OTF Security Assessment

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The **National Anti-Fraud Cent**er app is an Android and iOS application developed by **China's Ministry of Public Security**. According to reports many users have been forced to download the app to access things such as apartments, concerts, and driving tests. The information discussed in this report is specifically in regard to the iOS application; the Android application was not examined. The OTF red team was asked to review and analyze the application within a static time window to make as many observations related to privacy and security as possible within the time window.

The application is advertised as a utility to help detect and alert users to fraudulent calls, texts, and applications installed on the device. This type of functionality requires high-level permissions, which allow the application to perform actions such as access call logs and query all applications. The application utilizes many additional sensitive permissions as well; including but not limited to accessing location, using the camera and recording audio.

The application is only available for Apple accounts with China based locations and cannot be downloaded from accounts in other countries.
The next table summarizes the permissions requested by the application and the description provided by the developers on the reason for the request.

Permission	Meaning	Description given by the developer
NSCameraUsageDescription	Access the Camera	Access your camera, in order to perform real-name authentication or OCR text recognition and scan recognition and other functions
NSLocationAlwaysUsageDescription	Access location information at all times.	Location is required to find out where you are
NSLocationWhenInUseUsageDescription	Access location information when app is in the foreground	In order to provide you with anti-fraud knowledge and services in the corresponding region, turn on "Location Services" to determine your location
NSMicrophoneUsageDescription	Access microphone.	Access to your microphone for functions such as audio recording
NSPhotoLibraryUsageDescription	Access the user's photo library.	Visit your image library, in order to be able to choose the image you want to upload

The following table summarizes the information relating to the different releases of the application.

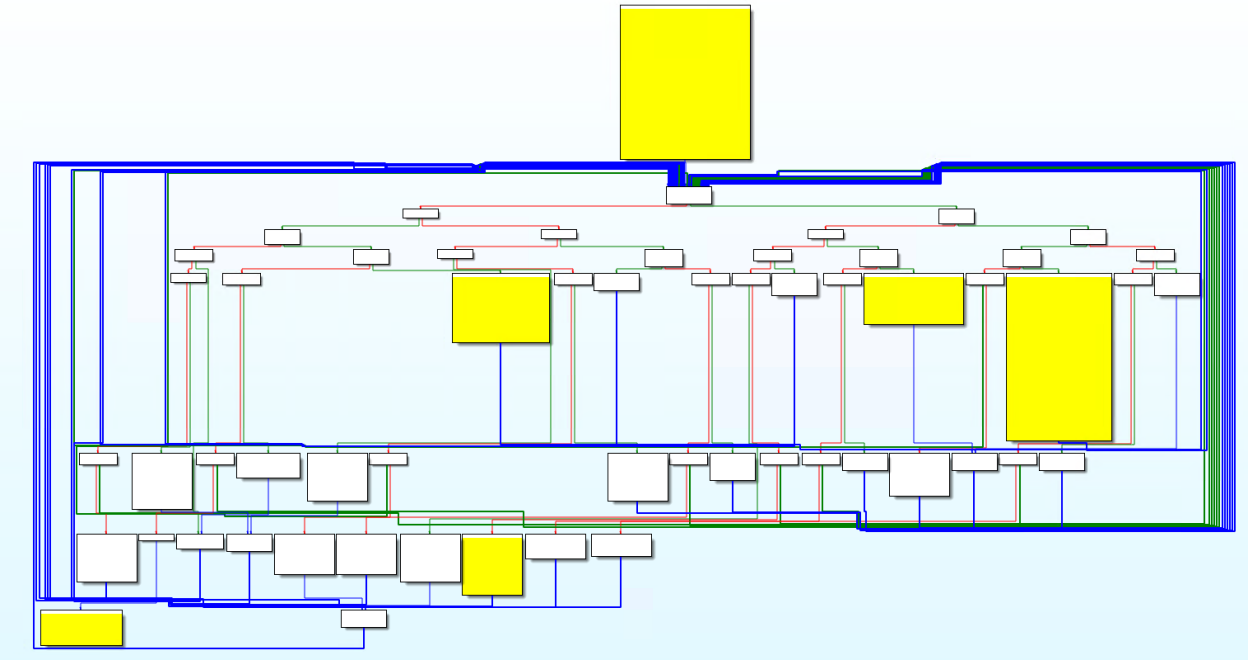
Version	Release date	Description
1.1.21	2022-01-31	Performance tuning, optimized experience
1.1.20	2022-01-24	Performance tuning, optimized experience
1.1.19	2022-01-10	Propaganda module optimization n2, performance tuning, and experience optimization
1.1.18	2021-11-20	Home page module optimization n2, announcement module optimization n3, performance tuning, and experience optimization
1.1.17	2021-11-06	Start page optimization n2, registration login module optimization n3, home page module optimization n4, I want to report module character limit optimization n5, report assistant module character limit optimization n6, performance tuning, and experience optimization
1.1.16	2021-10-22	Report assistant optimization n2, new case optimization n3, APP button anti-combustion limit optimization n4, performance tuning, and experience optimization
1.1.15	2021-10-01	I want to report module function optimization n2, slider verification module function new n3, statistical optimization n4, performance tuning, and experience optimization
1.1.14	2021-09-19	Function optimization of registration and login module n2, performance tuning, and optimized experience

1.1.13	2021-09-16	Home page module performance optimization rn2, login module function optimization rn3, my module personal information page function optimization rn4, performance tuning, experience optimization r
1.1.12	2021-09-01	I want to report module operation selection optimization rn2, performance tuning, and experience optimization
1.1.11	2021-08-06	Fix known bugs 2. Performance tuning, optimize experience n
1.1.10	2021-08-05	Visual optimization of home page module n2, new announcement function of home page module n3, experience optimization of incoming call warning function n4, experience optimization of registration and login module n5, function optimization of report assistant module n6, scam exposure module simplified publicity classification n7, performance tuning, optimization experience n
1.1.9	2021-07-21	The login password setting supports special characters n2, performance tuning, and optimized experience
1.1.8	2021-07-08	Function optimization of real-name authentication module n2. Function optimization of identity verification module n3. Performance optimization and experience optimization
1.1.7	2021-06-30	Function optimization of identity verification module n2, new logout function of account module n3, new voice upload n4 of report assistant module, optimization of image upload of report assistant module, signature submission process optimization n5, performance tuning, and optimized experience
1.1.6	2021-06-08	SDK initialization configuration optimization n2, scam exposure module optimization n3, performance tuning, and experience optimization n
1.1.2	2021-05-08	Open the identity verification function, one-click to clarify the real identity of the other party n2, the new account clearing cache function, speed up the application experience n3, performance tuning, optimize the experience
1.1.1	2021-04-12	Function optimization and adjustment
1.1.0	2021-03-23	Police officer terminal function optimization n2, support for message push function n3, new sharing function n4, optimization and adjustment of other functions
1.0.1	2021-03-12	Function optimization and performance tuning
1.0.0	2021-03-10	Initial Release

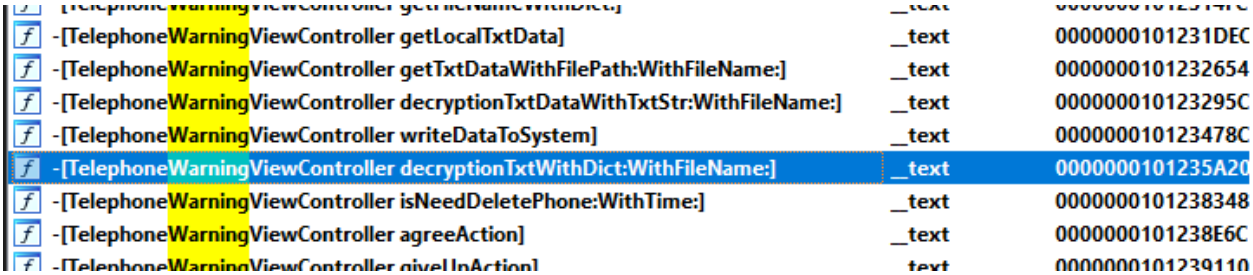
1. Code obfuscation

In the native application functions (the ones not related to third-party code) an obfuscation system based on the insertion of “dead code” was used together with the creation of loops that make the logical application flow difficult to follow.

The next image represents a visual display of a function within the application at a logical control flow oriented machine code level. Each box shown is a part of the executable code and the arrows are the logical control flows. The parts highlighted in yellow are the real code of the application, everything is associated with the obfuscator. As such it is seen that the authors of the application have taken steps to make the binary more difficult to analyze and understand.



Nonetheless, **no obfuscation of the names of the functions or of the text contained within the functions was found**, which make it possible to identify the application functions of interest.



2. Restricted access to the application from Tor network

The two main backend sites (fzapp.gjfzpt.cn and fzapph5v1.gjfzpt.cn) are not accessible from the the entire Tor (The Onion Router) network, in particular the connections cannot be established by most of the exit nodes.

However, it has been found that in some cases these connections are possible, therefore it is plausible that the dynamic updates of the exit node list is not completely done. Perhaps a point-in-time IP based access control list deployed.

This configuration is in line with the Chinese government's previously known limitations of the Internet; however it highlights how they still want to protect these services from possible attacks.

All requests made by the application to host “fzapph5v1.gjfzpt.cn” are answered with “403 Forbidden” both using Tor and also when using some VPN services. It is unclear if this is because the backend identifies that we are coming from non-china related IP addresses. More research could be done here to enumerate all allowed networks world-wide.

References:

https://www.torproject.org/

3. Protection of communications

Communications to the backends of the various services take place via HTTPS protocols and certificate pinning is implemented on some network connections to limit the possibility of data interception. All communications to the main backends (see the list below) are protected via certificate pinning.

“Appendix I: Frida script pinning bypass” contains the Frida script able to bypass all certificate pinning in the iOS application.

In iOS the application main backends are:

- fzapp.gjfzpt.cn
- fzapph5v1.gjfzpt.cn (sometimes used but appears to not always work correctly)

Furthermore, the **communications that take place towards the endpoints of the “fzapp.gjfzpt.cn” application are further protected by an additional layer of symmetric encryption (AES256)** which does not allow access to the data actually transmitted even in the case in which the certificate pinning can be circumvented.

Example of encrypted request.

```
POST /hicore/api/Account/login HTTP/1.1
Host: fzapp.gjfzpt.cn
Content-Type: application/json; charset=utf-8
Market: appStore
App-Version-Code: 66
App-Version: 1.1.21
Nodeid: 110101
Um-Devicetoken: a7b4ca2ebe0d37bfe8db077201286651e26e80a0dd60bd4ee716a253a579c672
Os-Model: iPhone 6s Plus
Api-Version: 163
Accept: */*
Accept-Language: en-GB;q=1
Accept-Encoding: gzip, deflate, br
Os-Type: 1
Deviceid: 221d0f8d965f8bd9153aaf05531dcc1c8eb4b141
User-Agent: guo jia fan zha zhong xin/1.1.21 (iPhone; iOS 13.6.1; Scale/3.00)
Content-Length: 188
Os-Version: 13.6.1
Connection: close

{"sign":"f435144e0e421b3b4f89b0a2c40bc8f6","data":"0wdYQu5QWdkVyOp4U\/I+6JfDvzG11Mik08u1I1jX5ZDk8Sik\/3x9DPbODsfSkuoWbuflYK0CUabv4BMPs5Rwt4BD0cw9RGBxM1SoGh+KaPo=","timestamp":"1648045216"}
```

“data” contains the Base64 of encrypted data, which is generate in this way:

```
sign = md5(cleartext).hexdigest()
iv = md5(timestamp).hexdigest()[8:24]
key = md5(sign)
data = base64( AES(cleartext,iv,key) )
```

This was defeated and further in this report “Appendix II: Burp Suite decrypter plugin” contains a Burp Suite plugin able to decrypt the transmitted data.

4. Usage of free China based phone number

The application has a functionality wall such that the majority of functionality can not be accessed without a China based phone number. As such, Dynamic analysis has been severely limited by not having access to the application's authenticated area, therefore the requests that could have been analyzed for dynamic analysis are very small.

For this reason, the decision was made to try to register through some services that provide free Chinese phone numbers for receiving SMS.

The team used numerous free SMS services (for example https://7sim.net/) but unfortunately the receipt of the registration confirmation SMS which are necessary to enable the accounts never occurred. It is not clear if the confirmation SMS did not arrive or if these services did not work properly. It is possible that the Chinese state has taken a census of these free numbers and limits their access to its services. As such more research can be done here to bypass the check, or by using a china based phone number directly for the end goal of furthering dynamic analysis of the application.

5. Use of unusual libraries

The analysis found that part of the app code is based on OpenUDID library, this library is to be considered obsolete and by now Apple natively provides APIs for the generation of unique identifiers.

The peculiarity of this library, however, is the use of customized Pasteboards to keep the generated identifiers in memory.

In recent years, Apple has carried out a whole series of activities aimed at limiting the possibility of creating unique identifiers that can allow even unintentionally to trace a device between different installations.

It is not clear whether the choice to use this library is given by habit or by the willingness to try to circumvent some of Apple's privacy mechanisms to allow the identification of a device through different installations.

Furthermore, the data present in these Pasteboards could also be accessible by other applications that have the same Team ID, thus allowing the correlation of data from different applications to be carried out.

Some of these local mobile handset identifiers are transmitted both to the application's backend servers and to other sites that are used to carry out usage statistics. It is not clear whether the Chinese government will subsequently have access to all this statistical data in order to correlate the information and have a more complete view of what the user is doing.

A more in-depth dynamic analysis that includes a Chinese telephone number and preferably a Chinese source IP could make it possible to verify that these identifiers are not used in particularly critical parts such as those of facial recognition. If that is not possible then either deobfuscation of the iOS app, or review of the pre-compilation source code of the app would be necessary. With the current static of reviewing the obfuscated , reverse engineering and further static analysis can proceed as about ~10% the speed of normal app analysis due to the overhead imposed by the obfuscator.

Classes based on OpenUDID are:

- OpenUDID
- UTDIDOpenUDID
- UMOpenUDID
- UMSocialOpenUDID

References:

Original source code of OpenUDID – https://github.com/ylechelle/OpenUDID
Apple Pasteboard – https://developer.apple.com/documentation/uikit/uipasteboard

6. Frameworks included in the IPA

The application contains numerous third-party libraries that are incorporated into the IPA file. These are mainly of two different types:

- libraries that remain "external" to the executable
- libraries that are "embedded" in the executable

External libraries are files inside the IPA that are loaded runtime by the system and can be called from the main executable.

Directory	Name	Description
Frameworks\AipBase.framework\	com.baidu.AipBase.AipBase	Baidu AI Platform
Frameworks\AipOcrSdk.framework\	com.baidu.AipOcrSdk	Baidu AI Platform
Frameworks\IdcardQuality.framework\	com.baidu.IdcardQuality	Baidu AI Platform
PlugIns\CallDirectory.appex\	com.AppStore.NationalFraudCenterHicode.CallDirectory	Same developer plugin

All iOS operating system executables are encrypted upon download from the AppStore. The system will decrypt the areas of interest at the kernel level while this is being done. For this reason, if an IPA that has just been downloaded from the AppStore is analyzed the machine code will not be easily seen. To overcome this problem there are multiple different techniques that dump the executables once they have been decrypted by the operating system.

In this case it was not possible to decrypt the "CallDirectory" file, it is not clear if this is due to additional protection or why it is loaded in a particular way by the main executable, and this happens only after logging in.

Based on the plaintext strings present in the executable, it is assumed that the library in question presents itself as having functionality to handle interacting with incoming calls to check if these are spam or not. The library in fact recalls the "CallKit" framework of Apple.

Based on the class names, filenames and other information present within the binary it can be assumed that the following SDKs were used within the application IPA itself.

Name	Site	Description
Sensetime Finance	https://www.sensetime.com/en	AI on financial transaction
Cloudwalk Face SDK	https://www.cloudwalk.com/en/Technology	Face Recognition
dpnet-ios	Unknown	Unknown
publish v7.1.0 – pplwrapper	Unknown	Unknown
cloudlan	Unknown	Unknown

alyun-oss-ios-sdk	https://github.com/aliyun/alibabacloud-oss-sdk	Alibaba Cloud OSS
Wechat SDK	https://open.weixin.qq.com/?lang=en	WeChat Integration SDK
Weibo sdk	https://open.weibo.com/wiki/SDK	Weibo SDK
OpenInstallSDK	https://www.openinstall.io/download.html	Openinstall SDK
ShareInstallSDK	http://www.shareinstall.com.cn/	Shareinstall SDK
umeng SDK	https://developer.umeng.com/	Umeng SDK
Amap SDK	https://lbs.amap.com/	Amap SDK

The presence of a **large number of different SDKs, including several that offer OCR, Face Recognition, Voice recognition and similar features is an important point of attention, these technologies can be used correctly for the application purpose, but they could also be used for malicious purposes without the user being notified**.

7. Privacy Policy

The privacy policy can be consulted at: https://fzapph5-chanct-cn.translate.goog/Agreements/policy.html

The privacy policy specifies that the information taken from the app is the following:

- Anything provided voluntarily by the user
  - Your mobile phone number, name and ID number filled in by the user;
  - Camera permissions (We will access the front camera of your mobile phone for facial feature recognition to obtain your facial recognition features)
  - You can also choose to fill in the information such as the region according to your own needs.
  - Information you actively upload when using the reporting service provided by the APP
  - The suspected fraudulent mobile phone numbers, call records, text messages, APPs, pictures, websites, audio and video, screen recordings, social accounts, trading accounts, etc. The sources of information include:
  - Contact information and call history
  - SMS message
  - Application file information
  - image information
  - Audio information
  - Video information
  - Microphone permissions
  - Capture screen display content
  - Storage permissions
  - Location permissions
  - The information you provide when using the identity verification function provided by the app – your mobile phone number, name, ID number filled in by yourself;
  - Camera permission – your face image.
- Automatically taken from the app:
  - Device Information (Information such as the brand, device model, system version, and device identification code of your device)

The most controversial part of the privacy policy is the following sentence:

This Privacy Policy applies only to any information we collect, and does not apply to the services provided by any third party or the rules for the use of information by third parties, and we are not responsible for any third party's use of the information provided by you. For the privacy policy of third-party services, please refer to Antiy Mobile Security AVL SDK Privacy Policy and Youmeng+ Privacy Policy.

Therefore, by carrying out a recursive analysis, the following situation arises:

- Main privacy policy
  - Antiy Mobile Security AVL SDK (https://www.avlsec.com/zh-hans/privacy-policy)
  - Umeng (https://www.umeng.com/policy)
    - Agoo SDK (https://terms.alicdn.com/legal-agreement/terms/suit_bu1_taobao/suit_bu1_taobao201703241622_61002.html)
    - Anti-hijacking SDK (http://terms.aliyun.com/legal-agreement/terms/suit_bu1_ali_cloud/suit_bu1_ali_cloud201902141711_54837.html)

However, the analysis highlighted how many more third-party services are used and no direct or indirect reference to them has been identified. For example OpenInstall SDK communicate some data to their backend but it's not mentioned in any privacy policy (https://www.openinstall.io/doc/rules.html).

From the analysis of the privacy policy there was not found a violation of the new national legislation of protecting user personal data (https://digichina.stanford.edu/work/translation-personal-information-protection-law-of-the-peoples-republic-of-china-effective-nov-1-2021/). However, a legal analysis should be carried out to verify that all preconditions are met. It is also possible that some translations carried out by automatic tools contain errors.

Furthermore, Personal Information Protection Law of the People's Republic of China appears to be a very general legislation of which some points can be interpreted in many ways, for example "personal information" in the law appeared to be defined as follows:

Personal information is all kinds of information, recorded by electronic or other means, related to identified or identifiable natural persons, not including information after anonymization handling.

8. "Localall.txt" file

Based on stake holder request, the **"localall.txt"** file was analyzed. The file is encrypted and used by the application.

The file is decrypted at runtime by the function:

```
-[TelephoneWarningViewController getLocalTxtData]
```

which in turn further uses other routines of the TelephoneWarningViewController class to perform the decryption.

The algorithm used for the encryption is "AES256" and the encryption keys are as follows.

```
Encryption Key:
            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
2835a89f0  36 36 37 65 33 39 33 39 31 32 65 35 38 64 32 36  667e393912e58d26
2835a8a00  38 30 39 38 33 32 34 34 32 39 33 30 34 39 34 35  8098324429304945
iv:
            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
283dd8ec0  64 30 33 31 35 36 33 64 62 64 62 31 36 30 30 61  d031563dbdb1600a
```

Even in the latest version of the executable it is possible to use the *"decrypt-localall.py" * script previously developed by Citizen Lab to perform the decryption.

The TelephoneWarningViewController class is directly connected with the “CRCallDirectoryManager” class which in turn is connected with the external executable “CallDirectory”, based on the application code and on the functionalities to which these classes are connected it is assumed that the application uses the **“localall.txt”** file to have a database of numbers considered fraudulent and/or spam and uses them to warn the user.

This hypothesis is also confirmed by the data present in the **“localall.txt”** file which, once decrypted, in addition to the telephone numbers, has a description some examples of which are reported below:

- Reported as a fraud by impersonating the Public Procuratorate
- Reported as a refund scam posing as a shopping customer service

9. Endpoint and information collected

Description:

In the iOS application the main backends are:

- fzapp.gjfzpt.cn
- fzapph5v1.gjfzpt.cn (used as backup but doesn’t work correctly)

The application communicates also with the following hosts:

- aaid.amap.com
- dualstack-arestapi.amap.com
- ios.bugly.qq.com
- api.openinstall.io
- msg.umengcloud.com
- ulogs.umeng.com
- api.weibo.com
- log.umsns.com
- ulogs.umengcloud.com

Further in this report please find “Appendix III. List of hostname and IP address" a list of all hostname identifiable from the binaries.

Based on the data intercepted between the device and the two main backends a lot of data is sent to/from the application. On all requests from the app the following HTTP headers are added.

Header Name	Content
Um-Devicetoken:	UID Generated from UMeng
User-Agent	User agent of the application which contains also the version of the iPhone
Os-Model	iPhone Model
Market	Application source market
Deviceid	UID Generated from UMeng
Api-Version	Api version
Os-Type	os type
App-Version	Application version
App-Version-Code	Application version
Os-Version	os Version

This information is managed and added by the class “HttpRequestAES” and in particular by the function “setSessionManagerCommonHeader:”

A point to call attention to is the fact that some IDs generated by the application are shared to different services, so hypothetically an attacker able to get the information from all sources could be able to correlate more details about the user.

Please note further in this report “Appendix IV. API Endpoint" contains a list of identifiable backend API endpoint.

Other information is transmitted from the libraries to the hosts mentioned above, such as.

Host	Information
aaid.amap.com	application id, os version, iphone model, umeng id
ios.bugly.qq.com	application id, os version, iphone model, umeng id
api.openinstall.io	application id, application version, os version, iphone model, local ip addresses, installation id
msg.umengcloud.com	application id, application version, os version, iphone model, language, country, screen resolution, jailbreak status, umeng id, connection type, accier name
api.weibo.com	application id,umeng id
log.umsns.com	application id, application version, os version, iphone model. connection type

10. Summary of Privacy/Security Implications for Users

The analyzed application's published feature goals are to perform various “anti-fraud checks”, based on a large amount of various data that the user shares with the application itself granted via the application's permissions manifest. For the correct “function” of the application the user must continuously send updated data passively via the application's data gathering features, the application publicly requests the user's permission to access this information and the user consciously must decide to share it with the application.

When this data is shared with the application it is sent to the backend servers, at which point it is not possible to know if this information is processed solely and exclusively for the purpose of anti-fraud or even for any other purpose.

The presence of a large number of software components, including several that offer OCR, Face Recognition, Voice recognition and similar features is an important point of attention. These technologies can be used correctly for the application purpose, but they could also be used for malicious purposes without the user being notified.

The analysis highlighted some anomalies in the use of obsolete and particular libraries that could further facilitate the correlation of data from different apps of the same developer, also the absence of some third parties within the privacy policy was found, which therefore they may receive user data without this being informed.

As a concluding thought….

Unfortunately, because of the application features and data that the user voluntarily decides to share, it becomes difficult to understand and evaluate if there is a risk for users who use this application. The trust falls solely on the publisher of the application as there were no major indicators (found within the time allocated to the application exploration) that would indicate any strong red or green flags.

APPENDICES

Appendix I – Frida script pinning bypass

```
if (ObjC.available) {
    var errSSLServerAuthCompleted = -9481;
    var kSSLSessionOptionBreakOnServerAuth = 0;
    var noErr = 0;

    var SSLHandshake = new NativeFunction(
        Module.findExportByName("Security", "SSLHandshake"),
        'int',
        ['pointer']
    );

    Interceptor.replace(SSLHandshake, new NativeCallback(function (context) {
        var result = SSLHandshake(context);
        if (result == errSSLServerAuthCompleted) {
            send("Replacing SSLHandshake");
            return SSLHandshake(context);
        }
        return result;
    }, 'int', ['pointer']));

    var SSLCreateContext = new NativeFunction(
        Module.findExportByName("Security", "SSLCreateContext"),
        'pointer',
        ['pointer', 'int', 'int']
    );

    Interceptor.replace(SSLCreateContext, new NativeCallback(function (alloc, protocolSide, connectionType) {
        send("Replacing SSLCreateContext");
        var sslContext = SSLCreateContext(alloc, protocolSide, connectionType);
        SSLSetSessionOption(sslContext, kSSLSessionOptionBreakOnServerAuth, 1);
        return sslContext;
    }, 'pointer', ['pointer', 'int', 'int']));

    var SSLSetSessionOption = new NativeFunction(
        Module.findExportByName("Security", "SSLSetSessionOption"),
        'int',
        ['pointer', 'int', 'bool']
    );

    Interceptor.replace(SSLSetSessionOption, new NativeCallback(function (context, option, value) {
        if (option == kSSLSessionOptionBreakOnServerAuth) {
            send("Replacing SSLSetSessionOption");
            return noErr;
        }
        return SSLSetSessionOption(context, option, value);
    }, 'int', ['pointer', 'int', 'bool']));

    //
    // OLD WAY
    //
    var kSecTrustResultInvalid = 0;
    var kSecTrustResultProceed = 1;
    var kSecTrustResultDeny = 3;
    var kSecTrustResultUnspecified = 4;
    var kSecTrustResultRecoverableTrustFailure = 6;
    var kSecTrustResultFatalTrustFailure = 6;
    var kSecTrustResultOtherError = 7;

    var SecTrustEvaluate = new NativeFunction(
        Module.findExportByName("Security", "SecTrustEvaluate"),
        'int',
        ['pointer', 'pointer']
    );

    Interceptor.replace(SecTrustEvaluate, new NativeCallback(function (trust, result) {
        send("Replacing SecTrustEvaluate");
        var ret = SecTrustEvaluate(trust, result);
```

```
        result = kSecTrustResultProceed;
        return ret;
    }, 'int', ['pointer', 'pointer']));

    //
    // COMMON FRAMEWORKS
    //
    /* AFNetworking */
    if (ObjC.classes.AFSecurityPolicy) {
        Interceptor.attach(ObjC.classes.AFSecurityPolicy['- setSSLPinningMode:'].implementation, {
            onEnter: function (args) {
                send("Replacing AFSecurityPolicy setSSLPinningMode = 0 was " + args[2]);
                args[2] = ptr('0x0');
            }
        });
        Interceptor.attach(ObjC.classes.AFSecurityPolicy['- setAllowInvalidCertificates:'].implementation, {
            onEnter: function (args) {
                send("Replacing AFSecurityPolicy setAllowInvalidCertificates = 1 was " + args[2]);
                args[2] = ptr('0x1');
            }
        });
    }

    /* Kony */
    if (ObjC.classes.KonyUtil) {
        Interceptor.attach(ObjC.classes.KonyUtil['+ shouldAllowSelfSignedCertificate'].implementation, {
            onLeave: function (retval) {
                send("Replacing KonyUtil shouldAllowSelfSignedCertificate = 1 was " + retval);
                retval.replace(0x1);
            }
        });
        Interceptor.attach(ObjC.classes.KonyUtil['+ shouldAllowBundledWithSystemDefault'].implementation, {
            onLeave: function (retval) {
                send("Replacing KonyUtil shouldAllowBundledWithSystemDefault = 1 was " + retval);
                retval.replace(0x1);
            }
        });
        Interceptor.attach(ObjC.classes.KonyUtil['+ shouldAllowBundledOnly'].implementation, {
            onLeave: function (retval) {
                send("Replacing KonyUtil shouldAllowBundledOnly = 0 was " + retval);
                retval.replace(0x0);
            }
        });
    }

    var os_version5 = ObjC.classes.AFSecurityPolicy["- evaluateServerTrust:forDomain:"];

    Interceptor.attach(os_version5.implementation, {
        onEnter: function(args) {
                console.log('---------------- > AFSecurityPolicy evaluateServerTrust > ----------------');
                console.log(ObjC.Object(args[3]).toString())
        },
        onLeave: function (retval) {
                console.log('---------------- < AFSecurityPolicy evaluateServerTrust < ----------------');
                console.log(retval)
                retval.replace(0x1)
        }
    });
```

Appendix II – Burp Suite decrypter plugin

```
# Code partially copied from https://github.com/parsiya/Parsia-Code/blob/master/python-burp-crypto/3-jython/

from burp import IBurpExtender
from burp import IMessageEditorTabFactory
from burp import IMessageEditorTab
from burp import IParameter
```

```
import sys
import json
import hashlib

# Jython imports
from javax.crypto import Cipher
from javax.crypto.spec import IvParameterSpec
from javax.crypto.spec import SecretKeySpec

from java.util import Base64

def get_md5_32(buf):
    return hashlib.md5(buf).hexdigest()

def get_md5_16(buf):
    return get_md5_32(buf)[8:24]

# getInfo processes the request/response and returns info
def getInfo(content, isRequest, helpers):
    if isRequest:
        return helpers.analyzeRequest(content)
    else:
        return helpers.analyzeResponse(content)

# getBody returns the body of a request/response
def getBody(content, isRequest, helpers):

    if isRequest:
        content = helpers.bytesToString(content)
    else:
        content = helpers.bytesToString(content)

    info = getInfo(content, isRequest, helpers)
    return content[info.getBodyOffset():]

# decode64 decodes a base64 encoded byte array and returns another byte array
def decode64(encoded, helpers):
    return helpers.base64Decode(encoded)

# encode64 encodes a byte array and returns a base64 encoded byte array
def encode64(plaintext, helpers):
    return helpers.base64Encode(plaintext)

# encryptJython uses javax.crypto.Cipher to encrypt payload with key/iv
# using AES/CFB/NOPADDING
def encryptJython(payload, key, iv):
    aesKey = SecretKeySpec(key, "AES")
    aesIV = IvParameterSpec(iv)
    cipher = Cipher.getInstance("AES/CBC/NOPADDING")
    cipher.init(Cipher.ENCRYPT_MODE, aesKey, aesIV)
    encrypted = cipher.doFinal(payload)
    return Base64.getEncoder().encode(encrypted)

# decryptJython uses javax.crypto.Cipher to decrypt payload with key/iv
# using AES/CFB/NOPADDING
def decryptJython(payload, key, iv):
    decoded = Base64.getDecoder().decode(payload)
    aesKey = SecretKeySpec(key, "AES")
    aesIV = IvParameterSpec(iv)
    cipher = Cipher.getInstance("AES/CBC/NOPADDING")
    cipher.init(Cipher.DECRYPT_MODE, aesKey, aesIV)
    return cipher.doFinal(decoded)

class BurpExtender(IBurpExtender, IMessageEditorTabFactory):

    #
    # implement IBurpExtender
    #

    def registerExtenderCallbacks(self, callbacks):
        # keep a reference to our callbacks object
        self._callbacks = callbacks

        # Parsia: obtain an extension helpers object
        self._helpers = callbacks.getHelpers()

        # set our extension name
        # Parsia: changed the extension name
        callbacks.setExtensionName("burp decrypter")
```

```
        # register ourselves as a message editor tab factory
        callbacks.registerMessageEditorTabFactory(self)

        # Parsia: for burp-exceptions
        sys.stdout = callbacks.getStdout()

    #
    # implement IMessageEditorTabFactory
    #

    def createNewInstance(self, controller, editable):
        # create a new instance of our custom editor tab
        return CryptoTab(self, controller, editable)

#
# class implementing IMessageEditorTab
#

class CryptoTab(IMessageEditorTab):
    def __init__(self, extender, controller, editable):
        self._extender = extender
        self._editable = editable
        # Parsia: Burp helpers object
        self.helpers = extender._helpers

        # create an instance of Burp's text editor, to display our decrypted data
        self._txtInput = extender._callbacks.createTextEditor()
        self._txtInput.setEditable(editable)

    #
    # implement IMessageEditorTab
    #

    def getTabCaption(self):
        # Parsia: tab title
        return "Decrypted"

    def getUiComponent(self):
        return self._txtInput.getComponent()

    def isEnabled(self, content, isRequest):
        return True

    def isModified(self):
        return self._txtInput.isTextModified()

    def getSelectedData(self):
        return self._txtInput.getSelectedText()

    def setMessage(self, content, isRequest):
        if content is None:
            # clear our display
            self._txtInput.setText(None)
            self._txtInput.setEditable(False)

        # Parsia: if tab has content
        else:
            # get the body

            body = getBody(content, isRequest, self.helpers)
            cont = json.loads(body)

            iv = get_md5_16(cont['timestamp'])
            key = get_md5_32(cont['sign'])

            try:
                decryptedBody = decryptJython(cont['data'], key, iv)
            except KeyError:
                decryptedBody = decryptJython(cont['sData'], key, iv)

            self._txtInput.setText(decryptedBody)

        # remember the displayed content
        self._currentMessage = content
```

Appendix III. List of hostname and IP address.

The location is based on https://www.ip2location.com/ service.

Hostname	IP	Location
aaid.amap.com	203.119.169.158	China
aaid.umeng.com	223.109.148.139	China
adiu.amap.com	59.82.29.155	China
ai.login.umeng.com	59.82.29.248	China
api.openinstall.io	163.181.56.173	Germany
api.weibo.com	114.134.80.166	Hong Kong
api.weixin.qq.com	203.205.239.82	China
apiinit.amap.com	47.246.110.95	Hong Kong
appsupport.qq.com	129.226.107.77	Hong Kong
apsctl.amap.com	59.82.14.192	China
blog.ibireme.com	172.104.87.16	Japan
bugly.qq.com	129.226.103.217	Hong Kong
c.umsns.com	59.82.29.248	China
cgicol.amap.com	120.77.134.87	China
check.shareinstall.com.cn	124.71.238.62	China
config.shareinstall.com.cn	124.71.238.62	China
developer.umeng.com	59.82.60.43	China
dns.qq.com	119.29.29.229	Hong Kong
dualstack-adiu.amap.com	59.82.31.100	China
dualstack-arestapi.amap.com	47.246.109.113	Hong Kong
dualstack-logs.amap.com	106.11.130.221	China
dualstack-m5.amap.com	120.77.134.208	China
fzapp.gjfzpt.cn	123.57.4.34	China
fzapph5v1.gjfzpt.cn	111.205.235.39	China
graph.qq.com	203.205.253.150	Hong Kong
help.wechat.com	203.205.239.179	China
hydra.alibaba.com	203.119.175.226	China
ios.bugly.qq.com	203.205.235.53	China
iploc.market.alicloudapi.com	120.78.168.91	China
log.umsns.com	59.82.31.92	China
logs.amap.com	106.11.23.110	China
long.open.weixin.qq.com	109.244.217.35	China
m.aliyun.com	203.119.207.129	China
m.weibo.cn	36.51.254.229	China
m5.amap.com	198.11.188.35	United States of America
media.weibo.cn	36.51.254.229	China
mobile.umeng.com	59.82.31.210	China
msg.umengcloud.com	59.82.58.90	China
national.fzlm.org.cn	114.80.187.5	China
oauth2.umeng.com	59.82.31.210	China
open.weibo.cn	114.134.80.166	Hong Kong
open.weixin.qq.com	203.205.239.154	China
openmobile.qq.com	203.205.239.162	China
oss.chanct.cn	121.14.45.21	China

oss.gjfzpt.cn	121.14.45.20	China
pslog.umeng.com	59.82.31.160	China
pv.sohu.com	52.156.120.137	United States of America
qm.qq.com	203.205.254.142	Hong Kong
qzs.qq.com	203.205.136.77	Hong Kong
restios.amap.com	47.246.110.95	Hong Kong
restsdk.amap.com	47.246.110.95	Hong Kong
rqd.uu.qq.com	203.205.239.17	China
service.weibo.com	36.51.254.229	China
stat.openinstall.io	123.57.46.86	China
statlog.shareinstall.com.cn	114.116.251.190	China
task.shareinstall.com.cn	124.71.238.62	China
ti.qq.com	203.205.254.62	Hong Kong
travis-ci.org	34.74.152.26	United States of America
ucc.umeng.com	203.119.169.175	China
ulogs.umeng.com	223.109.148.177	China
ulogs.umengcloud.com	47.246.109.108	Hong Kong
www.baidu.com	103.235.46.39	Hong Kong
www.qq.com	104.90.145.137	Germany
www.shareinstall.com.cn	163.171.133.124	United States of America
www.sina.com	47.246.20.229	United States of America
www.taobao.com	163.181.56.177	Germany

Appendix IV. API endpoint

- api/Feedback/GetDetails
- api/Verification/create
- api/area/getareajson?areaVersion =%@
- api/Account/bindaccount
- api/Account/changemobile
- api/Account/checkisverify
- api/Account/checksmscode
- api/Account/haspwd
- api/Account/login
- api/Account/logout
- api/Account/modifyregionv2
- api/Account/regist
- api/Account/userinfo
- api/Account/verify
- api/Account/verifyv2_1
- api/AppConfig/checkrenew?warningVersion=%&version=%&date=%@
- api/AppConfig/getalldictionary?dictionarykeys=%@
- api/AppConfig/getalldictionary?dictionarykeys=ProtorolVersion,SecretVersion
- api/AppConfig/getdictionary?dictionarykey=%@
- api/AppConfig/getdictionary?dictionarykey=ExamShare
- api/AppConfig/verifyversion
- api/AppVersion/ioscheck

- api/Area/checkareaversion?areaVersion=%@
- api/Area/treejson
- api/Banner
- api/CaseReport/CaseReportNumCurrentDay?submitterID=%@
- api/CaseReport/withoutreserve?recordid=%@
- api/ChannelStatistics/addchannel
- api/Concerns/getconcernslist
- api/DK/getcasecategorys
- api/EvidenceGather/withoutreserve?recordid=%@
- api/EvidenceType
- api/EvidenceType/getpaymenttypes
- api/Feedback
- api/Feedback/AddFeedBackv2
- api/Feedback/GetDetails
- api/File/cancelupdate?fileid=%@
- api/File/checkfilestatus
- api/File/endupload
- api/File/listenapp
- api/FraudGroup/addv2
- api/FraudGroup/removeleaguerv2
- api/HotInformation/gethotinformations?Page=1&Rows=30
- api/HotInformation/gethotinformationtoshare?id=%@
- api/Information/getinformationtoshare?informationID=%@
- api/Information/querylatestcases?Page=%ld&Rows=%ld
- api/Message/GetUnReadCount
- api/Message/SetRead?messageId=%@
- api/Notice/getlastestnoticeforuser
- api/PoliceUser/policelogin
- api/Popup/getpopup
- api/QA/getqalist
- api/QA/solve
- api/RealNameAudit/getauditinfo?number=%@
- api/RegionAccount/AppGetToken
- api/RegionApp/GetCityByPcode?pcode=%@
- api/RegionApp/GetOneRegionMain
- api/System/check?str=%&type=%
- api/Verification/verify
- api/XC/GetBackCaseCount
- api/XC/confirmwrite
- api/XC/getaccounttype
- api/XC/getdetails?id=%@
- api/XC/getpaymenttype
- api/XC/removepaymentdetail?id=%@
- api/XC/removesuspectfile?id=%@

- api/XC/removesuspectprintscreen?id=%@
- api/XC/removeurldetail?id=%@
- api/XC/removevictim?id=%@
- api/XC/sacanqrcode
- api/XC/savepayment
- api/XC/savesuspectrequet
- api/XC/saveurl
- api/XC/updateurldetailv3
- api/XC/uploadpaymentdetailv3
- api/XC/uploadsuspectfilev3
- api/XC/uploadsuspectprintscreenv3
- api/XK/addsmsinfo
- api/XK/deleteappinfo?id=%@
- api/XK/deleteconversation
- api/XK/deleteconversationdetail
- api/XK/deletepaymentinfo?id=%@
- api/XK/deletepersionnel?id=%@
- api/XK/deletetelrecord
- api/XK/deletetransferrecorddetail
- api/XK/deleteurlinfo?id=%@
- api/XK/getxkcasecategorys
- api/XK/savecaseinfo
- api/XK/savemobileinfo
- api/XK/savetransferrecord
- api/XK/sendsms
- api/XK/smsverify
- api/XK/uploadconversationdetail
- api/XK/uploadtransferrecorddetail
- api/account/sendidentitycode
- api/currentcount/statistic
- api/home/getreadpoint
- api/policeuser/sendsms
- api/file/upload
- api/CaseReport/getdetail
- api/CaseReport/getlist
- api/CaseReport/initialuploading
- api/CaseReport/iosreportapp
- api/CaseReport/removeapprecord
- api/CaseReport/submit
- api/EvidenceGather/getdetail
- api/EvidenceGather/getlist
- api/EvidenceGather/initialuploading
- api/EvidenceGather/iosreportapp
- api/EvidenceGather/removeapprecord

- api/EvidenceGather/submit
- api/EvidenceType
- api/EvidenceType/getsocialaccounttypes
- api/Feedback/AddFeedBack
- api/File/GetOssToken
- api/XC/deleteapp
- api/XC/getdocumenttypes
- api/XC/getedubg
- api/XC/getnations
- api/XC/getsocialaccounttypes
- api/XC/pagelist
- api/XC/removepayment
- api/XC/removesuspect
- api/XC/removeurl
- api/XC/saveappv3
- api/XC/savevictim
- api/XK/GetPersionnalNations
- api/XK/addappinfo
- api/XK/addpersionnel
- api/XK/addtepaymentinfo
- api/XK/addurlinfo
- api/XK/getpersionnaldocumenttypes
- api/XK/getpersionnaledubg
- api/XK/saveconversation
- api/XK/savetelnumber
- api/XK/searchdivisions
- api/XK/searchdivisions?codes=%@
- api/XK/updateappinfo
- api/XK/updatepaymentinfo
- api/XK/updatepersionnel
- api/XK/updateurlinfo
- api/file/upload
- api/xc/getxccasecategorys