Android Malware Detection Techniques- A Survey

Arun K, Preeja V

Abstract— Smart phone users are increasing day to day. Most of the smart phone users connect to internet for various purposes and still they remain unprotected from malware attacks. One of the most commonly used operating system in smart phones is Android. A lot of applications available in play store makes android popular. The available applications can be easily installed from play store. Still it is very difficult to distinguish between clean and malicious application. This paper reviews four different methods for detecting android malwares.

Keywords: Android, Malware, Mobile Malware, Malware Detection, Permission Pattern

I. INTRODUCTION

Nowadays smart phones usage has been increased dramatically. Most smart phones are based on android operating system. Android operating system is based on linux operating system [1]. The android operating system was designed primarily for the touch screen mobile phones. Each android application need to use system calls to interact with linux kernel. System calls invoked can be traced and is used to represent the application behaviour. Android applications runs on isolated area of the system that does not have access to rest for the system resources called sandbox. For accessing system resources permissions are required. Android developers assign different permission to their application according to the needs. Permission required by the application to run on the system are granted by the user during the installation time itself.

Android being open source helps users to install third party applications and also flashing android system leads to vulnerability in to attacks like mobile Trojan, bonnets [2,3,4] etc. “The year of malware” has been called on 2013 because android being a popular thus it becomes the target for mobile malware [5]. The study conducted by Jupiter research in 2013 also found that 80% of smart phones remain unprotected from malware attacks [6].

The discuss Attack tree based malware detection, malware detection based on permission, detection based on contrasting permission patterns and detection based on network traffic monitoring.

II. METHODS FOR DETECTING ANDROID MALWARE

The four different methods for android malware detection are

A. Attack Tree Based Detection
B. Based On Permission
C. Based On Contrasting Permission Patterns
D. Based On Network Traffic Monitoring

Revised Version Manuscript Received on April 24, 2016.

Arun K, M.Tech Student, Department of Computer Science, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram (Kerala). India.

Preeja V, Assistant Professor, Department of Computer Science, Sree Chitra Thirunal College of Engineering, Thiruvananthapuram (Kerala). India.

A. Attack Tree Based Detection

Attack tree based detection is based on both static and dynamic analysis [7]. The goal of the attack can be found through dynamic analysis. A tree structure represents the relation between attacks and their behaviour capabilities. For each attack path rules can be generated. A malware tree can is defined as T=(V,E) where V is non empty sets of AND-OR nodes representations, where root tells the goal of attack and leaf nodes tell the possibility of attacks. AND node represents different steps in achieving same goal whereas OR node represents different ways to achieve the goal. Each node is tagged, if ‘P’ the goal is realized else marked as ‘I’.

In static analysis phase Androguard is used for feature extraction [8], where security related parameters like permission, function calls, intent filters are extracted. Each tree nodes are marked in bottom up approach until the root is marked. Father node gets type value based on marking on child nodes. After marking each node are analyzed. If root denotes ‘P’ then the application is malicious and attack paths are recorded, else if root is ‘I’ the application is benign. In this phase there is no false negative procedure, only obtain potential attacks of the application. After completion of static analysis, dynamic analysis is done to attacks are utilized. In dynamic analysis features are extracted from logcat when applications execute in a TaintDroid [9] based device. The recorded attack paths from static analysis are used in behaviour analysis. Every rule is tested. During each detection stimulation takes more time overhead. Clustering is done for optimizing the same event for more than one rules and test parts are logical OR. After stimulation is over the rules are send. If all rules are mismatched then application is benign otherwise it is malicious.

B. Based On Permission

Android malware detection based on permission pattern uses Principal Component Analysis (PCA) and Support Vector Machine (SVM) for classification of collected data is malware or benign [10]. The permission is extracted as detection feature. To reduce classifier’s computation time PCA is used and SVM is used to maximize learning machine’s generalization by use of Vapnik’s structural risk minimization principle [11]. SVM was chosen for higher classification rate.

The detection framework consists of four modules. They are preprocessing module, Feature selection, SVM classifier, Feature Dataset. During preprocessing phase the system decompiles the software sample and feature set is extracted. This is done by unzipping the APK file where the global configuration file AndroidMainifest.xml contains the user id, permissions required, minimum API version etc. From each application permission list retrieved is compared with total android permission and stored the value as binary number in a sequence with comma separated. In feature selection phase PCA is used because of storage space,

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

computing power and other aspects are limited in smart phones. First the permissions are extracted from APK file and original feature matrix is calculated. The covariance matrix is calculated from original matrix and it is recorded, then eigen values and orthogonal eigen vectors are computed. After computation variance contribution is calculated and first k eigen value which meet the computation variance is selected. Then compute the first k principal components and a matrix is formed called the feature set, which can be used to train SVM. From feature vector future selection module consist of sample label {1,-1} where -1 represents benign software and 1 represents malware. The classifier is trained and established in a hyper plane. The sample space is divided into two parts which represents different procedures. Lagrange operator is used to maximize the objective function. By using kernel function and maximization objective function the application is classified as malware or benign. Feature Dataset is used for storing and updating features.

C. Based On Contrasting Permission Patterns

Malware detection can be classified as two categories: anomaly detection and misuse detection. Anomaly detection uses knowledge of normal behaviour to detect malicious of an application. Misuse detection uses the knowledge of what is malicious under inspection. Both detection techniques will have profiles representing its normal or abnormal application for detection of malwares. Both anomaly and misuse detection may have high false positive rate and high negative rate due to limitation of their profiles [12]. A hybrid profile is build using contrasting permission patterns. Hybrid profile consists of Malware Profile (MP), Clean Profile (CP) and Common Profile (CoP) [13]. Enclamald is constructed from all contrasting permission patterns of training dataset and parameters used for malware detection. First the training dataset D is divided into two subsets which contain malware and clean applications. Then association rules are done to discover contrasting permission patterns that will divide into any one of the hybrid profile. The output +1 indicated malicious and -1 indicated clean. The weight of weak classifiers is based on two metrics. First the length of pattern. A pattern has frequent item set consisting of one or more items. A pattern with longer length is taken. Next the support degree 'i' for dataset D is calculated. If pattern i is from MP the $sup_-(i)$ will be zero. Similarly if pattern 'i' is from CIP $sup_+(i)$ will be zero. If $sup_+(i) > sup_-(i)$ then 'i' can be malware, vice versa. Then weak classifier 'i', denoted as w_i is calculated with different scenarios as follows

i. If i € MP , $w_i = |i| \times sup_+(i)$
ii. If i € CP , $w_i = |i| \times sup_-(i)$
iii. If i € CoP , , $w_i = |i| \times (sup_+(i) - sup_-(i))$

Then the output of weak classifier will be {+1,-1}. After assigning weights enclamald computes dism(Y) for a given unlabeled instance Y. If dism(Y) is greater than coefficient value then it is malware else clean.

D. Based On Network Traffic Monitoring

Advanced Persistent Threat (APT) purpose is to know attacker needs and interference [14]. Malware attacks may cause serious damage to smart phone like user privacy, stealing user data, download and installing software without user permission, malware spreading [15] etc. Network traffic based malware detection uses data packets, extract the feature data, classification by using SVM classifier algorithm and determine the network traffic and also to locate application which produced abnormality. Network traffic monitoring framework consists of four components: Traffic monitoring, Traffic anomaly recognition, Response processing and Cloud Storage [16]. In traffic monitoring module the network data traffic data is extracted with the characteristics of smart phone. Two times the extraction from training data, one from google native system which gives traffic information extraction and other a malicious traffic information extraction. The week extracted traffic information is stored in the information database. After traffic monitoring, traffic anomaly recognition is done. Traffic anomaly recognition consists of abnormal finding engines and traffic correlation analysis. Abnormal found engine uses C-SVC [17, 18, 19] algorithm for traffic data classification and correlation analysis to determine abnormal software. The C-SVC and its decision function are constructed based on given training set with largest classification interval. To achieve the objective function with lagrange method dual problem with radial basis kernel functions and import KKT condition to get the conclusion. Then using primal dual relationship and kernel function reaches the decision function. Then support vectors label names and kernel parameters are stored for model prediction. Next abnormal traffic analysis is done to check network traffic is anomaly or not. First the network traffic vector is extracted. Each sample is marked with classification results and each value is composed of eigen value and its class variables. Store model parameter for testing and feature vector is extracted from test data. From the model parameters obtained then it can be determined whether the network traffic is anomaly.

Response processing is used for processing software which include software policy, database and access control. According to security policy of response processing the malware can execute automatically and by using access control module the malware can be disabled. Cloud storage is used for storing training data, abnormal traffic data, security strategy for new generated policy and testing data.

III. DISCUSSION

The four different methods help to detect malware in android OS. They are Attack Tree Based Malware Detection, Malware Detection based on Permission Malware Detection based on Contrasting Permission Patterns, Malware Detection based on Network Traffic Monitoring. Table 1 shows comparison of four different malware detection techniques. Attack tree based malware detection is done on both static and dynamic analysis. Android malware tree is constructed consist of finite set of AND-OR nodes. Where root node gives the attack goal. Each attack path is recorded and updated whenever new malware definitions are found. The same attack tree can be reused. Permission based detection first by using reverse engineering permission required is extracted. Then feature set is calculated. Using SVM the detection of malware is based on data set and the

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

input vector collected during selection phase. New malware can also be detected in this method.

Table 1: Comparison of the Methods

Method	Detection	Features
Attack Tree Based Detection	Static and dynamic analysis	Goal of Attack can be found Tree can be reused
Permission	Based on SVM	High detection rate Detect new malware without update of library
Contrasting Permission Patterns	Based on metric dism	Outperforms commonly used classifier for detection Length interval has significant impact on computation cost
Network Traffic Monitoring	Based on network traffic analysis	Defense against Malicious and APT attacks

Contrasting permission pattern uses hybrid profile where permission patterns are classified. Then an aggregate value is calculated for all permission. If aggregate value is less than clean application then it's a clean application else it's a malware. Network traffic monitoring based malware detection first the network traffic is analyzed. For each sample is denoted as -1 or 1. By analyzing model parameters we can determine the presence of malware or not.

IV. CONCLUSION

Four methods for detecting malware detection in android operating system has been analyzed. They are detection using attack tree, detection based on permissions, detection based on contrasting permission patterns and detection based on network traffic monitoring. In attack tree detection technique the goal of the attack can be found. Detection based on contrasting permission reduces the classification error. In network traffic analysis helps to improve defense against malicious attacks. All the detection techniques are efficient for detecting malware applications and easy to differentiate between malware application and clean application.

REFERENCES

1. http://developer.android.com/index.html
2. Fang Binxing, Cui Xiang, Wang Wei. "Survey of Botnets," Journal of Computer Research and Development vol. 48, pp. 1315-1331, 2011.
3. Yue Li, Lidong Zhai, Zhilei Wang, Yunlong Ren. "Control Method of Twitter-and SMS-Based Mobile Botnet," in Proceedings of Trustworthy Computing and Services in Beijing, pp. 644-650, 2013.
4. Bill Miller, Dale Rowe. "A survey SCADA of and critical infrastructure incidents," in Proceedings of the 1st Annual conference on Research in information technology, pp. 51-56, 2012.
5. Blue Coat. Blue Coat Systems 2013 mobile malware report. 2013.
6. BHAS N. Press Release: More Than 80% of Smart phones Remain Unprotected from Malware and Attacks, Juniper Research Finds [EB/OL].[2014-02-23] http://www.juniperresearch.com/viewpressrelease.php? pr=404.
7. Shuai Zhao, Xiaohong Li, Guangquan Xu, Lei Zhang and Z. Feng, "Attack tree based android malware detection with hybrid analysis," IEEE, pp.380-387, 2014.
8. Androguard. http://code.google.com/p/androguard/.
9. William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, et al. TaintDroid: an information-flow tracking system for real time privacy monitoring on smart phones. USENIX Symposium on Operating Systems Design and Implementation, 2012.
10. Zhao Xiaoyan , Fang Juan, Wang Xiujuan "Android Malware Detection Based on Permissions," IEEE, 2015
11. Song Jie, Party Li Cheng, Guo Chao, Zhao Meng, Security Mechanisms Analysis and Application Research of Android mobile platform [J]. Computer Technology and Development, vol. 20(6), pp. 152-155, 2010.
12. PIETRASZEK T, TANNER A. Data Mining and Machine Learning-Towards Reducing False Positives in Intrusion Detection [J]. Information Security Technical Report, vol.10(3), pp. 169-183, 2005.
13. Xiong Ping, Wang Xiaofeng, NIU Wenjia, ZHU Tianqing, LI Gang, "Android Malware Detection with Contrasting Permission Patterns," IEEE, 2014
14. Lidong Zhai, Yue Li, Zhaopeng Jia, Li Guo. "APT Threat Detection and Protection of Integrated Network Space," Netinfo security vol. 3, pp. 58-60, 2013.
15. Zhou, Yajin, and Xuxian Jiang. "Dissecting android malware: Characterization and evolution," Security and Privacy (SP), 2012 IEEE Symposium, IEEE. pp. 95-109, 2012.
16. Jun Li, Lidong Zhai, Xinyou Zhang, Daiyong Quan," Research of Android Malware Detection Based on Network Traffic" Monitoring", IEEE , pp. 1739-1744, 2014
17. C C Chang, C J Lin. "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, pp. 27, 2011
18. Boser,I. Guyon, and V. Vapnik. "A training algorithm for optimal margin classifiers," in Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152, 1992.
19. Cortes C, Vapnik V. "Support-vector networks," Machine learning vol. 20, pp. 273–297, 1995.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.