XPOSED(VXP)/FRIDA Report of PoRE

Student ID　19300240012
Name　杨济瑞

- **Tasks List**

Write down the tasks list you finished and the corresponding score
1.酷狗音乐去除开屏广告（frida）
2.微信发送延时消息
3.微信控制筛子点数
4.微信自动回复
5.微信防止撤回
(Or more)

- **Project Demo Video**

  链接：https://pan.baidu.com/s/1tEd4KRwewXMsXcC3gTOjGQ
  提取码：1234

- **Task1 酷狗音乐去除开屏广告**
- **Introduction**

  酷狗 app 来源：来自实验手机的应用商店

  酷狗音乐_10.5.8_10589_com.kugou.android_main_standalone.apk

  酷狗音乐是一个使用人数众多的应用 app，在启动酷狗音乐时，酷狗的欢迎界面会显示一个广告，广告可以跳过，本 task 就是模拟点击广告跳过按钮，实现跳过广告（减少广告出现时间）

- **How you find the target function**

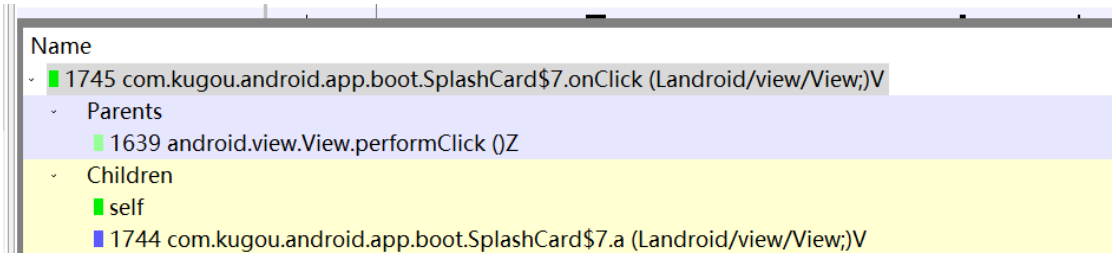
  Introduce how you reversed the target apk and located the target function you want to hook

  1.尝试用 DDMS 的 hierarchy Viewer

  成功使用 DDMS 的 hierarchy Viewer 截取到了开屏广告的按钮：

但是 resource-id 是空的，没有任何用处。

2. 尝试用 DDMS 的 method profiling



找到了这个函数

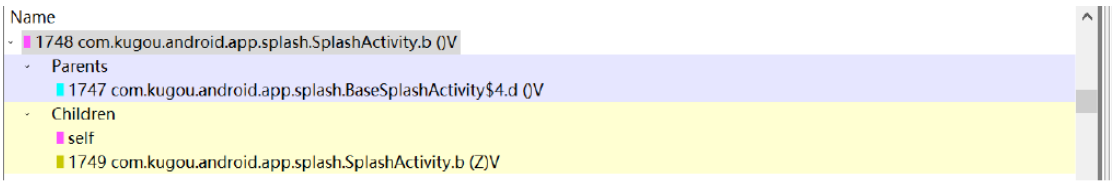
```
public void a(Activity activity, boolean z2, final c.a aVar) {
    if (com.kugou.common.q.b.a().an(com.kugou.common.environment.a.bM())) {
        this.f43179c.setCommissionVolumeOn(false);
    } else {
        this.f43179c.setCommissionVolumeOn(true);
    }
    a().e();
    a().m();
    if (z2) {
        a().n();
        this.q = e.d(activity);
        this.q.setOnClickListener(new View.OnClickListener() {
            /* class com.kugou.android.app.boot.SplashCard.AnonymousClass1 */

            public void onClick(View view) {
                try {
                    com.kugou.common.datacollect.a.a().a(view);
                } catch (Throwable unused) {
                }
                a(view);
            }

            public void a(View view) {
                c commissioVolumePack = SplashCard.this.f43179c.getCommissioVolumePack();
                if (commissioVolumePack.h()) {
                    commissioVolumePack.a(false);
                    c.a aVar = aVar;
                    if (aVar != null) {
                        aVar.b();
                    }
                } else {
                    commissioVolumePack.a(true);
                    c.a aVar2 = aVar;
                    if (aVar2 != null) {
                        aVar2.a();
                    }
                }
                SplashCard.this.f43179c.j();
            }
        });
```

高度怀疑是这个函数，现在尝试用 xposed 调用：

findandhook 可以直接找到匿名内部类，具体的类名还是通过 method profiling 发现，在源码中核对。

调用 onClick 函数，并不行，view 的值有办法调用出来，但处理不了匿名内部类的调用

尝试找其调用的方法，这些方法也大多是匿名内部类的方法，找到后面第 4 个：



- **How you hooked the function**

1. hook 一个每次启动 app 都会调用的方法

com.kugou.android.app.splash.SplashActivity.r

进行 afterHookedMethod

2.根据之前找到的方法执行在 hook 的方法后借助其实例的类，call b 方法，跳过广告。

本次实验同时用 frida 和 xposed 实现了

- **Task2 微信发送延时消息**
- **Introduction**

微信 apk 来源，mumu 模拟器自带应用商店。

第一次安装使用手机自带应用商店提供的版本，结果才登陆就导致微信被封号，所以只好使用 mumu 提供的版本。

Task 的目标，使得微信支持带有计时器的消息发送

延时消息发送格式：@Timer:XXs/min:具体消息

例如：@Timer:5s:测试

表示延时 5s 发送“测试”这一条信息

- **How you find the target function**
  Introduce how you reversed the target apk and located the target function you want to hook

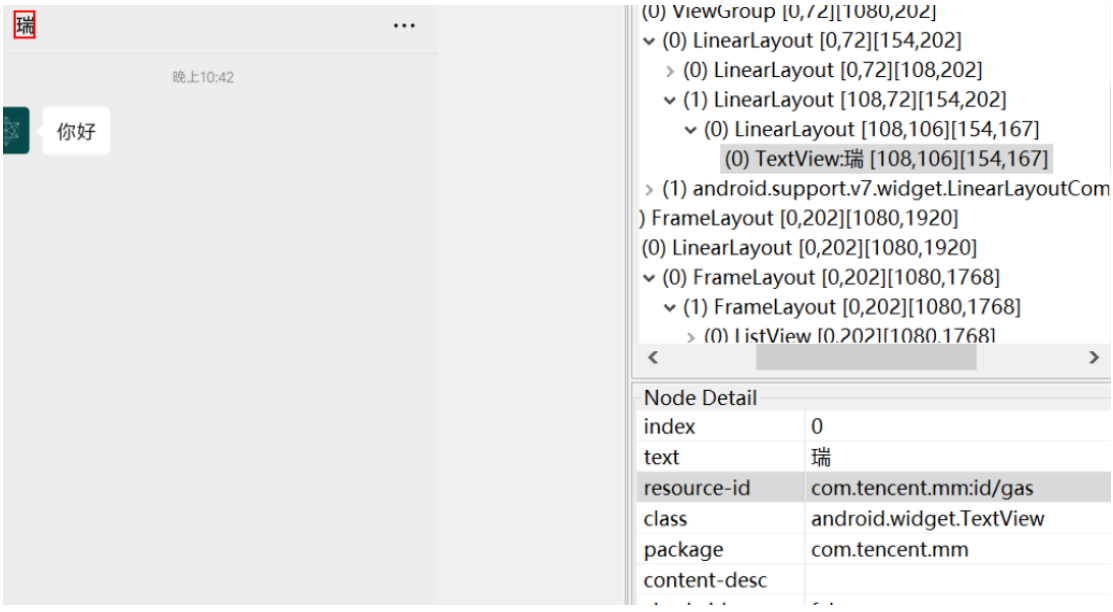
  1.找按钮，研究发消息的逻辑

  聊天框中发送按钮

  com.tencent.mm:id/anv

聊天对象：

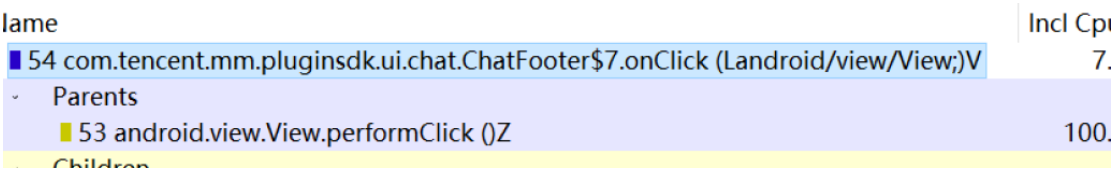
com.tencent.mm:id/gas



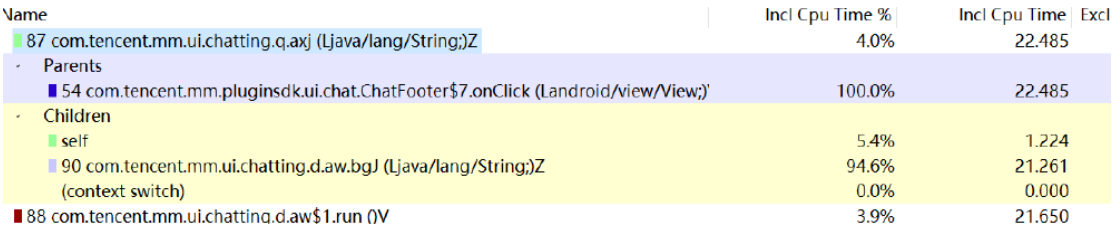
……

使用 method profiling：

在发消息的整个过程中只有一次 onclick 方法：



子方法中占用资源最多的是：



尝试 hook 看一下结果

直接成功了就是我发的值

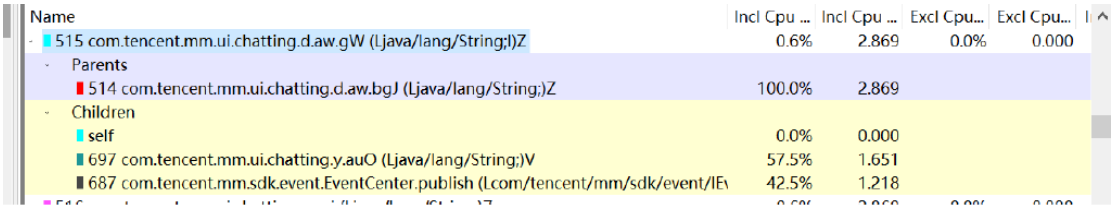
- **How you hooked the function**
  Introduce how you hook the target functions to realize your goal

1.延时的实现

要实现延时回复，最简单的方法就是，让这个线程休眠。

直接在这个方法上休眠，微信直接闪退。

随后找子方法，把所有的子方法依次试一下。



com.tencent.mm.ui.chatting.d.aw.gW

在这个方法上进行 sleep，微信不会闪退，可以说实现的延时发送。

2.优化方法

直接将在方法内 sleep 会直接导致输入框阻塞，无法进一步输入，采取创建一个新线程，到时间回调该方法的方式。

加上中间对具体延时标志的判断，再次尝试，就成功了。

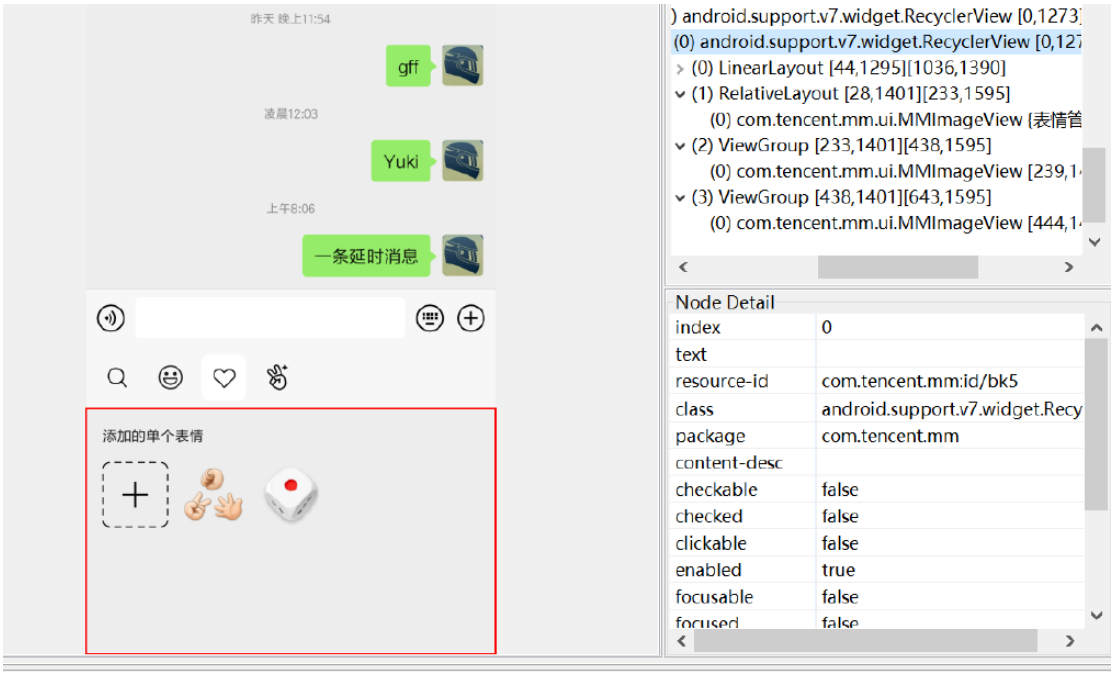
- **Task3 微信控制筛子点数**
- **Introduction**

微信默认表情中有一个骰子，每次发出表情会随机出现一个点数，这个 task 就是要控制出现的点数，在这里控制点数为 6.

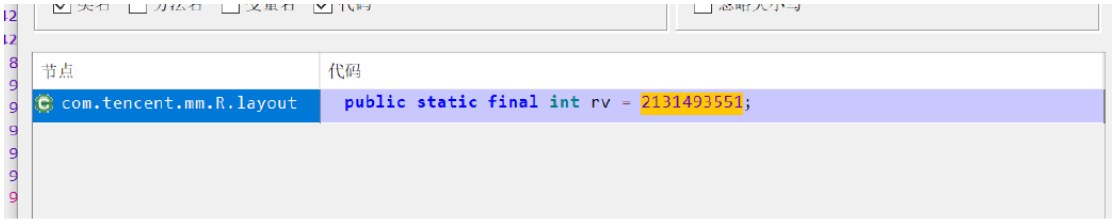
- **How you find the target function**

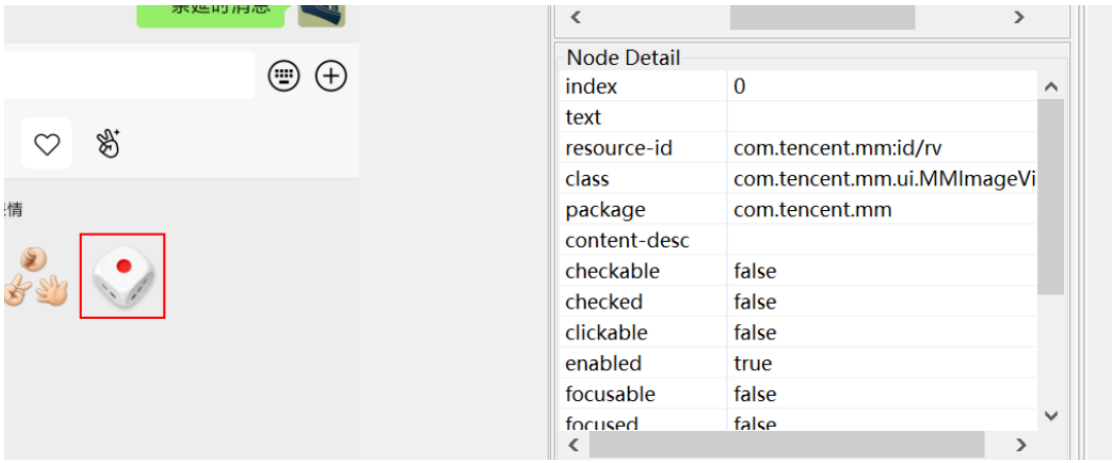
Introduce how you reversed the target apk and located the target function you want to hook

两种方法同时尝试：

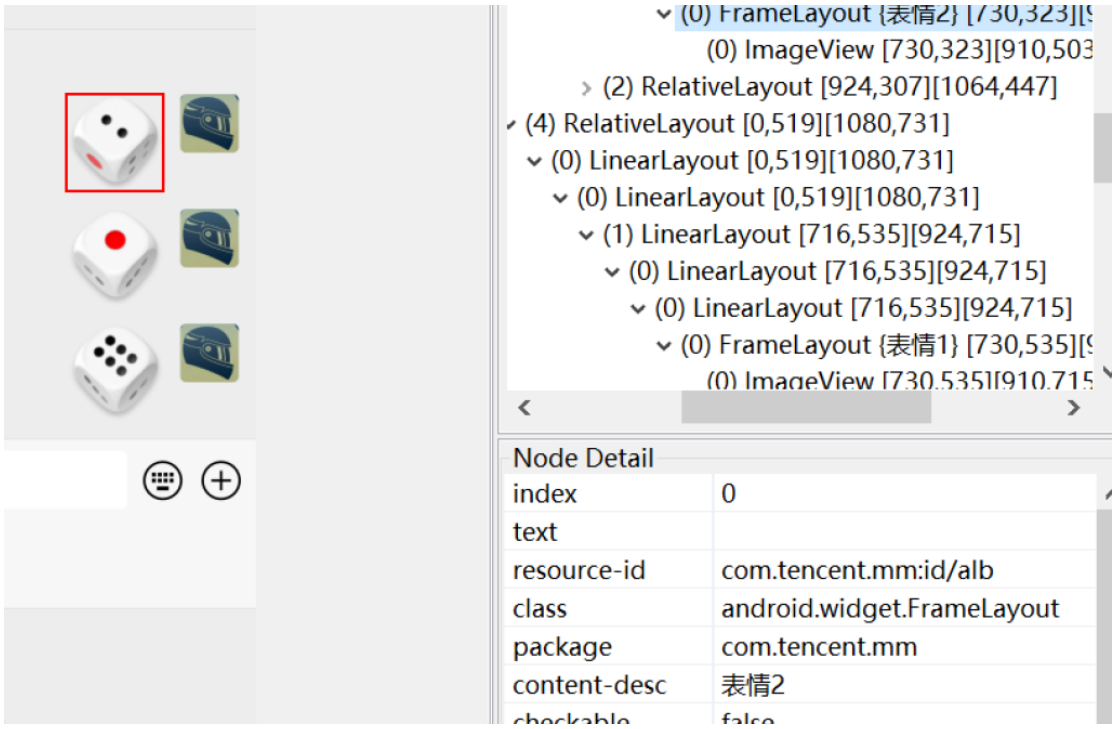


com.tencent.mm:id/rv
<public type="layout" name="rv" id="0x7f0c02af" />
2131493551

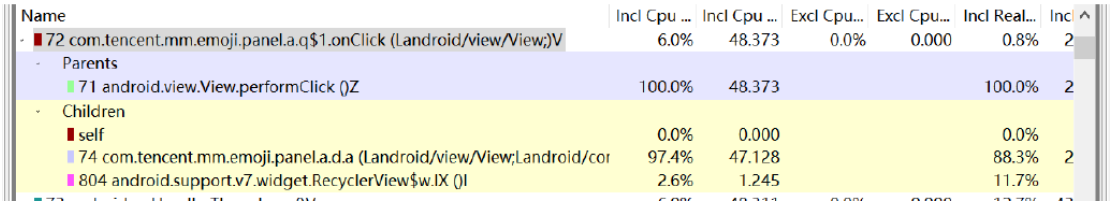




不同表情出来的 content-desc 不同

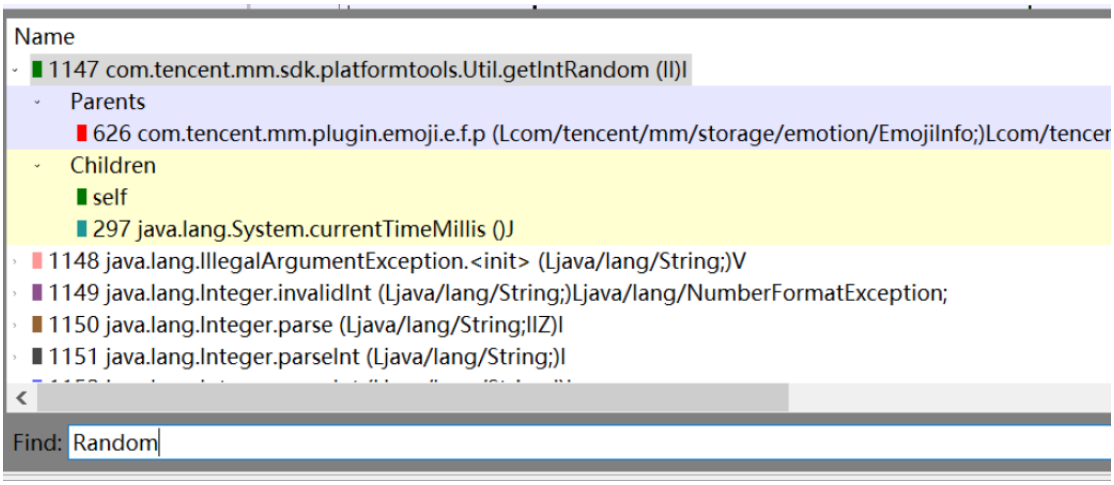


点下表情按钮：



因为点数是随机，大胆推测会调用随机函数，直接搜索 random

竟然有且只有一个：




```
public static int getIntRandom(int i2, int i3) {
    AppMethodBeat.i(157862);
    Assert.assertTrue(i2 > i3);
    int nextInt = new Random(System.currentTimeMillis()).nextInt((i2 - i3) + 1) + i3;
    AppMethodBeat.o(157862);
    return nextInt;
}
```

```
@Override // com.tencent.mm.pluginsdk.a.e
public final EmojiInfo p(EmojiInfo emojiInfo) {
    AppMethodBeat.i(108442);
    if (emojiInfo.field_catalog == EmojiGroupInfo.Sbm && emojiInfo.getContent().length() > 0 && EmojiInfo.arw(Util.getInt(emojiInfo.getConter
        Cursor afl = l.getEmojiStorageMgr().LAF.afl(Util.getInt(emojiInfo.getContent(), 0));
        if (afl != null && afl.getCount() > 1) {
            int intRandom = Util.getIntRandom(afl.getCount() - 1, 0);
            emojiInfo = new EmojiInfo();
            afl.moveToPosition(intRandom);
            emojiInfo.convertFrom(afl);
        }
        if (afl != null) {
            afl.close();
        }
    }
    AppMethodBeat.o(108442);
    return emojiInfo;
}
```

hook 一下，发骰子时这个函数的第一个参数永远是 5，第二个永远是 0。

- **How you hooked the function**

Introduce how you hook the target functions to realize your goal

基本确定它就决定这个骰子点数的关键随机函数。将这个函数的返回值修改为 5，

则每次发送都为 6 点，此时预期的目标已经实现。

但是，在这时，石头剪刀布的表情发不出来了，猜测它也是调用这个随机函数。

尝试调用父方法的方式来解决，但是始终没有成功，于是回到最简单的方法，先判断一下调用这个函数的传入参数，第一个参数是 5，第二个是 0，再改返回值，修改后，石头剪刀布正常了。

- **Task4 微信自动回复**
- **Introduction**

Introduce the task briefly

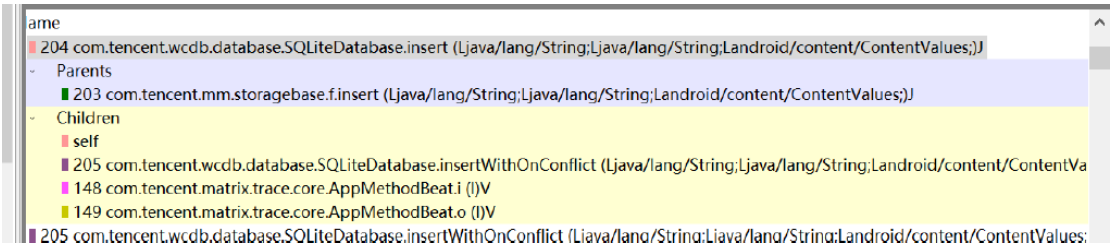
实现微信的自动回复，当对方发送消息的时候，能根据对方的消息自动进行回复。

- **How you find the target function**

Introduce how you reversed the target apk and located the target function you want to hook

对接收消息的逻辑分析

根据网的大佬帖子，微信收发消息使用的是数据库，直接在息屏状态下给手机发消息，用 monitor 记录期间调用的方法，查找数据库插入有关的方法，找到了下面这个。



尝试 hook 每次发消息，里面具体的信息。

```
 XposedHelpers.findAndHookMethod("com.tencent.wcdb.database.SQLiteD
atabase",
                    lpparam.classLoader, "insert", String.class,
String.class, "android.content.ContentValues",
                    new XC_MethodHook() {
                        @Override
                        protected void
beforeHookedMethod(MethodHookParam param) throws Throwable {
                            String old1 = (String) param.args[0];
                            String old2 = (String) param.args[1];
                            ContentValues content = (ContentValues)
param.args[2];



                            XposedBridge.log("Hook的string1:
"+old1);
                            XposedBridge.log("Hook的string2:
"+old2);
                            XposedBridge.log("Hook的
content: "+content.toString());
                        }
                    });
```

Log:（发送的信息为：洗个澡）

```
05-22 09:24:55.734 I/Xposed  ( 6231): Hook的string1: message
05-22 09:24:55.734 I/Xposed  ( 6231): Hook的string2: msgId
05-22 09:24:55.735 I/Xposed  ( 6231): Hook的content: bizClientMsgId=
msgId=277 msgSvrId=6327337558414731998 talker=wxid_2r139dxjtzho22
content=洗个澡 flag=0 status=3 msgSeq=711770794
createTime=1621646690000 lvbuffer=[B@2efb9de isSend=0 type=1
bizChatId=-1 talkerId=27
```

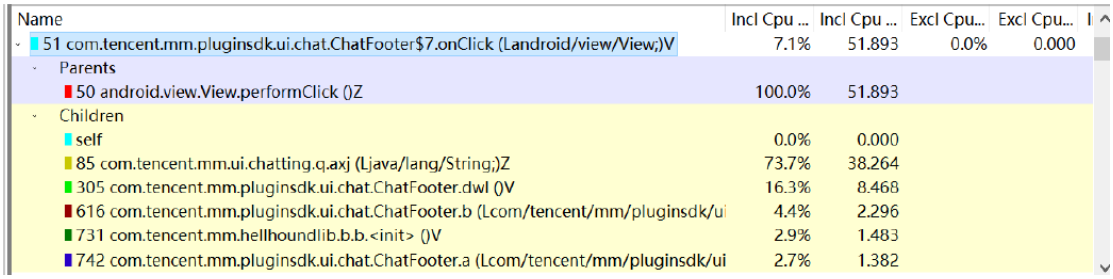
- **How you hooked the function**

Introduce how you hook the target functions to realize your goal

直接根据延时发送消息的经验，尝试实现主动发送消息。

发现一个问题，调用方法的时候很难判断，是在给谁发消息，所以需要同一个类中去找到实例化的对象

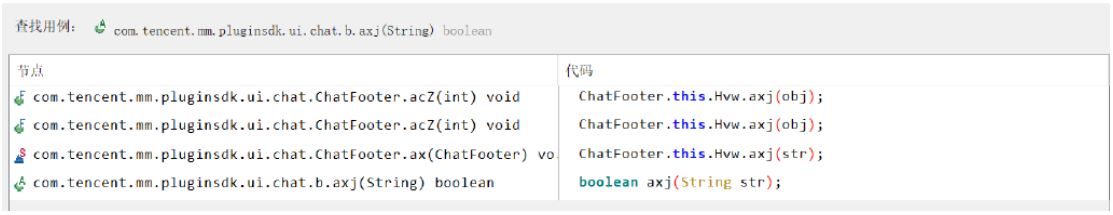
顺着之前的进入找 parents 方法，找到这个：



Name	Incl Cpu ...	Incl Cpu ...	Excl Cpu...	Excl Cpu...
51 com.tencent.mm.pluginsdk.ui.chat.ChatFooter$7.onClick (Landroid/view/View;)V	7.1%	51.893	0.0%	0.000
Parents				
50 android.view.View.performClick ()Z	100.0%	51.893		
Children				
self	0.0%	0.000		
85 com.tencent.mm.ui.chatting.q.axj (Ljava/lang/String;)Z	73.7%	38.264		
305 com.tencent.mm.pluginsdk.ui.chat.ChatFooter.dwl ()V	16.3%	8.468		
616 com.tencent.mm.pluginsdk.ui.chat.ChatFooter.b (Lcom/tencent/mm/pluginsdk/ui	4.4%	2.296		
731 com.tencent.mm.hellhoundlib.b.b.<init> ()V	2.9%	1.483		
742 com.tencent.mm.pluginsdk.ui.chat.ChatFooter.a (Lcom/tencent/mm/pluginsdk/ui	2.7%	1.382		

```
    public final void onClick(View view) {
        AppMethodBeat.i(206603);
        com.tencent.mm.hellhoundlib.b.b bVar = new com.tencent.mm.hellhoundlib.b.b();
        bVar.aQ(view);
        com.tencent.mm.hellhoundlib.a.a.b("com/tencent/mm/pluginsdk/ui/chat/ChatFooter$46", "android/view/Vie
        if (ChatFooter.this.HxE.getVisibility() == 0 && 8 == ChatFooter.this.HxD.getVisibility()) {
            String obj = ChatFooter.this.Hxf.getText().toString();
            if (ChatFooter.this.Hvw != null && !Util.isNullOrNil(obj)) {
                ChatFooter.this.HxN = false;
                ChatFooter.this.HwT.DNp = ChatFooter.this.DNb;
                ChatFooter.this.HwT.DNt = obj.length();
                if (1 == ChatFooter.this.HwT.DNw) {
                    ChatFooter.this.HwT.setExitType(8);
                } else {
                    ChatFooter.this.HwT.setExitType(5);
                }
                ChatFooter.this.Hvw.axj(obj);
                ChatFooter.this.Hvw.dGx();
                ChatFooter.d(ChatFooter.this, obj);
            }
        }
        com.tencent.mm.hellhoundlib.a.a.a(this, "com/tencent/mm/pluginsdk/ui/chat/ChatFooter$46", "android/vi
        AppMethodBeat.o(206603);
    }
});
```

在这个方式中查找消息字符串的使用情况，查看 ChatFooter.this.Hva.axj 的用例：



查找用例： com.tencent.mm.pluginsdk.ui.chat.b.axj(String) boolean

节点	代码
com.tencent.mm.pluginsdk.ui.chat.ChatFooter.acZ(int) void	ChatFooter.this.Hvw.axj(obj);
com.tencent.mm.pluginsdk.ui.chat.ChatFooter.acZ(int) void	ChatFooter.this.Hvw.axj(obj);
com.tencent.mm.pluginsdk.ui.chat.ChatFooter.ax(ChatFooter) vo	ChatFooter.this.Hvw.axj(str);
com.tencent.mm.pluginsdk.ui.chat.b.axj(String) boolean	boolean axj(String str);

最下面的用例是将这个方法实现成了接口。

```
package com.tencent.mm.pluginsdk.ui.chat;

import android.view.MotionEvent;

public interface b {
    long Oe();

    void T(MotionEvent motionEvent);

    void auO(String str);

    boolean axj(String str);
```

再找这个接口被调用的情况：

找到这个类 com.tencent.mm.pluginsdk.ui.chat.ChatFooter 的 etFooterEventListener，利用这个方法将实例化接口中的 axj 方法 setFooterEventListener 会把 private b Hvw 赋值，到这一步所有的条件都集齐，收得到对方的信息，也发得出信息。

实现思路：

1.hook com.tencent.mm.pluginsdk.ui.chat.ChatFooter 的 setFooterEventListener 方法，每次它被调用后，就去找到 Hvw 这个方法对象，准备给发消息用。

2.hook com.tencent.wcdb.database.SQLiteDatabase 的 insert 方法，收到信息后，根据收到的信息利用 Hvw 将回复发出。

- Task5
- Introduction

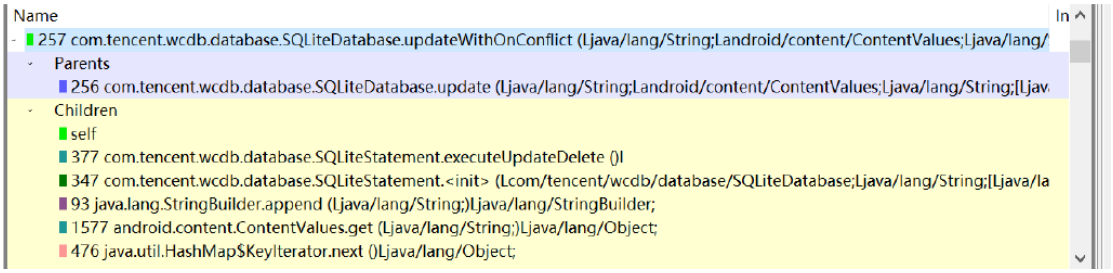
Introduce the task briefly

用微信发送消息，在一段时间内，发送方可以选择撤回，让接收方的手机上不再保存该消息，本 task 就是实现阻止微信消息的撤回

- How you find the target function

Introduce how you reversed the target apk and located the target function you want to hook

根据之前的经验，直接从 method profiling 入手，撤回一条信息，看看期间微信调用了什么方法，以数据库、删除、更新为关键词找到下面这个：



hook 一下看内容，正是我要撤回的信息

- **How you hooked the function**

Introduce how you hook the target functions to realize your goal

要实现这个效果，直接采取最简单，最暴力的方法，在 updateWithOnConflict 方法调用前直接把传入参数都改为空，自然阻止了撤回。