Proceedings of the 2014 International Conference on Wavelet Analysis and Pattern Recognition, Lanzhou, 13-16 July, 2014

ANDROID MALWARE DETECTION THROUGH PERMISSION AND PACKAGE

XIANGYU-JU

Department of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
E-MAIL: juxiangyu@126.com

Abstract:

Malicious Android applications are a seriously problem due to the large share of the Android operating system market and also the flexibility of Android. An application should be checked before installing to a phone to avoid the privacy information leak. This paper proposes a static android malware detection method by using not only the permission but also the package of an Android application. The experimental results show the proposed method can detect the malicious software effectively. It suggests that the information provided by the package is useful for detection.

Keyword:

Malware; APK; DEX; Permission; Package; Android

1. Introduction

Smart phones become popular nowadays. Android, a smart phone operation system, possess the highest market share of smartphone. Due to the flexibility, Android is more vulnerable to attacks in comparison with iOS. According to a research report from Lookout [1], during the first half of 2011, the number of malwares on the Android platform has been growing rapidly, including from 700,000 apps and 10 million units worldwide data. Malware infection increased by 2.5 times more than half a year ago. During the first half of 2011, malicious codes have increased from 80 to 400 in the market of independent applications in average.

Nowadays, there are mainly two kinds of Android malware detection technique: signature-based (static) and behavior-based (dynamic) detection [5]. Malware is detected without executing the applications by studying the characteristics of the codes. For the reason that this method can only learn features from the datasets which only contain detected malware and normal applications, it cannot detects the malwares which contains a new attack pattern [5]. Moreover, the malicious codes may be easily modified or encrypted [19]. On the other hand, behavior-based detection method executes applications and determine whether it is a legitimate or malicious software by monitoring its behaviors. The drawback of these methods is that the sandbox simulation environment is not easy to setup and costly [5].

In this paper, we propose a method to provide static analyst on android platform. Rather than only considering the permission information [6, 19, 20] , more information (i.e., the package information in DEX files) is also measured since some studies have shown that package information contains useful information of Android applications. We believe the proposed feature set enhances the testing accuracy since more useful information is provided. The features set is then evaluated by applying to well-known classification methods including KNN, Bayes, Linear Discrimination function and RBF Network.

The result of the paper is organized as follows. Section 2 mentions the recent research focus on features and classifiers used in the malware detection. Section 3 mentions the proposed android malware Detection method based on Permission and Package. Section 4 shows the experimental result of the detection accuracy. Section 5 give the conclusion.

2. Related work

In this section, we will introduce the Android malware detection research in recent years including research on feature and research on classifier

There are 135 permissions in Android platform, for example, android.permisssion.SEND_SMS and android.permission.READ_SMS. When a new application is installed, the permissions used in the application should be approved by users. However, due to the carefulness and lack of the knowledge, users usually accept the requests without considering the risks seriously. This issue has drawn the attention from many researchers. Woodpecker model [2] has been used to extract the Android API and the characteristics of permissions in APK by preloading application. A path or a link is detected to identify loopholes in the Android. Labor Schmidt et al. [8] adopted readeK command to extract ELF

978-1-4799-4211-4/14/$31.00 ©2014 IEEE

executable file as a function call list using PART. Some studies only focused on the most useful permission by feature selection [9, 4]. Besides the permission information, Intent and API are helpful in improving the performance of the detection [6, 10, 20]. Some studies have discussed the adversarial attack. For example, repackage method can hide the malicious code [11] and the behavior of the permission can be invoked indirectly [14]. The classes.dex file containing the actual Dalvik byte code for execution has been focused. The opcode [12, 13] extracted from class.dex files are applied for detecting repackaged smartphone applications.

Ben-gurion et al. [9] proposed the methods of using different classifiers to detect malicious software with 1000 features extracted from xml files and dex files. Nine kinds of classification methods are mentioned (the Bayes Net, Naive Bayes theorem, Decision Tree, PART, Boosted Naive Bayes theorem, Boosted J48, the Random Forest and VFI), three feature selection methods (Info Gain, Fisher Score and Chi - Square). The results show that the combination of Boosted Naïve Bayes and the top 800 features selected using InfoGain yield get an accuracy level of 91.8% with a 0.172 FPR. Dong-jia Wu [10] use K-NN, K-means, EM and Naïve Bayes to detect the malware. The result shows that when using K-NN and K-means as classification method, the accuracy shows best.

3. Proposed Android Malware Detection Method Based on Permission and Package

The proposed Android malware detection method based on permission and package is devised in this section. The feature vector for each application is constructed from the permission and package information of an Android application are firstly extracted from the manifest file and the DEX file respectively. Then the application is passed into a classifier to detect whether it is malicious. This process is illustrated in figure 1. The detail of the permission and package feature extraction is described in next sections.

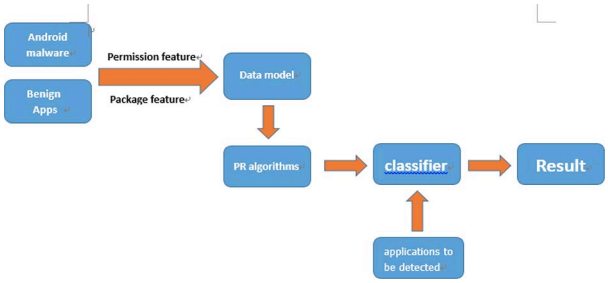

Figure 1. Android malware static detection model

3.1 Permission

Permission for actions in Android application, for example, resource accessing and network connecting, should be granted. Otherwise Android application is not able to work properly. Android contains 135 kinds of permissions, including mobile phones, cameras, Internet, keyboard, Bluetooth, SMS, etc [2]. If users need to access protected resources, corresponding permissions must be declared in AndroidManifest.xml file. Figure 2 shows an example of AndroidManifest.xml file.

```
<manifest . . .>
    <uses-permission android:name="android.permission.SEND_SMS" />
</manifest>
```

Figure 2. Example of Manifest code file

The permission is important features for Android malware detection because the permission shows whether a malicious application need to access any sensitive resources. For example, READ_SMS is a permission that allows the program to read text messages. Benign software, except communication related software, rarely apply for this permission. However malicious software may need this permission in order to get the user's privacy. 135 permissions stored in binary values have been used to detect the malicious applications. We use the model below to extract permission features.

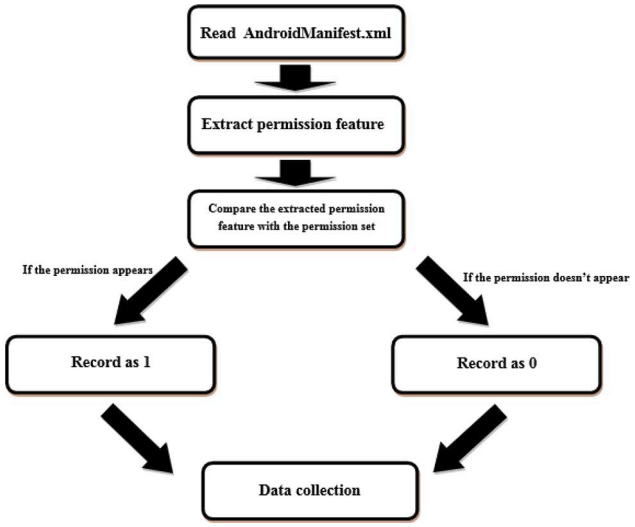

Figure 3. Permission feature extraction model

3.2 Package

The packages can be divided into three types. As for the package with beginning of Java, it provides the program basic functions such as Java.io.* allows program to execute the input or output functions. The package begins with android can call Android system functions. For example, android.locatoin.* allows the application to get the location information of users. The package begins with org is typically used for web services. An example of package information is shown in figure 3. We only select packages beginning with android as feature since they can call android system functions directly. For example, android.wifi.* calls the Wi-Fi service of the phone. As for packages beginning with Java, it only call the basic functions of program and packages beginning with org only focus on the part of web

```
// Decompiled by Jad v1.5.8g. Copyright 2001 Pavel Kouznetsov.
// Jad home page: http://www.kpdus.com/jad.html
// Decompiler options: packimports(3)

package android.telephony;

import android.os.Parcel;

// Referenced classes of package android.telephony:
//            NeighboringCellInfo

final class a
    implements android.os.Parcelable.Creator
{
```

Figure 4. Source code of Android application

Similar to the permissions, if users need to invoke the Android system functions in the program, corresponding package need to be imported. As a result, package provides useful information for malware detection. For example, android.location.* is a permission that allows the program to get the user's location. Benign software, except map application or some communication related software, rarely apply for this package.

Packages beginning with android can be separated into three parts. For example, android.os.Parcel, each word represent each part. Only first two parts are considered:. Android.os.Parcel and android.os.Handler will both be seen as android.os.*.

We select all 37 package information to detect the malicious applications. If the application calls the package, we record the number of total calls of one package as feature. We use the model below to extract package features.

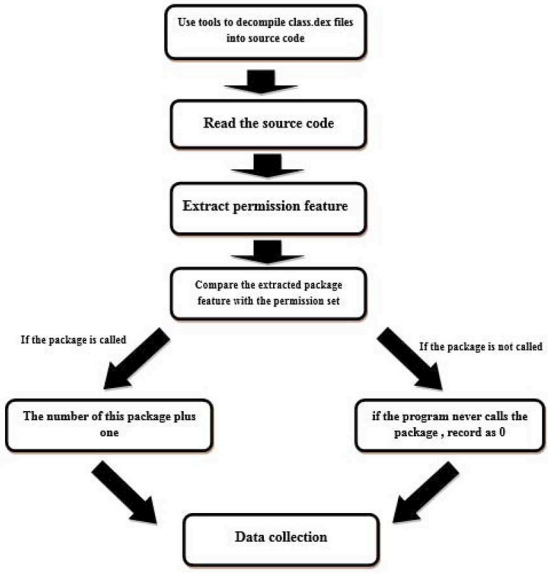

Figure 5. Package feature extraction model

4. Experiment

The performance of the proposed feature set is evaluated experimentally in this section. The Android APK dataset from Android network [2, 3] containing 622 normal and 174 malicious applications is used as the dataset. Each application is decompressed to get the source codes. The permission features from AndroidManifest.xml and package features from DEX file are extracted. Our proposed feature set is compared with the ones only using permission and package individually. The dataset is randomly spitted into half for the training and testing set in each experiment. The five well-known classifiers, including k-NN, Linear Discriminant Function, RBF Network, and Bayes are applied. The testing classification accuracy is adopted for evaluation. Each experiment is repeated 10 times independently.

The experimental results are reported in Table 1. The classifiers using only permission feature set achieves satisfying results, as shown in previous studies [20, 10, 6], which is at least 80% accurate. However, the performance of the classifier only using package feature set (less than 70% in average) worse than the ones with permission. It shows that the information provided by the permission is more useful than the package for Android malware detection.

On the other hand, our proposed feature set achieves the best performance among these three methods in terms of accuracy in all cases except K-NN. In average, the accuracy

of the proposed feature set is 84.4%. The result suggests that the package and permission complement each other in order to increase the accuracy of the detection. The proposed method which applies not only the permission but also the package information can improve the accuracy of Android malware detection.

Table 1. Detection accuracy of Bayes, knn, Linear discrimination function, and rbf network with permission, Package and Permission+Pakcage feature set

Accuracy	Feature Sect		
	Permission	***Package***	***Permission + Package***
Bayes	80.2%	60.1%	81.5%
KNN	85.7%	87.5%	84.5%
Linear Discrimination function	88%	52.5%	89.5%
RBF Network	80.5%	77.4%	82.1%

5. Conclusion

The proposed method combines these two types of information (permission and package) for malware detection. The experimental results indicates that the proposed feature sect outperforms the methods only using permission or package information. It suggests that using both types of information improve the accuracy of the detection system. As for the only using package as feature, the accuracy is much lower than the other two method. Further research will focus on this part and we will try to extract more useful information from DEX file.

Acknowledgements

This work is supported by a Fundamental Research Funds for the Central Universities (10561201326).

References

[1] Lookout report. http://www.36kr.com/p/15444.html
[2] Android developers. http://developer.android.com/
[3] Android APK. http://www.anzhi.com/
[4] Android downloads. http://os-android.liqucn.com/
[5] Nwokedi Idika, Aditya P. Mathur, Survey of Malware Detection Techniques, In Proceedings of 2013 5th Conference on Information and Knowledge Technology (IKT), page 113-120, 2013
[6] Peiravian, N.; Dept. of Comput. & Electr. Eng. & Comput. Sci., Florida Atlantic Univ., Boca Raton, FL, USA ; Xingquan Zhu, Machine Learning for Android Malware Detection Using Permission and API Calls, In Proceedings of, 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI), page 300-305, 2013
[7] Michael Grace, Yajin Zhou, Zhi Wang, Xuxian Jiang. Systematic Detection of Capity Leaks in Stock Android Smartphones. In Proceedings of the 16th Network and Distributed System Security Symposium, NDSS12, February 2012.
[8] Aubrey-Derrick Schmidt, Rainer Bye, Hans-Gunther Schmidt, Jan Clausen.Static Analysis of Executables for Collaborative Malware Detection on Android. In the prceedings of ICC'09. IEEE International Conference on Communications, page 1-5, 2009
[9] Asaf Shabtai, Malware Detection on Mobile Devices, In Proceedings of 2010 Eleventh International Conference on Mobile Data Management (MDM), page 289-290, 2010
[10] Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, Kuo-Ping Wu, DroidMat: Android Malware Detection through Manifest and API Calls Tracing, In Proceedings of 2012 Seventh Asia Joint Conference on Information Security (Asia JCIS), page 62-69, 2012
[11] Detecting repackaged smartphone applications in third-party android marketplaces. In Proceedings of the second ACM conference on Data and Application Security and Privacy, CODASPY '12, pages 317 - 326, New York, NY, USA, 2012.ACM.
[12] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. In Proceedings of the 19th Annual Network & Distributed System Security Symposium, Feb. 2012.
[13] Juanru Li, Dawu Gu, Yuhao Luo, Android Malware Forensics: Reconstruction of Malicious Events, In Proceedings of 32nd International Conference on Distributed Computing Systems Workshops, 2012
[14] Adrienne Porter Felt, Helen J. Wang, Alexander Moshchuk,Steven Hanna, Erika Chin, Permission Re-Delegation: Attacks and Defenses, In Proceedings of 2012 International Conference on Computer Science & Service System (CSSS), page 871-874, 2012

[15] Cooper, V.N. , Shahriar, H. ; Haddad, H.M., A Survey of Android Malware Characterisitics and Mitigation Techniques, In Proceedings of New Generations (ITNG), 2014 11th International Conference on Information Technology, page 327-332, 2014

[16] Yerima, S.Y. , Sezer, S. ; McWilliams, G., Analysis of Bayesian classification-based approaches for Android malware detection, Information Security, IET (Volume:8 , Issue: 1), page 25-36, 2014

[17] Hyo-Sik Ham, Mi-Jung Choi, Analysis of Android malware detection performance using machine learning classifiers, In Proceedings of International Conference on ICT Convergence (ICTC), page 490-495, 2013

[18] Samra, A.A.A., Kangbin Yim ; Ghanem, O.A., Analysis of Clustering Technique in Android Malware Detection, In Proceedings of 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), page 729-733, 2013

[19] Xing Liu , Jiqiang Liu, A Two-Layered Permission-Based Android Malware Detection Scheme, In Proceedings of , 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), page 142-148, 2014

[20] Glodek, W. ; Harang, R., Rapid Permissions-Based Detection and Analysis of Mobile Malware Using Random Decision Forests, In Proceedings of Military Communications Conference, MILCOM 2013 - 2013 IEEE, page 980-985, 2013