file, Our approach uses a 3 step decompilation strategy which is used most advance tools for APK analysis namely, 1. Dex2jar, 2. Java decompiler, 3. Class file analysis Where, we first use a dex2jar tool to find out the encrypted jar files, then a java decompiler is used to get the .class files, once the class files are obtained we use a class to java converter to get the original source code patterns. These patterns are studied and the network database is trained for getting malware and normal app signatures. This approach improves the speed of the system, and also improves the overall quality of malware detection and identification.

1. Training



2. Testing and Evaluation



From Fig1.first develop Android app for training the data, then android extract .apk file from application and gives to PHP. PHP give to java for detection of pattern. These pattern save to the PHP and retrieve it as malware,

spyware or normal signature app. Fig.2 In testing same process are used as in training. Pattern which is save in PHP it give to android for compare to find .apk type as malware, spyware or normal signature to application.

SIMULATION RESULT

Table : Calculation of Accuracy and Delay time

No. Of Entries in DB	No of APKs tested	Correctly classified	Accuracy	Delay
10	5	4	80	15
10	10	9	90	25

In table show calculation of Accuracy and delay time. After that we calculate accuracy by following formula.

$$Accuracy = \frac{\text{Correctly Classified}}{\text{No.of APKs Tested}} \times 100$$

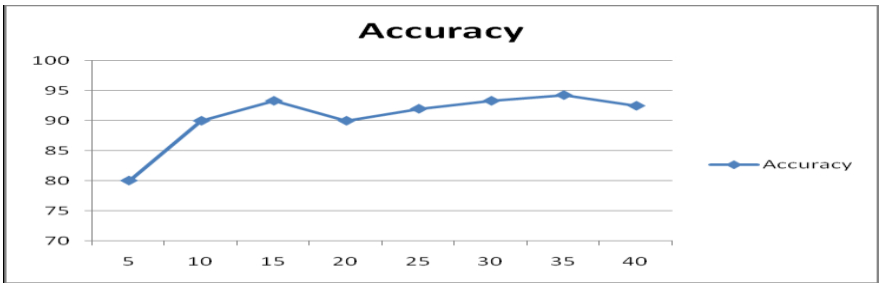


Figure : Accuracy result with no. of APK tested.

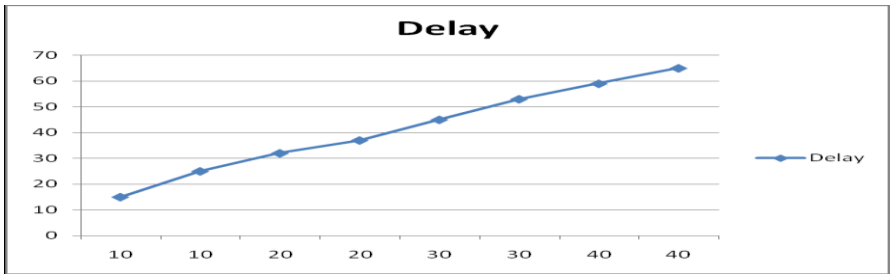


Figure: Show the delay with respect to No. entries in DB.

CONCLUSION

In this paper proposes to use machine learning algorithm and decompiled source code and beyond. It uses machine learning, which is focus on the pattern along with the permission and work on improved pattern analysis or in-depth analysis of signature pattern. So proposed system used training as well as testing data and after it gives to probabilistic discriminative model for malware detection with decompile source code. To implements models of project we using SVM (support vector machine) are supervised learning models with associated learning algorithm that analyze data for classification and identification. This approach improves the speed of the system and also improves the overall quality of malware detection and identification.

REFERENCES

[1] Nisha Badwaik, Prof. Vijay Bagdi. "A SURVEY ON SUPERVISED METHOD FOR DETECTION OF MALWARE". In *International Journal Of Engineering Science And Research Technology.* ISSN (Online): 2277-9655 Impact Factor: 4.116, Vol. 5, Issue 6, June 2016.

[2] Lei Cen, Christoher S. Gates, Luo Si, and Ninghui Li, Senior Member, IEEE. "A Probabilistc discriminative model for android malware detection with decompiled source code" JULY/AUGUST 2015.

[3] JianlinXu, Yifan Yu, Zhen Chen, Bin Cao, Wenyu Dong., Yu Guo, and Junwei Cao. "Mobsafe:cloud computing based Forensic analysis for massive mobile application using Data mining ".

[4] Khalid alfaqi,Rabayyi alghamdi,mofaresh waydom."Android platform malware analysis", International journal of advance computer science & application, vol.6, no.1, 2015.

[5] Minzheng, pratick p.c. Lee, and John C.S Lue,"ADAM:An Automatic & extension platform tostress test android antivirus system", Dept of CSE.
[6] Kriti Sharma, Trushank Dand ,taeoh & William Stake pole "Malware analysis of android operating" 8th anal symposium on information assurance (ASIA113), June 4-5, 2013,ALBANY, NY.
[7] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in Proc. 16th ACM Conf. Comput. Commun. Security, 2009, pp. 235–245.
[8] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android permissions: A perspective combining risks and benefits," in Proc. 17th ACM Symp. Access Control Models Technol., 2012, pp. 13–22.
[9] H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita- Rotaru, and I. Molloy, "Using probabilistic generative models for ranking risks of Android apps," in Proc. ACM Conf. Comput. Commun. Security, 2012, pp. 241–252.
[10] A. P. Felt, E. Chin, S. Hanna, D. Song Wagner, "Android permissions demystified," in Proc. 18th ACM Conf. Comput. Commun. Security, 2011, pp. 627–638.