ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 134

2nd International Conference on Control, Automation, and Artificial Intelligence (CAAI 2017)

Security Test Method Technology for Mobile Application Based on Code Analysis

Zhenyu Liu

Shanghai Development Center of Computer Software Technology, Shanghai Key Laboratory of Computer Software Testing and Evaluating, 3F, Technical Building, No 1588, Rd Lianhang, Shanghai, CHINA

***Abstract*—The rapid development of mobile application, this paper gives a test method to find the security issues in the with mobile application software base on code analysis. The paper analyzes the code features of Android application and proposes static test method for Android application. According to code and configuration characteristics, the paper gives the knowledge base and makes the code feature comparison using the existing knowledge base. The full test procedure also proposed and helps the tester to make test design and test execution. The method is used to test some mobile application and find possible security problem to avoid the security attack or personal information leak.**

Keywords- mobile application; security test; code analysis

I. INTRODUCTION

With the development network and hardware technology, the more and more mobile application has emerged in these years. With the era of Internet coming, the speed of software develop is quickly and period of application publish is short to face the pressure of market. The initial mobile device is only a communication device and provides the basic functions, including phone, message function, just like a phone. But now, the upgrading of network technology, the emergence of new technologies are affecting the life of person. The person could use the mobile application to do the e-commercial, social activities and so on.

The mobile device means not only the traditional communication, but also capability of process data transmission through the development of mobile data networks. The emerging artificial intelligent technology also integrated into the mobile device. Therefore the current mobile device has been integrated the different hardware or sense device, such as camera, Bluetooth, GPS, NFS and other components. These components integrated into the mobile device and provide the more powerful and useful ability to assist the function of application. In recent years, these components, which from ever-increasing screen resolution to 64-bit processors, have become increasingly tightly integrated with mobile applications and further improved through different communication protocols. To support all platform development of HTML5 technology, mobile applications gradually replace the application software in the traditional computer device.

The mobile applications become an important software in the mobile Internet compared to the traditional computer. The traditional computer could not portable and the users must use computer device with various peripheral equipment. But mobile devices integrate the all the peripheral within one portable device. Therefore, the mobile devices have the more ability to manage these components with the user or application permit. On the other side, the mobile device provide the data directly without support of wired network. However, the mobile application security has been more and more impact on end user. The users of mobile application focus potential security danger in the era of mobile Internet. Whether it is iOS operation system or Android system, application security of mobile software is the key issues. The increasingly safety threaten also comes from the emerging malicious software. The safety threaten are not only a serious threat to the user's personal privacy, but also to the user caused property damage.[1][2]

Malware is a software or program that is intentionally implanted into the mobile software for malicious purposes. The malware is controlled by destroying the software process. The malware mainly has several types, such as viruses, worms, trojans, trapped doors, logic and time bombs. The virus refers to a group of self-replicating damage to the hardware, software and data via program code. The worm is able to spread their own consumption of the network and local system resources to refuse to attack the malicious code. Trojan refers to some look hidden code of the system that is useful or harmless or exploits or damages the program. The trap door is a secret entry into the program that is often designed by hackers to enter the system in a special, unauthorized manner. The logical and time bombs are a piece of code embedded in some legitimate procedures will under certain conditions in the implementation of harmful program damage to the system. As for mandatory installation, it is difficult to select the different components. The malware will produce browser hijacking, advertising pop, malicious collection of user information, malicious deductions, malicious uninstall, malicious bundles and other violations of the user's right to know.[3]

The mobile application in application market cannot identify malware which led to the proliferation of malicious software. With the rapid growth in the number of applications, Android mobile phone applications in a large number of applications store and website. Users cannot identify whether or not exist malicious code in mobile application to be downloaded.[4]

The main structure of this paper is as follows: the second part gives the related works. Section 3 introduces the test method of code analysis. The next part gives test procedure for

mobile application. Finally, there are conclusions and future works.

II. RELATED WORKS

Mobile application software security detection consists of two typical method, one is static detection analysis and the other is dynamic behavior detection analysis.

Code static analysis refers to the static analysis of the application package, such as through the pattern matching method or disassemble the source code in software package. The intelligent pattern recognition used to identify the application of malicious behavior instructions.[5][6]

According to the mobile application malicious code characteristics and malicious behavior of the knowledge base, mobile application software, also called package, could be tested using the results of comprehensive static analysis technology to finally determine the application of security. [7]Security testing includes application security, data security, operating system security, including:

- Software permissions: run application, consider whether there will be deductions risk, disclosure of privacy risks, unauthorized access to such factors.
- Installation and uninstall security: the installation of application, whether the inclusion of digital signatures, whether the bundled with other software, whether from the start, uninstall the use of other data and so on.
- Data security: whether application process some sensitive data. The data should not be in the form of plain text data stored in other documents or temporary files. The temporary files should be deleted in time to avoid the data suffered attacks, theft, cause unnecessary losses and so on.

The dynamic analysis of malicious code recognition technology is identify characteristics in he dynamic state of the program running to analyze whether it contains malicious code Intelligent mobile device malicious code has common dynamic behavior characteristics includes of software installation malicious plugins, forced boot from the start without the user confirm, mandatory networking access without the user confirm , uninstall is not clean, send malicious SMS, read user privacy information, malicious group send text messages.

Dynamic analysis method is to monitor and record the behavior of the application software to determine whether it has malicious behavior. The active defense technology is similar to find the security danger during the app running. For Android compatible operating system, there are two ways to achieve the application software behavior monitoring.[8] One is the use of open source in Android system , modify the operating system and insert a specific behavior monitoring layer code. The operating system software could not be modified, but the application can use the hook technology. The hook technology implements the key sensitive API to achieve information or action. Dynamic behavior analysis can run on the real terminal of the intelligent mobile device, but in order to improve the efficiency of automated detection, you can run the test in the PC simulator environment. There are two main ways to realize the malicious code recognition technology based on dynamic analysis. The one is to establish the underlying system detection module, so that it can detect, intercept, record malicious behavior. Another way is to use hook technology to detect the behavior of the sensitive API call. The establishment of the underlying system module refers to the existing Android system source code transformation, adding security detection module. The detection tool can detect, record and process the behavior of sending the deduction information, illegal link, illegal content and stealing user privacy data during the operation of the software.[9][10]

III. TEST METHOD

The safety issues come from the mobile application. The mobile applications has their own character, such as code, configuration, parameter setting and platform. Therefore, the code of mobile applications are studied and we propose the method to find the security issue

A. Code Check

Android code confusing technology is an Android application protection technology. The technology is used to protect APP from being cracked and reverse analysis. Android is a Linux-based free and open source operating system. Therefore, android application is most of the Java language development, compile will produce Dalvik byte code (ie dex file). The instruction execution of Davik virtual machine which run by Android is interprets execution. The use of Java development of the application is easy to be reverse crack, the current more popular Java program anti-compiler tools are baksmali, dex2jar, jd-gui, apktool and so on.

In order to prevent the mobile application is reverse crack, anti-decompilation tools become an important means and the anti-compiler tools can not run properly. The basic principle of code obfuscation is to reorganize and process the program so that the processed code and the code before the completion of the same function, and confused code is difficult to be decompiled. It is hard to get the true semantics of the program even if the decompile program.successfully. Also it is difficult to read and understand the decomplie program. The main purpose is preventing reverse analysis code.

Currently known Android application code obfuscation techniques are:

- Java class name and method name confused: in the Android SDK comes with proguard code obfuscator. It will delete some debugging information and use some meaningless sequence of characters to replace the class
- Java code confused: after the process of the code order confusion, the actual run-time confused on the code flow. The use of anti-compiled code is difficult to read and understand, so as to enhance the code reverse difficulty. However the original code flow is the same, reading the decompilation of the code flow static and the original process is also very different, so that crackers are difficult to understand the code through static analysis function. The protection code is not reverse analysis. The Dalvik Bytecode Encryption: encrypts some or all of the Dalvik bytecode in the dex file. Each time it is executed by a dedicated native code that is responsible for dynamic

decryption. And the static decompiled code becomes unusually unreadable

B. Config Check

All Android applications composes with configuration file, which is important for software running. This configuration file is named Android Manifest.xml and store in the root directory of each Android application.

This configuration file is part of each application installed by the user, and part of the Android platform itself. Android Manifest file is a important control file that defines the related mobile application information. Therefore the configuration should be also checked with the different ways. The first way is check the permissions whether enough and minimal. The checklist table is used to fulfill this work. The detail the checklist can see the Table I.

TABLE I. CHECKLIST

No	Check Description
1	permissions are available to the application
2	applications have access to these permissions
3	application that can be called across a privilege barrier
4	permissions required for these calls
5	operation of the user application
6	process of component operation
7	component visibility and access rules
8	various libraries and functions
9	minimum Android version for running the application

This configuration file defines the access control policies for your applications and software components. The configuration file also defines the details of the application and component hierarchy that the Android system uses to interact with your application. In the configuration file, the software can perform operations such as declaring permissions, sharing a process with other applications, supporting external storage, managing the visibility of the component. Therefore, the AndroidManifest.xml file is the most important file in the entire application and is critical to the application security. The configuration file is not extensible so that the application cannot add its own attributes or tags. The complete list of labels and their nested tags in the configuration file

IV. TEST PROCEDURE

Here we give the full procedure for testing the mobile application with the static method. The each steps in this procedure will be introduced.

A. Procedure

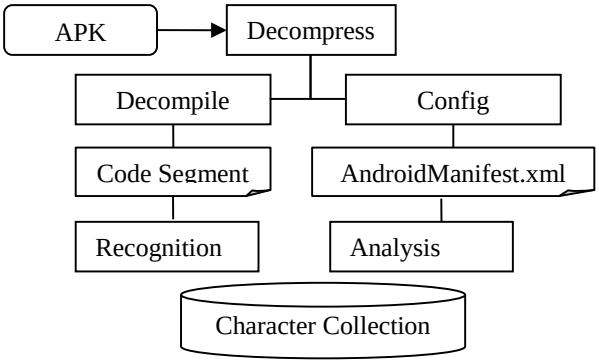

FIGURE I. TEST PROCEDURE

The code scan generally refers to static source code scanning, which is one of the most recent software application security solutions that have been mentioned in recent years. It refers to the software engineering, the programmer composes the source code without the compiler. The scanning tools could directly use some scanner to find out some code semantic defects or security vulnerabilities in the solution. The current static scanning technology has evolved from the 90's era, the coding rules that match the analytical techniques that have been extended by compiling techniques to the full path of the program simulation. Thus, this simulation performs more execution paths than dynamic execution Many, can find a lot of dynamic test difficult to find defects. Static code scan is characterized by the actual implementation of the program. The implementation of software will be fast, high efficiency, but the main drawback is false positives or omission to some extend. But the use of the recent second generation of static code scanning technology, a special query Language CxQL is used to find security issues, you can effectively eliminate false positives and solve omission problems. And now most of the use of static source code scanning tool to help the software development team in the software development process quickly and accurately find, repair and manage security issues in software code. The issued can not only reduce the application system security risks but also reduce the application code in the preparation. The issues may occur security vulnerabilities and improve the application system's own security capabilities. So this solution will ultimately develop a safe and reliable application.

There are many typical code scan software. PMD is an open source tool for scanning Java code errors. PMD uses static analysis to get code errors. That is, it can report an error without running a Java program. PMD comes with a number of rules that can be used directly, using these rules to find many problems with Java source programs, such as potential bugs, unused code, optional code (String, StringBuffer abuse), complex expressions, duplicate code, and unallocated resources and so on. In addition, users can define their own rules. Check that the Java code conforms to certain specific encoding specifications.PMD can be integrated as a plug-in to many popular IDE (such as Eclipse, IntelliJ IDEA, etc.), many of the plug-ins are included in the PMD jar file. The jar file contains a rule set..

B. Character Collection

The collection of malicious code samples is mainly reported by the security software in the mobile device, as well as the search engine cluster in the network. After obtaining a malicious code sample, it is filtered by static feature matching detection at first. And then we discarded if malicious code is known, and automatic detected based on virtual machine analysis for suspicious programs for unknown malicious code samples and client upload files. Finally, through the artificial intervention and analysis, the malicious code characteristics which newly extracted will be update the malicious code into signature library. The implementation process is as follows:

1) Decompile after the AndroidManifest.xml file permissions analysis.

2) Android commonly used permissions are randomly arranged, and that serial number, constitute a set of permissions: P={P1,P2,...,Pn}，Each of the arrays represents a class of permissions, 1 indicates that the permission, 0 means no such permission.

3) Extract the permissions used by the software to be tested, and expressed as a binary permission set.

4) Compare the binary feature set of the software to be tested with the binary feature subset of the known malicious permission combination in the library.

C. Code recognized

Malicious code recognition technology is divided into two categories: based on the static characteristics of the malicious code recognition technology and dynamic analysis based on malicious code recognition technology. In this study, we will take a combination of static analysis and dynamic analysis of the technical method.

Based on the static characteristics of the malicious code, recognition technology is analyze software code from the software producer, software unique ID, signature certificate information, version, installation package file features, executable file features, permissions and so on. The static feature information determine whether the malicious code after analysis of suspicious programs and feature matching using the code signature library.

D. Permission Analysis

Android system could protect system resources to prevent unauthorized to visit with the design of the authority mechanism. When a developer uses certain permissions in mobile software, it must declare the relevant rights required for the operation in the configuration file The user must achieve the software authority to function properly. For different levels of harm, developers will be divided into four levels: normal, dangerous, signature, and signature or system.

If an application applies for high-risk authority, the software is highly risky. Due to most applications do not apply for such high-risk sensitive rights according to statistic analysis. The applications that apply for such privileges generally have some special features, it should filter out the application of such authority firstly. When you need to use SET_DEBUG_APP authority, the program will be able to configure a program for debugging mode. The malware developers can download the source code containing the hidden API, and modify the application. Therefore, the application unset for SET_DEBUG_APP authority will lead to possible to prevent some anti-malware operation.

The Android application shell is packaged as an APK file, also similar to the Java jar file. Each APK must contain a Manifest file that requests access to certain rights to the Android operating system. The file consists of access to various hardware devices, sensitive information about the operating system, and access to certain exposed parts of other applications.

V. CONCLUSIONS

This paper studied the static code analyze for mobile application. The paper firstly introduces the security danger of malware and consequently carried out test method of code analyze and configuration check. The paper gives the test procedure for executing step for code analysis. The well designed test technology and related test method could get code analysis results, the character code will be compared with the knowledge and achieve the good results.

ACKNOWLEDGMENT

The work is supported by Shanghai STCSM Program under Grant No. 16511101202.

REFERENCES

[1] Himanshu Dwivedi, Chris Clark, David Thiel. Mobile Application Security. 2012.

[2] Neil Bergman, Mike Stanfield, Jason Rouse, Joel Scambray, CISSP. Hacking Exposed Mobile: Mobile Security Secrets & Solutions. 2014.

[3] Dominic Chell, Tyrone Erasmus, Jon Lindsay, Shaun Colley, Ollie Whitehouse. The Mobile Application Hacker's Handbook. 2015.

[4] T Luo, H Hao, W Du, Y Wang, H Yin. Attacks on WebView in the Android system. Twenty-seventh Computer Security Applications Conference, 2011:343-352.

[5] V Rastogi, Y Chen, W Enck. AppsPlayground: Automatic Security Analysis of Smartphone Applications. Acm Conference on Data & Application Security & Privacy, 2013:209-220.

[6] Y Zhou, X Jiang. Detecting Passive Content Leaks and Pollution in Android Applications. Proc. of the Symposium on Network and Distributed System Security (NDSS), 2013.

[7] M Grace, Y Zhou, Z Wang, X Jiang. Systematic Detection of Capability Leaks in Stock Android Smartphones. 2012.

[8] P Brussee, J Pouwelse. Autonomous smartphone apps: self-compilation, mutation, and viral spreading. 2015.

[9] Asaf Shabtai, Dudu Mimran, Yuval Elovici. Evaluations of Security Solutions for Android Systems. 2015.

[10] Biswajit Panja; Dennis Fattaleh; Mark Mercado; Adam Robinson; Priyanka Meharia. Cybersecurity in banking and financial sector: Security analysis of a mobile banking application. 2013.