A New Security Reinforcement Technology of Mobile Application

Chao Zhou, Yajuan Guo, Wei Huang, Jing Guo, Haitao Jiang

State Grid Jiangsu Electric Power Research Institute, Nanjing 210000, China

Keywords: mobile applications, android, security reinforcement, code confusion

Abstract. Android platform has gradually become the most popular mobile terminal operating system, while the number of software applications based on which is also quite amazing. At the time, the security threats are also increasing, and the degree of threat is gradually deepening. Therefore, we need to create an Android application security reinforcement system. This paper is a study on the new method of strengthening the mobile application. First of all, this paper introduces the risk that Android face and the core technology of software security reinforcement; then this decides the basic function of the reinforcement system, design and realize the dex reinforcement technology based on code confusion and bin file loading technology based on security shell technology. Finally, this paper carries out the implementation and testing of the system. The test results can be seen in Figure 5 and Figure 6, and the test results show that the dex file after code consolidation reinforcement cannot be successfully compiled source files and the corebin is from the disk. The proposed Android application security reinforcement system design is reasonable, and the security reinforcement system we designed achieves the expected goal.

1.INTRODUCTION

Smart phones are rapidly becoming popular around the world in recent years. The current mainstream mobile phone operating system including iOS Symbian, Windows, Phone BlackBerry and OS Android [1]. Android is a Linux kernel based operating system, and it is mainly used for mobile phones and tablet computers, and gradually extended to other related areas. With the rising market share of Android system and the increasing number of Android users and application software, the attacks on Android system are becoming more and more [2-4].

The inherent rights control and digital signature can be cracked by root authority or signature, and its security is limited. With the development of various types of application stores and operator's store, there are more and more applications for people to choose [5]. After considering the security of the application and the convenience of the legal application, we put forward the idea of software reinforcement, and the theory of software reinforcement is put forward from the point of view of the third party. The function that it wants to realize is to carry on the reinforcement of the code to any application, and this requires reinforcement system must be universal, and it can complete the reinforcement process under the premise of not getting the application code and prevent threats from malicious attackers (these include: illegal copying and unauthorized use). This topic will study the protection technology of Android platform, and put forward a security reinforcement system based on the Android platform application software to achieve the purpose of protecting the Android application software.

2.RESEARCH CONTENTS

2.1 The risk of Android

In the same time when the Android application software is developing rapidly [6], China's Android users are also facing a lot of threats. And the common malicious threats are as follows:

(1) Malicious deduction

In the most common malicious behavior, the mobile malicious code is malicious deduction. The threat of attack is in the user's knowledge or unauthorized circumstances using illegal means to allow users to order all kinds of charges or use mobile phone payment services in the unconscious condition, and it will shield the service SMS send back by service providers, and damage the

normal function of the system.

(2) Privacy theft

Privacy theft is a threat to the popular application in recent years, and it is in the case of users do not know or without authorization to steal the user's secret information. After the phone infected with this virus, the virus began to steal user privacy information through the background, including call recording, message content, IMEI, IMSI, geographical location, address book, browser history information, and then the virus uploaded the information to the remote server is controlled by hackers. If the user has a network payment and other acts on the phone, the secret of the account will also be a serious threat [7].

(3) Remote control

Virus boot automatically in the background, and it communicated with the server in the case of users do not know or not authorized. Then it used the interaction with the server without permission to carry out the deduction of fees and download rogue software and other malicious acts. For example, the famous AnserverBot virus, is the remote control through Sina blog,

(4) Tariff consumption

In the case of users do not know or not authorized, the threat lead to the loss of user charges though automatically sending SMS, MMS, email, network connection etc.

(5) Malicious communication

In the case of users do not know or not authorized, it spread itself, its derivatives or other mobile Internet malicious code by copying, infecting, delivering, and downloading.

(6) Other

Virus download a lot of software in the background, and consume the user's mobile phone traffic, or perform some of the more power consumption operation to consume mobile phone power, thereby affecting the normal mobile phone communication.

2.2 Core technology of software security strengthening

2.2.1 White box encryption algorithm

Due to the limited processing performance of mobile phone, the traditional non symmetric encryption algorithm has high encryption security, but the processing speed is too slow, and it will affect the application usability. So we can only use the symmetric encryption algorithm. White box algorithm using AES (Advanced Encryption Standard) encryption algorithm. AES is a more advanced algorithm in the field of symmetric encryption, and white box AES algorithm refers to the use of AES algorithm of the white box encryption scheme. Using this scheme can be done every host has a customized decryption box. In this way, a terminal is broken, and it will not affect the normal use of other terminals. This is of great significance to improve the security of Android applications [8-9].

2.2.2 Integrity check

For the processing power of Android mobile phones, the most suitable for the mobile phone to achieve the anti-tampering technology is the integrity of the check, and the integrity protection is achieved by hash [10]. Each file can be calculated using the MDS Hash validation program to calculate a fixed MDS code. Our reinforcement scheme is to calculate the hash value of the program written by the hash algorithm and store it in the configuration file, then carry out the integrity test in key processes when starting the program to ensure the integrity of the application. And after studying the file format of Android application, we intend to carry out the following integrity protection:

First, the program should unzip the application, and then calculate the classes.dex file to verify the value. After the verification value is encrypted by the white box algorithm, it needs to check value replacement to safety in loader by means of key exchange. Then it has to calculate the check value of security loader and the check value of the bin file in the original APK package. Then the two parity values are added to the bin file generated after the file encryption. When the program starts, classes.dex will will start security loader when it called the bin file, and the security loader will first test the integrity of the classes.dex file.

If (! verify Hmac (“classes. dex")) //Integrity check

Return 0; //If the validation fails, the security is not loaded and return 0

2.2.3 Anti compile / anti disassembly / dynamic debugging

Java code is compiled into the middle of the code is very easy to be reduced to source. Because of its anti - Anti - Compilation and other work does not have a strong sense, so for the Java code, we generally make code confusion. By the principle or object to classify the confusion technology, the code confusion technology can generally be divided into shape confusion, control structure confusion, data confusion, prevention of confusion and so on. The main means of realization of the confusion is to delete or change the name of the program. The deletion refers to delete some debugging information that does not affect the execution order of Nirvana. The changing name refers to the transformation of the identifier in the program, which is meant to prevent an attacker from understanding the program, and the identifier includes variable names, names, the name of the class and method names, etc. Control structure confusion is to adjust the control flow of the program, so that the attacker cannot understand the control flow of the program. Data confusion refers to the process of restructuring data in a logical manner, and the data confusion algorithm achieved the protection of the program by increasing the difficulty of the attacker to attack. The prevention of confusion is designed for a specific anti compiler. Generally speaking, this kind of confusion is the use of the anti-compiler of the defect or Bug to confuse the code, so as to realize the defense function.

For disassembly, we use the way to join the flower instructions. The disassembly needs to determine the starting position of the first byte of the instruction, and the flower instruction is actually a kind of no sense in the process of running some of the assembly instructions. The reverse dynamic debugging technology under the Android platform and the reverse dynamic debugging technology under the Linux are basically the same. Generally, it can be carried out by the following 2 ways: to check if there is a debugger in the system; to detect whether the running speed is attenuated

2.3 Scheme design of software reinforcement technology

2.3.1 Dex file reinforcement

For the classes.dex of Android applications, we use method of code confusion to realize the reinforcement. Here, we need to confuse the original dex header files in the application. The original DEX head has only 70 bytes, and we add the dex file of application to the header of the dex file. After such confusion, the anti-compiler cannot be the main line of the normal anti compiler. In the process of execution, it will solve the 1000 bytes at first, and then execute the remaining axes of the DEX file, so that the application can be normal call up.

2.3.2 bin file reinforcement

For the bin file which can be executed in the APK file, we carry out the way of safe loading。 First, we need to rename the bin in the source file to corebin, and then name the security loader as the original bin file name. When the classes.dex file calls to bin file, it will first call our loader. And then we can call the corebin though the loader. So we can not only realize the protection function of the software in the loader, but also can realize the safe loading of corebin. The loading process is shown in Figure 1 below:

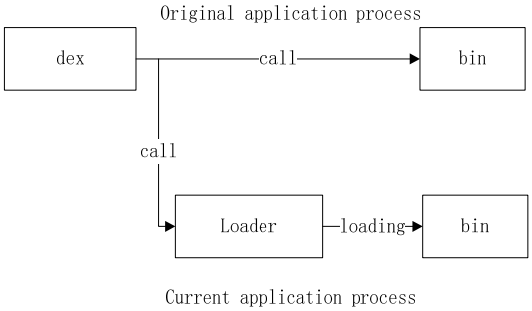

Fig.1 the Process of loading safely

2.4 Implementation of security reinforcement system

The software reinforcement system is a general system that can be used to reinforce any application. This system can strengthen the dex files and bin files, and protect the security of the code, and it can do the integrity check of DEX file and bin file. The reinforcement system can be divided into the following subsystems: APKfuscator, corebin encryption and cutting, key exchange in loader, signature package. Their flow charts are shown in Figure 2, Figure 3 and Figure 4.

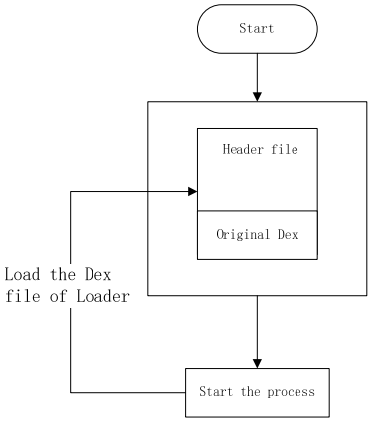

Fig.2 the Flowchart of Confuse

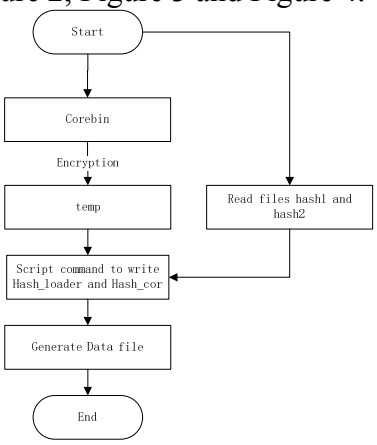

Fig.3 the Flowchart of Encrypting and Cutting

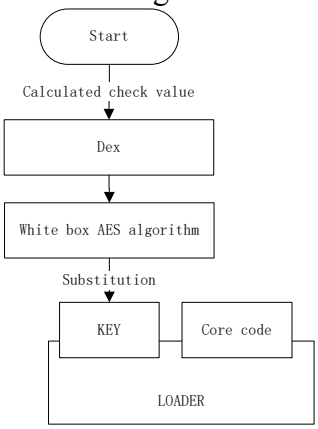

Fig.4 the Flowchart of Key Replacement

APKfuscator is the realization of code confusion technology, and it carries out the work of code confusion for dex files. The dex files after APKfuscator confusion will not be able to use the anti-compiler software to crack. This method first uses the XOR algorithm to achieve an encryption program, and then uses Linux shell script and use script commands to read the hash value of hash_loader and hash_core stored in hash1 file and hash2 file, finally, it writes them to the end of file and generate new data files. After extracting the rename APK, we calculate the calibration value of the DEX file. For this value, we first perform the encryption of the white box AES, and then use the replacement method to write it into the loader. After all of the above work is completed, we need to put dex, loader, corebin and other files extracted from the APK together to resign, and then we need to pack them again into a APK, which is the application of the reinforcement for the user to use.

3.TEST RESULTS AND DISCUSSION

In order to confirm the validity of the code, we use the anti-compiler tool to decompile the DEX file before and after code confusion separately. And the test results show that the dex file after code consolidation reinforcement cannot be successfully compiled source files.

In order to validate the security loading technology, we use adb shell to enter the application after the installation of the file directory to observe. And the results are shown in the following figure.

```
InsertCoin@/data/data/com.test# ls -l
total 68
drwxrwx--x    2 10138    10138        4096 May  3 06:33 cache
-rwxrwxrwx    1 10138    10138       50628 May  3 06:33 cmcc_omp_safetybin
drwxr-xr-x    2 1000     1000         4096 May  3 06:33 lib
-rw-rw-rw-    1 10138    10138         120 May  3 06:33 random.reinforce
-rw-------    1 10138    10138           0 May  3 06:33 seed.reinforce
drwxrwx--x    2 10138    10138        4096 May  3 06:33 shared_prefs
InsertCoin@/data/data/com.test#
```

Fig.5 the APK File before Reinforcing

```
linux-sm9i:/home/ss # adb shell
InsertCoin@/# cd data/data/com.test
InsertCoin@/data/data/com.test# ls -l
total 108
drwxrwx--x    2 10138    10138        4096 May  3 06:33 cache
-rwxrwxrwx    1 0        0           91636 Apr 26 08:17 cmcc_omp_safetybin
drwxr-xr-x    2 1000     1000         4096 May  3 06:33 lib
-rw-rw-rw-    1 10138    10138         120 May  3 06:33 random.reinforce
-rw-------    1 10138    10138           0 May  3 06:33 seed.reinforce
drwxrwx--x    2 10138    10138        4096 May  3 06:33 shared_prefs
InsertCoin@/data/data/com.test#
```

Fig.6 the APK File after Reinforcing

Through the above figure, we can find that there is no corebin file in the directory of the program after the reinforcement, and it only contain the loader, which has been renamed to the original bin file name. So it is confirmed that the corebin is removed from the disk.

4.CONCLUSIONS

Through the analysis of the above results, we can find that the dex file after code consolidation reinforcement cannot be successfully compiled source files; and there is no core bin file in the directory of the program after the reinforcement, and it only contain the loader, and the core bin is removed from the disk. All of these show that the proposed Android application security reinforcement system design is reasonable, and the security reinforcement system we designed achieves the expected goal.

REFERENCES

[1] Wang L, Gao L, Kong F. The application of anti-seismic reinforcement technology in primary and secondary schools reinforcement project [J]. Shanxi Architecture, 2012.p.12-14

[2] Qiu-Hua J I, Amp G S, Co T. Research on Security Reinforcement Technology of Cloud Operating System [J]. Mobile Communications, 2014.p.10-17

[3] Li Y, Ji C, Fan G. Designing of Mobile Marketing System Based on the Internet of Things Technique[J]. Jiangsu Electrical Engineering, 2015.p.80-84.

[4] Hu Y, Wang C, Yuan J. Design and Realization of a Mobile Application System for Electric Distribution Network Rush Repair[J]. Jiangsu Electrical Engineering, 2014.p.49-52.

[5] Gong X. On the Bridge Reinforcement Technology's Application and Analysis [J]. Friend of Science Amateurs, 2011.p.37-39

[6] Zhao B. The application of W-beam guardrail reinforcement technology in Xi'an mountain road section [J]. Shanxi Architecture, 2013.p.42-48

[7] Wang G, Yao D, College S P. Ponder over the Reinforcement of Ethic Education in Police Colleges and Universities from the Perspective of Cultural Security [J]. Journal of Shanghai Police College, 2015.p.15-18

[8] Bai L. The Reinforcement Technology of Construction [J]. Urbanism & Architecture, 2013.

[9] Cao X Z. Research on the Reinforcement Technology of the Fourth Ring Under-railway in Shenyang [J]. Value Engineering, 2015.p.36-39

[10] Li P. Application research on the strengthening method for reinforced concrete columns and shear walls by equal surface layer section replaced with high-strength materials [J]. Building Structure, 2013, 43(1):85-90.