Reverse analysis of Wangxin application based on android platform

Yu Shi*, Lulin Ni, Qingbing Ji
Science and Technology on Communication Security Laboratory, Chengdu 610041, China

ABSTRACT

Wangxin application is an instant messaging software launched by Alibaba Group that can support Windows, Android, and iOS operating systems, and its chat messages are stored in the local folders of smart terminals. In this paper, we study the encryption protocol of the Wangxin application and propose a reverse analysis scheme for the Wangxin application. Since the data encryption such as chat messages are stored in the database, we analyze the encryption mechanism of the local database in detail and give a decryption key extraction method based on dynamic binary instrumentation. Finally, through practical tests, our scheme can accurately extract the decryption key and restore the local database, providing a complete solution for the Wangxin application data forensics.

Keywords: Reverse analysis, encryption protocols, data decryption, dynamic binary instrumentation

1. INTRODUCTION

With the development of the Internet, people's demand for information exchange has become higher and higher, and instant messaging software has emerged[1]. The main popular instant messaging software in China are WeChat, QQ, and Wangxin, and foreign instant messaging products such as Whatsapp, Telegram, and Signal.

With the improvement of instant messaging software functions and enhanced services, the impact on people's life is also increasing, so the analysis of communication protocols of instant messaging software becomes more and more important. Ni et al.[2] analyzed the behavioral characteristics of WeChat on the Android side from the traffic level, and further analyzed the traffic consumption in each scenario such as WeChat login, logout, and message interaction to establish; Wan et al.[3] used reverse analysis and debugging techniques to analyze that WeChat version 4.5 login authentication uses RSA for login authentication and then sends the session key to the client via AES while the session mode of WeChat uses the session key as a symmetric key and encrypts the session plaintext using AES-CBC-128; Lijun Zhang et al.[4] described the encryption algorithm, key export principle, and decryption reduction of WeChat database by analyzing the WeChat private protocol in detail.

Wangxin application is an instant messaging software launched by Alibaba Group specifically for individual consumers to facilitate transaction communication. In addition to supporting chat communication it also supports logistics message access, group chat, and other rich features. Users also save their personal account information in their smart terminals while using the Wangxin application, in addition to a large amount of chat data. However, Wangxin uses a private encryption protocol where the data is stored locally through encryption and the encryption keys are stored in different memory segments. This makes it impossible for ordinary methods to obtain user data.

In this paper, we will study the local database encryption mechanism of the Wangxin application, propose a reverse scheme for Wangxin application based on the Android platform, and give a detailed decryption key extraction method, and finally, through practical tests, the key we extracted can correctly decrypt and restore the Wangxin application database file.

2. ANDROID PLATFORM REVERSE ANALYSIS RELATED TECHNOLOGIES

2.1. APK File decompilation

APK[5] is the source code of Android application packaged into a kind of application package file format recognized by Android system through compilation, APK file includes compiled code files, file resources, native resource files, certificates, and manifest files. Therefore, in order to analyze the execution code of the original application, the APK file

*sherryxupt1118@163.com

International Conference on Computer Application and Information Security (ICCAIS 2021),
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12260, 122601O · © The Authors.
Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2637666

must be decompiled[6]. APK file decompilation means using decompilation tools such as Apktool to convert the code file in the APK file, i.e. .dex file, into the registered language of the Dalvik virtual machine, Smail language.

Apktool is a Java open source project by Google and can be downloaded from http://code.google.com/p/android-apktool/. The process requires the JRE (Java Runtime Evironment) to be installed. Using Apktool, you can disassemble the APK installation package into resources, configuration files, Smail code files, and so on. Among them, the AndroidManfifest.xml file contains some configuration information of the application, and the .so file stored in the lib directory is the dynamic library file written in C[7].

2.2. Static analysis techniques

Static analysis is a technique that uses lexical and syntactic analysis to scan executable program files to generate disassembly code without executing the program, and then read and understand the disassembly code to understand the implementation principle of the program[8]. For Android programs, the main purpose is to understand the operation mechanism of the program by analyzing three files, namely, AndroidManifest.xml, Smail code, and libxxx.so in the lib directory, which are obtained after decompiling the APK file.

From AndroidManifest.xml, you can get information about the entry class of the application, whether there is a custom Application class, the requested permission and the registered Activity, etc. The Smail code is a bit more difficult to read, usually you need to use jd-gui to analyze the jar converted from dex2jar. The most difficult to read is the dynamic library file in the lib directory, which is a native library file developed in C/C++ and compiled with the NDK[9], and disassembled to obtain CPU-level assembly language. Because of the security of .so files, many Android applications currently use the NDK to put their core code such as registration verification, encryption and decryption into .so files, and then call it in Java layer as JNI. The analysis of .so files generally requires tools such as IDA and GDB.

2.3. Dynamic binary instrumentation

Dynamic binary instrumentation (DBI)[10] is one of the dynamic analysis techniques, which refers to the insertion of a specific analysis code during the execution of a program according to the user's analysis requirements without affecting the dynamic execution results of the program to achieve the monitoring and analysis of the dynamic execution process of the program.

The key point of the dynamic binary instrumentation technique is the use of Hook, also known as hook function, the main principle is that the hook program can capture the message before the system calls the function, the hook function gets control first, then the hook function can both handle or change the execution behavior of the function, but also get information about the parameters of the function, and even force the end of the message delivery. At present, the widely used dynamic binary analysis platforms are Frida, Pin[11], DynamoRIO[12], and Valgrind[13], among which Frida is a lightweight Hook framework that can run on Android, Linux, and Windows platforms, and the dynamic binary instrumentation of the Wangxin application in this paper mainly utilizes Frida.

3. WANGXIN APPLICATION REVERSE ANALYSIS SOLUTION

In this paper, we mainly analyze the local data encryption and decryption protocol of the Wangxin application using static analysis technique and dynamic binary instrumentation technique. The analysis process is shown in Figure 1, and the specific scheme is as follows.

(1) Decompile the APK file of the Wangxin application installer with the Apktool tool.

(2) Static analysis of the obtained decompiled code, locating the program entry class through AndroidManifest.xml file, locating the code segment of local data encryption and decryption process realized by Wangxin application through Smail code, analyzing and sorting out Wangxin local data encryption algorithm and key exporting principle.

(3) Analyze and locate the Hook point that is the key generation function, write Frida script based on python for the key generation function, and use Frida binary instrumentation platform to obtain the local data encryption and decryption key during the operation of Wangxin application.

(4) Decrypt the local ciphertext database of the Wangxin application according to the principle of local data encryption and the key obtained through binary staking to restore the plaintext database.

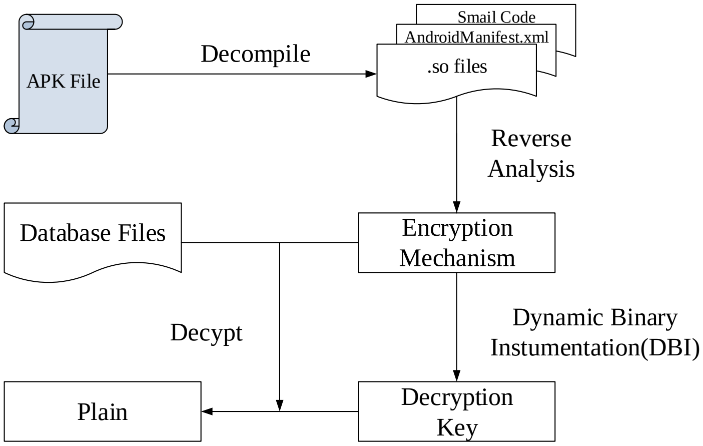
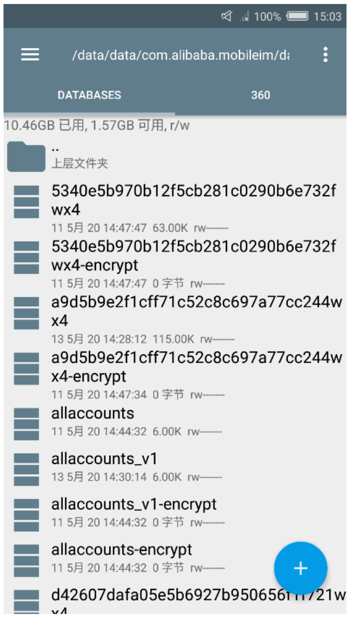


Figure 1. Wangxin application reverse analysis process.

4. WANGXIN APPLICATION LOCAL DATA ENCRYPTION MECHANISM

4.1. Wangxin data local storage

Different applications in Android have a specific file to store their application data, and the local storage path of the Wangxin application in Android is "/data/data/com.alibaba.moblileim". The Wangxin app stores the user's chat data in the "/data/data/com.alibaba.moblileim/database" folder, where the cipher data file is named "MD5(ID)-wx4", as shown in Figure 2. MD5(ID) is the MD5 hash value, and the ID is the Wangxin user account, so if more than one Wangxin account has logged in before, there will be multiple files named "MD5(ID)-wx4" in the "database" folder.

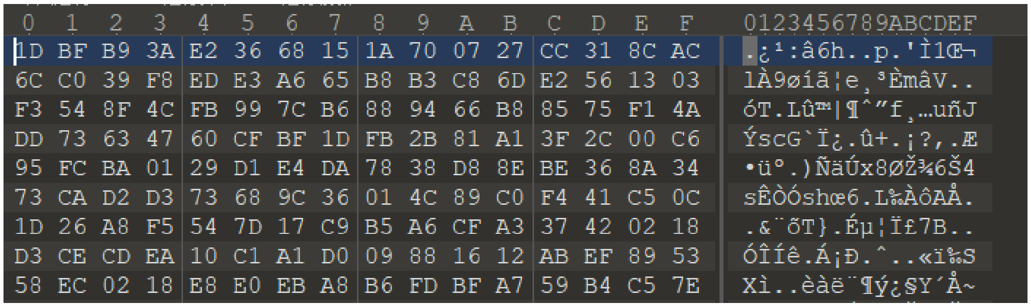


```
10.46GB 已用, 1.57GB 可用, r/w
..
上层文件夹
5340e5b970b12f5cb281c0290b6e732f
wx4
11 5月 20 14:47:47  63.00K  rw-------
5340e5b970b12f5cb281c0290b6e732f
wx4-encrypt
11 5月 20 14:47:47  0 字节  rw-------
a9d5b9e2f1cff71c52c8c697a77cc244w
x4
13 5月 20 14:28:12  115.00K  rw-------
a9d5b9e2f1cff71c52c8c697a77cc244w
x4-encrypt
11 5月 20 14:47:34  0 字节  rw-------
allaccounts
11 5月 20 14:44:32  6.00K  rw-------
allaccounts_v1
13 5月 20 14:30:14  6.00K  rw-------
allaccounts_v1-encrypt
11 5月 20 14:44:32  0 字节  rw-------
allaccounts-encrypt
11 5月 20 14:44:32  0 字节  rw-------
d42607dafa05e5b6927b950656
```

Figure 2. Wangxin application cipher data file.

4.2. Encryption mechanism

By reverse analysis of the Wangxin application, the Wangxin database uses an encryption tool called "sqlcrypto" to encrypt the entire database file, making the resulting database a completely unreadable cipher file. An encrypted Wangxin database is shown in Figure 3, from which you can see that the file header is also encrypted.



```
 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
1D BF B9 3A E2 36 68 15 1A 70 07 27 CC 31 8C AC   .¿¹:â6h..p.'Ì1Œ¬
6C C0 39 F8 ED E3 A6 65 B8 B3 C8 6D E2 56 13 03   lÀ9øíã¦e¸³ÈmâV..
F3 54 8F 4C FB 99 7C B6 88 94 66 B8 85 75 F1 4A   óT.Lû™|¶ˆ"f¸…uñJ
DD 73 63 47 60 CF BF 1D FB 2B 81 A1 3F 2C 00 C6   ÝscG`Ï¿.û+.¡?,.Æ
95 FC BA 01 29 D1 E4 DA 78 38 D8 8E BE 36 8A 34   •üº.)ÑäÚx8ØŽ¾6Š4
73 CA D2 D3 73 68 9C 36 01 4C 89 C0 F4 41 C5 0C   sÊÒÓshœ6.L‰ÀôAÅ.
1D 26 A8 F5 54 7D 17 C9 B5 A6 CF A3 37 42 02 18   .&¨õT}.Éµ¦Ï£7B..
D3 CE CD EA 10 C1 A1 D0 09 88 16 12 AB EF 89 53   ÓÎÍê.Á¡Ð.ˆ..«ï‰S
58 EC 02 18 E8 E0 EB A8 B6 FD BF A7 59 B4 C5 7E   Xì..èàë¨¶ý¿§Y´Å~
```

Figure 3. An encrypted Wangxin database file header.

By analyzing the “libdatabase_sqlcrypto.so” file of the Wangxin application, we can obtain the encryption mechanism of the Wangxin database.

Encryption mechanism: sqlcrypto specifically uses “sqlit3” related functions to encrypt the database file, where the encryption algorithm is AES, and the process of implementing this algorithm uses Electronic Codebook Book (ecb mode), and key of 128 bits length.

4.3. Extracting the decryption key

Through detailed analysis, we can see that the key generation function is encapsulated in the dynamic library “libdatabase_sqlcrypto.so” by NDK, so it is impossible to get the key accurately by static analysis, so we need to use dynamic binary instrumentation technique to get the decryption key while the program is running. Therefore, we need to use the dynamic binary instrumentation technique to get the decryption key during the running of the program. The key extraction process is shown in Figure 4, and the steps are as follows.

(1) Installed frida module in the cell phone and PC respectively, where the phone is installed server program, is through the injection process to achieve the hijacking of the application function, the PC side is mainly used for communication;

(2) Analysis to locate the need for binary staking functions: through analysis, “libdatabase_sqlcrypto.so” in the aes_encrypt_key128 () and aes_decrypt_key128 () function that is the Hook point; for the Hook point to write Frida scripts based on python and javascript for communication, where the python language is mainly sends the javascript code to the device, javascript code for the core code of the hook, including getting parameters and return values or modify the parameters and return values, etc.;

(3) In the cell phone side to run frida-server, and use adb to connect the phone to the PC side, and then execute the Wangxin application, at the same time need to run the python script file in the PC side, you can get the local database decryption key in the process of running the program.

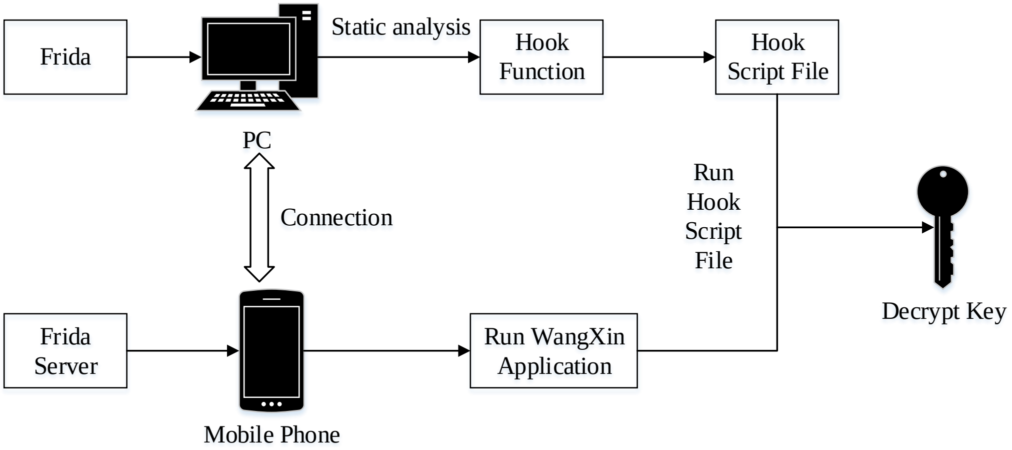


Figure 4. The decrypt key extraction process.

5. WANGXIN APPLICATION LOCAL DATABASE DECRYPTION

Once we get the correct decryption key, we can decrypt the ciphertext database according to the encryption principle of “sqlcrypto”. When decrypting, we need to pay attention to the setting of parameters, including decryption algorithm, decryption key, database file, etc.

Once the database file is decrypted correctly, we can parse the database file to get the user chat content. The list of friends is stored in the table “user”, including UserId and nickname; the chat messages are stored in the table “messages”, including messageId, conversationId, content and time, etc.; “onversation” table stores all conversations of the user, including messageTime and readTimestamp; “ExpressionPkgMain” table stores the expressions used by the user, including custom expressions and team expressions, etc.

6. CONCLUSION

In this paper, we studied the encryption protocol of Wangxin application, proposed a reverse analysis scheme for Wangxin application, detailed analysis of data encryption algorithm, and gave a method to obtain the decryption key under Android platform, detailed decryption and restoration of the local encrypted chat log database of Wangxin application. This study can provide an important reference for investigators as well as researchers in data forensics.

ACKNOWLEDGMENTS

This work is supported by Sichuan Science and Technology Program. The authors also acknowledge the reviewers for their useful opinions to improve this paper.

REFERENCES

[1] Li, Y., Liu, W., Zhang, Y. and Liang, L., “Analysis of test message transmitting protocol in popular instant messaging software,” Application Research of Computers, 22(7), 243-5+250 (2005). (in Chinese)

[2] Ni, L. and Shuang, K., “Android Wechat network behavior analysis,” [EB/OL], http://www.paper.edu.cn/releasepaper/content/201309-223, (2013). (in Chinese)

[3] Wang, Y., Gu, Y. and Qiu, W., “Research on interactive protocol and encryption mode of Wechat,” Microcomputer Applications, 2, 31-34 (2015).

[4] Zhang, Y. J., Yu, F. and Ji, Q. B., “The forensic analysis of WeChat message,” 2016 Sixth Inter. Conf. on Instrumentation & Measurement, Computer, Communication and Control, Harbin, China, (2016).

[5] Zheng, X., [The Research and Design of an Android Applications Decompiler], Beijing Institute of Technology, Beijing, Master’s Thesis, (2016). (in Chinese)

[6] Liu, Y. and Zhu, G., [Android Security and Decompilation Practice], Tsinghua University Press, Beijing, (2015). (in Chinese)

[7] Zhang, Z., Wang, Y., Weng, Y. and Mi, B., “Research on reverse analyzing of Android Application,” Information Network Security, (6), 65-68 (2013).

[8] Feng, S., Android Software Security and Reverse Analysis, Beijing Posts & Telecom Press, Beijing, (2013).

[9] Cinar, O., [Pro Android C++ with the NDK], Apress, USA, (2012).

[10] Nethercote, N., [Dynamic Binary Analysis and Instrumentation or Building Tools Is Easy], University of Cambridge, Cambridge, (2004).

[11] Luk, C. K., Cohn, R. and Muth R., “Pin: Building customized program analysis tools with dynamic instrumentation,” ACM Sigplan Notices, 40, 190-200 (2005).

[12] Bruening, D. L., [Efficient, Transparent, and comprehensive Runtime Code Manipulation], Massachusetts Institute of Technology, Massachusetts, Master’s Thesis, (2004).

[13] Nethercote, N. and Seward, J., “Valgrind: A framework for heavy weight dynamic binary instrumentation,” ACM Sigplan Conf. on Programming Language Design & Implementation, 89-100 (2007).