An Autonomous Protection Algorithm for Android Malware Attacks Based on Multiple Features

Zhu Xiaojing

College of Information Engineering, Yunnan Business and Technology University, Kunming, Yunnan, China

Keywords: Multi Feature, Android, Malware, Attack Behavior, Autonomous Protection Algorithm

Abstract: At Present, the Method of Detecting Android Malware Based on Permission is More Popular, But the Accuracy Rate Needs to Be Improved. However, the Detection Method Based on Function Call Has Obvious Disadvantages of Difficult Feature Extraction. Based on the Traditional Permission Feature Reservation, Android Malware is Detected by the Combination of Permission and Resource File Features, Which Has Good Convenience and Accuracy, and Can Significantly Avoid the Shortcomings of Traditional Detection Methods. Therefore, Based on Multiple Features, This Paper Discusses the Autonomous Protection Algorithm of Android Malware Attacks. through Combing the Current Relevant Research Literature, This Paper Makes a Detailed Description of Feature Extraction. Then the Experiment is Designed to Test the Effectiveness of the Proposed Algorithm. I Hope That through the Exploration of This Paper, We Can Provide Some Reference and Inspiration for the Improvement and Improvement of Android Malware Detection Methods.

1. Introduction

1.1 Literature Review

As Intelligent Terminal Devices Have Been Widely Used in the World, Malicious Code Detection Has Been Widely Studied. According to Sheng Jie and Other Scholars, the Current Research Methods Tend to a Single Feature and a Single Algorithm, with Low Accuracy. in This Regard, Sheng Jie and Other Scholars Proposed Static Detection Method Based on Multi Feature and Stacking Algorithm. Feature Vectors Are Composed of a Variety of Feature Information, and Classifiers Are Formed by Training Samples to Effectively Make Up for the Shortcomings of a Single Feature and a Single Algorithm (Sheng et al, 2018). According to Wei Lihao and Other Scholars, the Continuous Increase of Smartphone Usage Further Aggravates the Scale and Complexity of Mobile Malware. in This Regard, Based on Multi-Feature Collaborative Decision-Making, an Innovative Android Malware Detection Method is Proposed. by Analyzing and Extracting the Characteristic Attributes of Android Applications, and Combining Machine Learning Model and Classification Algorithm, We Can Make Timely Judgment on Malware (Wei et al, 2016). Wang Yong and Other Scholars Based on the Behavior Characteristics of Android Malicious Applications, Combined with Dynamic Analysis and Static Analysis. by Extracting Multi-Dimensional Features of Application Software, Such as System Commands, Function Api Call Sequences, Permission Requests, Etc., and Filtering Features, the Problem of Feature Imbalance Can Be Effectively Alleviated (Wang et al, 2018). Zhang Jiawang and Li Yanwei Proposed That Using Single Feature Machine Learning Algorithm is Difficult to Effectively Process Data and Detect Attacks. in This Regard, by Combining the Random Forest Algorithm with the Speech Recognition Model, Based on the Comprehensive Apk File Multi Class Features, an Innovative n-Gram Model is Designed (Zhang and Li, 2017).

1.2 Research Purposes

Android is a free application system, which surpasses many mobile platforms and is one of the most popular operating systems on the market. However, at present, there are more and more malware for Android platform on the market. According to 360 Internet Security Center statistics, the total number of malicious program samples intercepted has exceeded tens of

  DOI: 10.25236/ISMHI.2019.107

millions(Zhang,2014). This kind of malicious software steal privacy information, lock screen, blackmail and other attacks, will cause serious threats to the user's operation security, privacy security and so on. In addition, it will reduce users' trust in Android system. At present, there are two main detection methods for Android malware, namely static detection and dynamic detection, each with its own advantages and disadvantages. In this paper, we mainly choose static detection method and use classification algorithm to extract permission features and resource features. Thus, the false alarm rate of the method can be effectively controlled.

2. Feature Extraction of Classification Algorithm

2.1 Authority Characteristics

Some permissions are defined in Android system, which are usually used to control application access to sensitive resources. When the developer makes a certain label in the manifest file, the application can obtain the corresponding permission(Lu and Liu,2018). When the application for calling user rights appears in Android system, the system will remind the user whether to allow the program to obtain these rights. When the user is authorized, the module components of the application will access sensitive resources. In general, the system aapt.exe tool can be used for permission extraction, which is stored in the system's \ Android \ platform tool \ file. Specifically, when the system executes AAPT D permissions test.apk, it can obtain the test.apk permission information(Gao and Ye,2018). In this case, the system calculates different sample sets, such as the frequency of malicious samples, as a priori probability of permission samples.

2.2 Resource Feature Extraction

In general, an APK file is a compressed file. When APK files are generated in the compiler, configuration files, project resource files, and java files need to be packaged together, and the corresponding files can be obtained by decompressing the compressed package in the Linux environment (Zheng and Xian,2018). In fact, XML file is formed after encryption, and there will be garbled code after decompressing directly. Therefore, when extracting resource files, we usually use apktool open source tools, and then execute apktool D test.apk to extract the specific files. At the same time, we will use Python language to write programs to scan and extract files, so as to extract file features. In addition, the experimental extraction features also include the number of image files, XML files, and related file resources. In this case, both the picture and the XML file are saved under the layout folder. At this point, these components can also be rendered by scanning the Android manifest.xml file.

3. The Algorithm Design of Android Autonomous Protection Against Malware Attack under Multiple Features

3.1 Experimental Hypothesis

First, it is assumed that resource file features and permission features are independent of each other. The existing theory points out that there is a certain correlation between Android application permissions, and the focus of this experiment is to keep the characteristics of permissions, on the basis of which it is clear whether the characteristics of resource files can improve malicious attacks, so it is assumed that permissions are independent of each other. At this stage, few literatures point out that there is correlation between the features of resource files and the features of permissions are independent of each other, so it can be assumed that these conditions are true.

Secondly, it is assumed that the file features used in this experiment meet the normal distribution. Generally speaking, the number of resource files is in a continuous state, so it is difficult to get a prior probability by calculating the frequency. Therefore, the number of experimental samples (malicious attack samples, normal application samples) is more than 1500. At this time, it can be assumed that the number of sample resource files can be normally distributed. Then, we test the

hypothesis of normal distribution by sampling samples. However, the number of resource files is almost normal distribution, so the experimental hypothesis can be established.

3.2 Experiment Design

In terms of the source of experimental sample selection, this malicious sample library comes from virusshare. Download 5526 malicious samples from this source, and ensure that each sample is determined as malware by antivirus software. The normal samples come from Anzhi market, and the 5021 downloaded software is determined by anti-virus software to have no malicious behavior. In the experimental environment, the author uses a desktop computer, the CPU is Intel i5-8400, the main frequency is 3.0GHz, feature extraction and malicious attack test are using python programming.

3.3 Experimental Environment and Process

The first step is sample pretreatment. In this process, the author uses three anti-virus software to detect the downloaded samples and delete the samples that do not conform to the requirements. Through sample preprocessing, 4926 malicious samples and 4806 normal samples were obtained.

The second step is to generate training samples and test samples. In this experiment, 1200 normal samples and malicious samples were selected from the sample set, and 8 times were selected repeatedly. One group of data was used as training samples, and the other five groups were used as test samples.

The third step is to calculate the prior probability and extract features. In this case, the permission set is represented by P, and the probability is represented by φ. Assuming that the malicious sample set and the normal sample set are represented by M and N respectively, the authority Pi is the number of malicious samples in the normal sample set, and the number of malicious samples in the normal sample set is d_i. Then, in calculating the probability of occurrence of malicious samples and normal samples, formulas (1) and (2) are used respectively.

$$\varphi(P_i / M) = c_i / 1200 \tag{1}$$

$$\varphi(P_i / M) = d_i / 1200 \tag{2}$$

In order to test the attack behavior of Android malware, this experiment calculates and compares the application probability of malicious samples and normal samples. After calculation, the probability formula of this verification is derived as follows. The resource eigenvector of the samples to be classified is $X = \{x_1, x_2, \cdots, x_6\} \quad (1 \le i \le 6)$. These contents represent the number of files, the number of pictures, and the number of four components. In order to test the attack behavior of Android malware, this experiment calculates and compares the application probability of malicious samples and normal samples. After calculation, the probability formula of this verification is derived as follows. The resource eigenvectors of the samples to be classified are, which represent the number of files, pictures and the number of four components. In this case, the classification sample set is $P = \{p_1, p_2, \cdots, p_m\}$, in this case, malicious attack test.

3.4 Experimental Results

Before evaluating the attack behavior of Android malware, you need to understand the following concepts. The number of malicious samples correctly detected, expressed in TP, otherwise expressed in FN. Normal samples are correctly classified as normal samples, which are represented by TN, and vice versa by FP. From the above concepts, the following indicators are derived. A is the accuracy rate.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

Through the statistical analysis of the overall sample, it shows that the distribution of the number of resource files and components is similar to the normal distribution. Analysis component includes

Android application development component, and resource file includes picture and XML file. Through the detailed analysis, we can see that there are great differences between malicious samples and normal samples in the distribution of characteristic probability. At the same time, on the basis of known resource files and basic component probability, naive Bayes algorithm is used to classify them respectively, and permission and resource multi features are combined to detect the specific effect.

Experimental results show that malware is usually generated by injecting malicious code into the software after decompilation, which has very similar characteristics with normal software. In this case, we need to design an independent protection algorithm in Android system to prevent malware and protect the privacy of users.

4. Conclusion

Based on the characteristics of permission and function call, it is the traditional way to detect Android malware. These traditional methods have obvious inherent defects, which are not conducive to the optimization and improvement of Android system. In this regard, a new autonomous protection algorithm is proposed on the basis of preserving the authority characteristics. By adding classification features, we adopt normal distribution model and naive Bayes classification. It is found that the new self-protection algorithm has a good detection effect, which can provide a new improvement idea and optimization direction for malware attack detection.

References

[1] Sheng J., Liu Y., Yin C.Y. (2018). Detection method of Android malware based on multi feature and stacking algorithm. Computer system application, 27 (2), 197-201.

[2] Wei L.H., AI J.Q., Zou H., Cui L., long Z.Y. (2016). Multi feature collaborative decision detection method for Android malware. The only official website for computer engineering and application, 52 (20), 5-13.

[3] Wang Y., Cai J.Y., Meng C., Liu Z.Y., Xue J.F. (2018). Detection method of Android malicious application based on multi feature fusion. Journal of information security, 3 (04), 59-67.

[4] Zhang J.W., Li Y.W. (2017). Android malware detection system based on machine learning algorithm. Computer application research, 34 (6), 1774-1777 + 1782.

[5] Zhang S.Q. (2014). Android malware detection based on Improved Bayesian classification. Radio communication technology, 40 (6), 73-76.

[6] Lu X.R., Liu Z.Y. (2018). Malware behavior detection based on Android API call. Computer and digital engineering, 47 (3), 710-715.

[7] Gao Z.H., Ye M. (2018). Android malware detection based on simhash algorithm. China new communication, 20 (19), 91-92.

[8] Zheng Y.M., Xian Q. (2018). Classification detection of Android malware based on permission information. Modern computer (Professional) (6), 67-71 + 75.