International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

An Innovative Technique to Detect Malicious Applications in Android

Sharvari Prakash Chorghe[1], Dr. Narendra Shekokar[2]

[1]Computer Department, D J Sanghvi College of Engineering, Vile Parle, India
[2]Computer Department, D J Sanghvi College of Engineering, Vile Parle, India

Abstract: ***Android Operating System is gettingpopular withmajority of users of the entire genre. Ever increasing mobile malware is a major threat to Android OS. Hence it is essential to add an extra security layer to android smart phone. Various malware detection techniques use Mutual Information or Information Gain for feature selection. Mutual Information calculates relevance of a feature and the class variable. Feature selection or ranking is crucial step before applying any machine learning classification algorithm on the pre-processed data. The proposed system uses improved mutual information feature selection (IMIFS) method to maximize relevance and minimize redundancy in the selected features. This improves the accuracy of the classification model. The proposed system will use features from manifest.xml as well as application code for malware detection. Bayesian Classification is used to develop classifier model to detect malicious applications before installation on the device.***

Keywords: Android malware detection; improved feature selection;bayesian classification; data mining; machine learning

1. Introduction

Malicious applications targeting smart phones with Android operating system is increasing rapidly. Android has been vulnerable to malware because it is open source and many devices still run on older version of Android operating system. Android operating system had added 400 million users since May 2014, for a total of 1.4 billion [2]. Android by now has become one of the most popular operating system for smart phones.

One in every Six Android applications is a malware, according to the study by a security company Symantec. [1]According to the research conducted by Symantec on 6.3 million Android applications 17% of all the apps come under classification of malware. The study found that nearly one million malwares are created every day across the globe. Around 317 million malwares were created in 2014.There was 26% increase in creation of malware from 2013[1]. Application is a very common way by which operating system and data stored in it get compromised. Users get attracted to third party applications and thus are vulnerable to malware installation.

There are different types of android malwares- Adware that continuously displays ads as urgent notification, Spying Tool sends GPS location, tracks text messages and call log, Rooter allow attacker to send command to the affected device, Data Stealer steals data from victims device and send it to attacker, Premium Service Abusers subscribes infected phones to premium services without permission, and Malicious Downloader downloads malware on infected device[3].

There is a broad category of malware detection techniques: Anomaly based detection, Special type of Anomaly based detection is Specification based and Signature based Detection[4].Anomaly based detection is based on the knowledge of the technique as to what constitutes as benign behavior to identify malicious application. Specification based techniques define certain specifications or rules for normal behavior to determine suspicious applications. Signature based techniques use definition of known malwares for detection of malicious application.

The existing commercial software solutions use signature based detection as it is easy to implement. Signature based methods can easily be cracked by the malware developers. Few solutions use pattern matching approach which can be circumvented easily by the attackers once the pattern is acknowledged by them. Researchers have proposed dynamic malware detection techniques that analyze network traffic or frequency of remote call made to the server. The drawback of these systems is that it requires the application to be installed on the device and continuous monitoring of the behavior of the application once it is installed on the device. Some techniques employ static analysis of permissions and code properties. But very few of them have concentrated on Feature selection technique. Feature Selection improves the overall performance of the classifier.

The proposed system detects malware based on the features extracted from manifest.xml file and code properties from the android application package (APK) of the application. It focuses on improving feature ranking technique to decrease the error rate due to empty feature vectors. The system uses improved mutual information based feature selection/ranking method [5] to improve the detection efficiency of malware detection by maximizing the relevant feature selection and minimizing selection of redundant features. After feature selection Bayesian classification is used to classify testing dataset as benign or malicious application.

In machine learning, feature selection is used to select relevant features for construction of classification model. Feature Selection help to select features that can get accurate results. There are three categories of feature selection algorithm: Filter Method, Wrapper Method and Embedded Method [10]. Best features are selected in Filter method using predefined criteria without recounting to any classification algorithm. In wrapper method different

subsets are evaluated using a predictive model and the subsets are compared based on the results of previous subsets. In embedded systems independent measure is used to select the subsets and machine learning algorithm is used to find the best subset.

The rest of this paper is organized as follows: Section 2 describes the related work of Android malware detection. The proposed system is described in Section 3. Finally, we conclude our work in Section 4.

2. Related Work

Malware detection techniques are of two types Static and Dynamic. Static Malware detection is extraction of features from apk file of application to be installed and then apply classification algorithm on extracted data. Dynamic Malware detection includes online analysis of installed applications.

In [6] Stephen Feldman proposed a tool Manilyzer that disassemble APK files and generate feature vectors of each application characteristic. Characteristics that were observed from malicious applications manifest.xml file were certain permissions were requested only by malicious applications, Request to intent filter with high priority may be malicious and low version numbers. Manilyzer used Weka Tool to implement multiple machine learning algorithms, including Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbors algorithm (KNN), and C4.5 decision tree algorithm, for classification. Dynamic analysis was proposed in this model to analyze network traffic using Wireshark and android Emulator. Three malicious applications -HGSpy, Simplocker, and Minimob were examined using this approach.

In [7] Zhao Xiaoyan proposed malware detection framework that extracts features from manifest.xml file. The framework has mainly following modules:

Pre-processing, Feature Selection, SVM Classifier. In Pre-processing android apk file is decompressed to extract permissions to form feature vectors. The Feature Selection module uses Principal Component Analysis Algorithm. SVM Classifier is trained using training dataset; the trained classifier can detect malicious applications in testing phase.

Quentin Jerome[8] proposed a feature based malware detection tool based on op-code sequences. All possible k-grams were collected from the application's classes.dex file. In this approach op-code were used without operands as they can be changed without altering the semantics of code. Relevant features are selected by calculating information gain for each feature. Linear implementation of SVM was used for classification.

Droid Detective proposed by Shuang Liang and Xiaojiang Du [9] generates rule sets based on Permission Combination. The system identified that certain rule combination were requested by malware applications but not by benign applications.

Dynamic malware detection technique was proposed by Anshul Arora et al. [10] using network traffic analysis. Average Packet size, duration of the flow, time interval between packet sent, Ratio of incoming to outgoing bytes, Ratio of incoming to outgoing packets such features were analysed to generate a rule based model for identification of malicious applications. Seven distinct features were identified in malware traffic.

Various better feature selection techniques have been proposed to reduce the error rate of the classifier.

The authors of [11] proposed a method, minimum redundancy maximum Relevance for feature selection using average of the feature-feature mutual information. The proposed formula is given by

$$f\ mRMR = I(Ci, Xi) - \frac{1}{|S|}\sum_{Xes} I(Xs, Xi)$$

La TheVinhet al. [12] proposed normalized feature selection algorithm. The formula proposed by the author for feature ranking is given by,

$$f*(Xi) = \hat{I}(C, Xi) - \frac{1}{|S|}\sum_{Xes} \hat{I}(Xs, Xi)$$

Where $\hat{I}(Xs, Xi) = \frac{I(Xs,Xi)}{\min\ (H(Xs),H(Xi))}$.

It was observed that very few researchers have focussed on feature selection algorithm. Feature selection leads to reduction in features to get relevant subsets of features. Mutual Information or Information Gain is used by most of the authors for feature selection. Mutual Information calculates the relevancy of the features with the class variable. It is necessary to calculate redundancy among the features to get better output of the classifier. Hence the proposed system uses improved feature selection algorithm that uses conditional mutual information to calculate redundancy among the features. Various techniques use only permissions from manifest.xml file. The proposed system uses features from both manifest and code for malicious application detection.

It was observed that very few researchers have focussed on feature selection algorithm. Feature selection leads to reduction in features to get relevant subsets of features. Mutual Information or Information Gain is used by most of the authors for feature selection. Mutual Information calculates the relevancy of the features with the class variable. It is necessary to calculate redundancy among the features to get better output of the classifier. Hence the proposed system uses improved feature selection algorithm that uses conditional mutual information to calculate redundancy among the features. Various techniques use only permissions from manifest.xml file. The proposed system uses features from both manifest and code for malicious application detection.

3. Proposed System

The main initiative of the proposed system is to improve the accuracy of malware detection using any machine learning algorithm for classification. The system consists mainly of three modules Feature Extraction, Feature Ranking/Selection, Classifier.

A. Feature Extraction

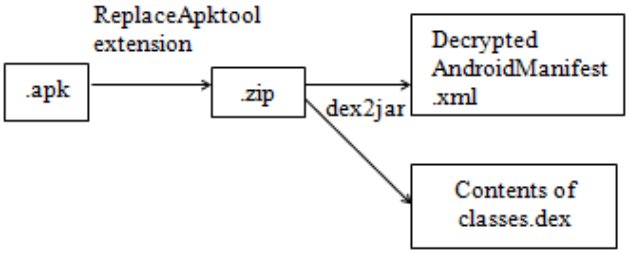

Figure 1: Steps of Feature Extraction

There are two phases of feature extraction. Features are extracted from Manifest.xml and classes.dex files of apk file of Android application. Initially for any application to be decompiled the .apk extension of a file has to be renamed to .zip. Then the contents of the file can be extracted. To retrieve the content from AndroidManifest.xml file third party reverse engineering tool Apk tool is used. To get the contents of classes.dex file dex2jar tool is used to get .jar files from the file with .dex extension.

- Permission Extraction from Manifest.xml

In Manifest.xml <USES-permission> tag is used to declare the permission user has to accept before installing the application.

The training dataset must be divided into malicious and benign applications.137 standard android permissions must be parsed and the count must be maintained separately for malicious application dataset and benign application dataset.

- Code properties extractions from classes.dex

The code based features that can be recorded areAndroid and Java API calls, Linux system commands and some Android-based commands and notifications.58 code based properties can be used for feature extraction. The count is to be recorded for these features for malicious and benign applications.

B. Feature Ranking/Selection

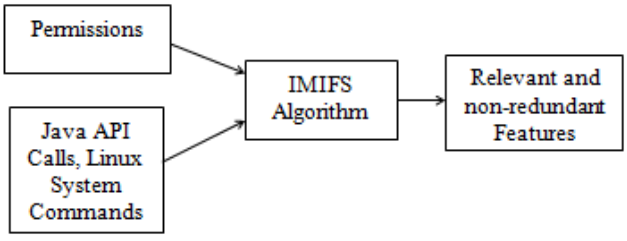

Figure 2: Steps of Feature Ranking

Feature selection has a major impact on accuracy of the classifier. If irrelevant features are selected it will affect the efficiency of the system. Mutual Information can be used to measure how much a feature tells about the class i.e. Benign or Malicious. Mutual Information is used to calculate relevant features from the candidate features. Condition Mutual Information is used to calculate redundancy among the selected features. Bayesian Classifier requires features to be independent to each other. Thus proper feature selection technique can improve the overall performance of the classifier. Hence the proposed system uses improved mutual information feature selection (IMIFS) algorithm [3] for appropriate selection of features. IMIFS uses Mutual information and Conditional mutual information to select relevant and non-redundant features respectively.

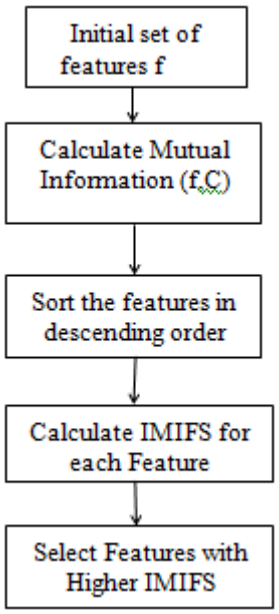

Figure 3: Improved Mutual Information Feature Selection Algorithm (IMIFS)

Mutual information is measure that indicates how much a feature conveys about the class benign or malicious. The mutual information between the feature fi and class C can be calculated using [5]

$I(X, Y) = -$ $\sum_{x \in X} \sum_{y \in Y} p(x,$.

The count recorded in feature extraction process is used here to calculate mutual information in above formula.

The system uses Conditional mutual information to get non-redundant features in feature vector.

Conditional Mutual information (CMI) is defined as

CMI (fs,fi |C) = H (fs|C) H (fs|fi, C).

Mutual information between a feature f and class variable C is the reduction in uncertainty of feature f due to the knowledge of class variable C.

Entropy is measure of uncertainty of any random variable X given by, [10]

H(X) =

Conditional entropy calculates the amount of information required to describe the outcome of random feature f given the value of class C as benign or malicious. [5]

H (f|C) = $\sum_{f\epsilon F}\sum_{c\epsilon C}$

IMIFS= $\frac{2*I(fi,C)}{H(fi)+H(C)} - \frac{1}{|S|}\sum_{fs\epsilon S}\frac{2*CMI(}{H(fs/C)}$

The input to IMIFS algorithm [5] is dataset of n features and class variable C=Benign or Malicious. The output is S i.e. selected features.

Top n features are to be selected for construction of feature vectors for testing application dataset.

C. Classifier

Bayesian Classifier is used to classify the applications as benign or malicious. Bayesian Classifier is used because of its firm mathematical basis and stable classification efficiency [11].Bayesian classifier assumes strong independence among the features. The classifier will calculate the probabilities of each application feature vector for each class. Application will be classified based on the probabilities calculated. Classifier will be built in two phases. Learning phase uses training dataset and detection phase uses testing dataset. According to Bayes theorem the probability of application being benign or malicious Class(C) can be calculated using formula.

P(C=c|F=fs) = $\frac{P(C=c)\prod_{s=1}^{s}P(F}{\sum_{i\epsilon\{0,1\}}P(C=ci)\prod_{s=1}^{s}}$,

Where P(F=fs|C =ci) and P(C=ci) are the probabilities calculated on learning dataset. Heresis the number of features used by the classifier; c is considered 0 for benign class and c=1 for suspicious class.

Every new app to be classified is represented by the vector

f(f1, f2,...fn). The app is classified as benign if the probability of the app belonging to benign class is more compared to suspicious.

P(C=benign | F=fs)>P(C=suspicious | F=fs)

Otherwise, it is classified as suspicious.

D. Expected Results

The proposed model is expected to detect zero day attack by calculating the Bayesian probability. The model uses improved feature selection technique to improve the accuracy of the system. Improved feature selection is expected to reduce the number of empty feature vectors that results in increased error rate.

4. Conclusion and Future Work

The proposed system focuses on feature selection to improve the performance of the selected classifier. Bayesian classifier is suitable for this approach as output for large dataset can be estimated faster. The implementation of Bayesian Classifier is easier compared to other machine learning algorithms. In future Weka tool can be used to test the selected dataset on other machine learning algorithms.

References

[1] Times of India.“Android malware study”,[onlie]:http://timesofindia.indiatimes.com/tech/tech-news/One-in-six-Android-apps-is-a-malware-says-study/articleshow/47021326.cms
[2] Alistair Barr.“Android Users”,[online]:http://www.wsj.com/articles/google-says-android-has-1-4-billion-active-users-1443546856
[3] TrendMicro.“Android Malware”,[online]:http://blog.trendmicro.com/trendlabs-security-intelligence/infographic-behind-the-android-menace-malicious-apps/
[4] Nwokedi Idika et al. “A survey of malware detection techniques”, Research supported by committee on Institutional Cooperation and General Electric,2007
[5] Palanichamy, Jaganathan, and Karthikeyan Ramasamy. "An improved feature selection algorithm with conditional mutual information for classification problems." Human Computer Interactions (ICHCI), 2013 International Conference on. IEEE, 2013.
[6] Feldman, Stephen, Dillon Stadther, and Bing Wang. "Manilyzer: Automated Android Malware Detection through Manifest Analysis." Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on. IEEE, 2014..
[7] Xiaoyan, Zhao, Fang Juan, and Wang Xiujuan. "Android malware detection based on permissions." Information and Communications Technologies (ICT 2014), 2014 International Conference on. IET, 2014.
[8] Jerome, Quentin, et al. "Using opcode-sequences to detect malicious Android applications." Communications (ICC), 2014 IEEE International Conference on. IEEE, 2014.
[9] Liang, Shuang, and Xiaojiang Du. "Permission-combination-based scheme for android mobile malware detection." Communications (ICC), 2014 IEEE International Conference on. IEEE, 2014.
[10] Arora, Abhishek, Shelly Garg, and Sateesh K. Peddoju. "Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices." Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on. IEEE, 2014.
[11] Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." Pattern Analysis and Machine Intelligence, IEEE Transactions on27.8 (2005): 1226-1238.
[12] Vinh, La The, Nguyen Duc Thang, and Young-Koo Lee. "An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information." Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on. IEEE, 2010