Smart Terminal Malware Detection Technology Based on Abnormal Network Behavior

Minglan Yuan1,*

1Department of Commerce and Trade Management, Chongqing Business Vocational College, 401331, Chongqing, People R China

*Corresponding author e-mail: yuanminglan1123@cqbvc.edu.cn

Abstract. In recent years, the use of smart mobile terminals has increased rapidly, especially mobile phones on the Android platform, which account for almost half of the mobile phone market. Many attackers regard smart terminals as the target of their attacks. They can easily collect user privacy information, thus posing a huge threat to users. In this regard, researching a malware detection technology for smart terminals is of great significance to the safe use of mobile phones in my country. The purpose of this article is to study the smart terminal malware detection technology based on abnormal network behavior. In this paper, the method of reducing feature data is used to preprocess the captured original network data to improve the applicability of the naive Bayes algorithm in the network behavior data processing environment. The preprocessing part first cleans up the network data and removes useless information; then uses methods such as establishing a static address library and field query to divide the data, normalize the complex data, and finally construct the feature vector. The main part of the anomaly recognition module in this paper is the naive Bayes classifier, and the preprocessed data is introduced into the naive Bayes classifier as a feature vector for classification detection. This article introduces the privacy data monitoring technology to track the data flow of the malware detected by the anomaly recognition module, and determine the leakage path of its privacy data, so as to clarify the magnitude of its harmfulness for further processing. Research shows that this article has tested the performance of the system, and the time-consuming items all use the stopwatch timer that comes with the mobile phone to assist in timing, which can be accurate to 0.01 second.

Keywords: Malware Detection, Network Behavior, Private Data, Detection Technology

1. Introduction

In recent years, the rapid development of network technology has made the network an indispensable part of people's daily life. However, although the network has brought convenience to users, there are more and more attacks on the network [1-2]. Although many organizations and government enterprises have established relatively safe protection mechanisms, the attack methods are becoming more and
more diverse, and the consequences are becoming more and more serious \cite{3-4}. Under such severe circumstances, the need to take measures to prevent the deterioration of the network ecological environment is becoming more and more urgent \cite{5-6}. However, it is impossible to ensure that cybercrimes absolutely do not occur. Only effective methods can be used to detect network anomalies and deal with them in time. This method is called network anomaly detection \cite{7-8}.

In the research of intelligent terminal malware detection technology based on abnormal network behavior, many scholars have studied it and achieved good results. For example, Muhtadi AF extracts the destination IP address in the packet header, and then discrete wavelet transforms it. Perform statistical analysis to determine whether an abnormality occurs \cite{9}. Yen Y S proposed to detect DDoS attacks by calculating the energy distribution of wavelet analysis. The so-called energy distribution is actually a parameter that indirectly reflects whether the flow rate changes. If there is no abnormality in the flow rate, then the parameter value distribution will be relatively stable. On the contrary, it means that there is an abnormality \cite{10}.

This paper reduces and normalizes the characteristic data, and uses Bayesian classification algorithm to classify the data to identify whether it is abnormal software; for networked malware, this paper proposes a detection scheme based on network behavior, directly from the network behavior data generated by the malware is analyzed from the perspective of the network data, and the network data is extracted at the Linux kernel layer to ensure the authenticity of the data.

2. Malware Detection Technology for Smart Terminal with Abnormal Network Behavior

2.1. Network Packet Capture Technology

The complete Netfilter-Iptables network packet processing framework in the Linux kernel consists of two parts, one part is the "hook" HOOK function of Netfilter, and the other part is Iptables.

(1) Netfilter framework

Netfilter is a series of call entrances that are introduced by the 2.4.x kernel and embedded in the IP protocol stack of the kernel. In addition to the function of a firewall, the biggest feature is that it can perform various message processing operations such as message encryption and message statistics. The Netfilter framework places some hooks on the network protocol stack. Developers can place some processing functions at the detection points, and the processing functions will be triggered when the data passes through these HOOKs.

(2) Privacy data monitoring technology

In the Android dynamic monitoring solution, the concept of tainted data is extended to private data in the Android system, such as SMS data, contact data, location information, etc. TanitDroid can analyze many behaviors of the application, such as the Activity, Service, and BroadcastReceiver components that the program starts.

2.2. Principles of Bayesian Classification Algorithm

The theoretical basis of Bayesian classification algorithm is Bayes' theorem, which provides a method of calculating hypothesis probability, according to the prior probability, the probability of observing different data and the method of calculating the posterior probability of the observed data itself. Among them, Naive Bayes is a classification method to deal with discrete features. The following are the known conditions and required problems of the naive Bayes algorithm:

(1) Known conditions

1) Classification attributes. $C = C_1, C_2, ..., C_N$, and it is required that this is a complete set, that is, it can contain all categories;

2) The overall prior probability $P(C_i)$ of each category attribute;

3) The conditional probability of each attribute, if the sample is expressed as $X = (x_1, x_2, ..., x_n)$, then the conditional probability is $P\left(\frac{x_i}{C_i}\right), j = 1 - n, i = 1 - N$.

(2) Question
Assuming that $X_1 = (x_1, x_2, ..., x_n)$ is any random vector in the feature space, Bayesian theory is used to determine the attribution class of the feature vector. The formula is as follows:

$$P\left(\frac{C_i}{X}\right) = P\left(\frac{X}{C_i}\right) \frac{P(C_i)}{P(X)} \quad (1)$$

in case:

$$P\left(\frac{C_i}{X}\right) = \max_{j=1, \ldots, N} (P\left(\frac{X}{C_j}\right)) \quad (2)$$

Then $X_1 \in C_i$ can be determined.

2.3. Overall Design of System Framework

(1) Management module design

The function of the network information acquisition module is to monitor the network behavior information generated by the current software; the abnormal identification module is to detect abnormal software based on the network behavior information; the privacy data monitoring module is to track the data flow of the detected malware according to the user's needs to determine its malicious type; finally, exit the module management interface and save the setting information.

(2) Network information acquisition module design

Network information acquisition function: Obtain network behavior data of third-party application software installed in the system, and provide data to be detected for malware detection. Through a large number of literature reading and actual research, the following three conclusions are reached:

1) The Trojan or botnet control terminal will use an IP address to communicate with the controlled client to achieve the purpose of command transmission and operation result submission. After the client is controlled, it will often visit a certain malicious address and check it every once in a while. Check whether the network environment is connected. If it can be connected, initiate a link to the control terminal and go online. Therefore, this article needs to monitor the source/destination address.

2) After analysis, the application obtains the terminal's IMEI and mobile phone system version information, and then sends it back through the network. Often, some visited network URLs contain IMEI and mobile phone system version information, so the URL will also be one of the attributes detected in this article.

(3) Design of abnormal recognition module

The main algorithm of this module is the Naive Bayes algorithm. Its theoretical basis is Bayes' Theorem, which is an algorithm for calculating hypothesis probability. The key data is the source/destination address, port, URL, and upstream/downstream traffic. Before applying the algorithm, the obtained data must be preprocessed to make it more suitable for Bayesian algorithm.

(4) Naive Bayes classifier design

The classification process of the classifier is divided into two stages: training and testing. In the training phase, the classifier is trained with the labeled training sample set to generate the classifier model. In the testing phase, the data extracted by the network behavior data acquisition module is preprocessed and imported into the classifier for identification. Label candidate samples. The classification result of the data $s_i$ in each training sample is $m_1 = 1$ or, then $m_2 = 0$. Each sample is composed of eigenvalue vectors and class variables, which can be expressed as two-tuples $(X, m_1)$ and $(X, m_2)$. Construct a training sample set. Randomly select normal samples and abnormal samples to form a training sample set:

$$D = \{(\tilde{x}_1, m_1), (\tilde{x}_2, m_2), ..., (\tilde{x}_{m-1}, m_1), (\tilde{x}_m, m_2)\}, m_1 = 1, m_2 = 0 \quad (3)$$

(5) Design of response processing module

The response processing module is the part that processes the detection results in the entire system. Other modules send information and trigger the module. The workflow of the module is: the abnormal identification module and the privacy data monitoring module send the results to the module, and the module processes according to the user's choice. Corresponding software, and save the processing results into the database, that is, the operation log.

(6) Storage module design

The function of the storage module is to access, modify and delete the data generated during the
operation of the system. This module uses the lightweight database that comes with the Android system, namely the SQLite database. The data stored in the local storage module in this system includes: training data, captured network behavior information, model parameters, operation logs of each module, etc.

2.4. **Abnormal Recognition Module**

The main algorithm of the anomaly recognition module adopts the naive Bayes algorithm. Since this module is in the Linux kernel layer, it is written in C language. There are two main data files, namely train and test. After the network information acquisition module captures the network behavior information, it sends the information to the abnormal identification module. The abnormal recognition module is divided into two processing processes, namely the training process and the testing process.

1. **Training process**: The training data is a data set with category marks, and the Bayesian classifier of the abnormal recognition module is trained to determine the model parameters;
2. **Testing process**: The test data is data without a mark, and the data category is determined according to the model parameters during the test.

3. **Experimental Research on Malware Detection Technology of Intelligent Terminal for Network Abnormal Behavior**

3.1. **Test Platform**
In order to improve the authenticity of the test environment, the system is tested on Android phones.

3.2. **Test Data**
Part of the malware selected for this test comes from the collection and sorting of malware sample sets on the Internet. These malware data sets are divided into 125 malware samples. Since mobile phone malware has many variants and fast updates, this article collects and adds 7 newly discovered malwares such as “time synchronization” to the malware collection. In addition, in order to ensure the balance of the data set, this article has downloaded as many current popular online applications as possible in the official Android market and application store.

3.3. **Function Test of Exception Handling Module**
Before system testing, first create a static address library in the data preprocessing stage; then collect as balanced data as possible to train the Bayesian classifier of the anomaly recognition module to determine the parameter model; finally, a test data set must be constructed.

4. **Experimental Research and Analysis of Malicious Software Detection Technology on Smart Terminals for Abnormal Network Behavior**

4.1. **Comparative Test Analysis**
In order to illustrate the effectiveness of the system scheme, this paper selects the existing detection scheme for comparison test. The detection scheme is a detection scheme based on system information and kernel call information. It collects system information and software kernel call information when the software is running, uses the K nearest neighbor algorithm to complete machine learning, and identifies malware similar to the sample. Compare this scheme Accuracy rate, false alarm rate, and false negative rate are three indicators. If the software that cannot be determined in the experimental results is considered to be an error judgment, the results are shown in Table 1.

| Table 1. Comparative test results |
|----------------------------------|
| **Comparative Test** | **System detection scheme** | **Template detection scheme** |
| Accuracy               | 94.47               | 88.63               |
| False alarm rate       | 5.63                | 13.24               |
| False negative rate    | 11.68               | 11.85               |
Figure 1. Comparative test results

The comparative test results show that the accuracy and false alarm rate of the proposed scheme are better than the existing detection schemes, and the scheme in this paper has high reference significance for mobile malware detection systems.

4.2. Performance Impact Analysis

This system may affect the performance of the mobile phone. In this paper, the performance of the system is tested, and the test results are shown in Table 2. Average power consumption refers to the power consumption per hour under the same conditions, which is measured by the device's own "battery" function; the power-on time is the time from pressing the power button to the display of the mobile phone desktop, and the time-consuming items are all red the stopwatch timer that comes with the Mi mobile phone to assist in timing, can be accurate to 0.01 second.

Table 2. Performance test results

| Test items                  | Before system installation | After system installation |
|-----------------------------|---------------------------|--------------------------|
| Average power consumption(%)| 4.2                       | 7.5                      |
| Average boot time(s)        | 12.14                     | 6.21                     |
| Time consuming(s)           | 0.43                      | 0.64                     |
| Average time to open an application(s) | 2.5                    | 8.3                      |
Figure 2. Performance test results

As shown in Figure 2, it can be seen that the system will have a certain impact on the performance of the mobile phone, but the impact is within the range acceptable to users.

5. Conclusion

This paper conducts a functional effectiveness test and a performance impact test on a malware detection system based on network behavior, and analyzes the test results. For each module function of the system, this article selects software samples that have traffic consumption and privacy stealing behaviors from the malware collection and the normal application software of the official application store to test the effectiveness of the system function. The test results are deeply analyzed and the system is still existing problems. Finally, I tested and analyzed the impact of the system on the performance of the mobile phone, and the test results showed that the impact was within an acceptable range.

References

[1] Jiwei M , Zhipeng L , Xuejiao L . Feature Contour Construction Methods Based on Neural Network for Network Abnormal Behavior Detection[J]. Journal of Physics: Conference Series, 2020, 1693(1):012001 (4pp).
[2] Saidah Mastura A. Ghani, Mohd Faizal Abdollah, Robiah Yusof. Recognizing API Features for Malware Detection Using Static Analysis[J]. Drake Law Review, 2015, 26(3):10-10.
[3] Huang X , Ma L , Yang W , et al. A Method for Windows Malware Detection Based on Deep Learning[J]. Journal of Signal Processing Systems, 2020(1):1-9.
[4] Lu T , Liu X , Chen J , et al. AFCGDroid: Deep Learning Based Android Malware Detection Using Attributed Function Call Graphs[J]. Journal of Physics: Conference Series, 2020, 1693(1):012080 (8pp).
[5] Kim J I , Lee H S , Jung Y G . Malware Behavior Analysis based on Mobile Virtualization[J]. The Journal of The Institute of Internet Broadcasting and Communication, 2015, 15(2):1-7.
[6] Yao H , Liu Y , Fang C . An Abnormal Network Traffic Detection Algorithm Based on Big Data Analysis[J]. International Journal of Computers, Communications & Control (IJCCC), 2016, 11(4):567.
[7] Du J , Xie P , Zhu J , et al. A Method for Detecting Abnormal Behavior of Users Based on Selective Clustering ensemble[J]. Iet Networks, 2017, 7(2):85-90.
[8] Xiao F , Lin Z , Sun Y , et al. Malware Detection Based on Deep Learning of Behavior
Graphs[J]. Mathematical Problems in Engineering, 2019, 2019(PT.4):1-10.

[9] Muhtadi A F, Almaarif A. Analysis of Malware Impact on Network Traffic using Behavior-based Detection Technique[J]. International Journal of Advances in Data and Information Systems, 2020, 1(1):17-25.

[10] Yen Y S, Sun H M. An Android mutation malware detection based on deep learning using visualization of importance from codes[J]. Microelectronics Reliability, 2019, 93(FEB.):109-114.