XPOSED(VXP)/FRIDA Report of PoRE

Student ID　20307130350
Name　陈丹纯

Ps：我文件中上传了两个 xposed 对应的 java 代码，但为了方便助教老师核对，将两个 task 写在同一个模块中并生成了一个 apk 文件，可以用该 apk 文件同时验证两个 task。

- **Tasks List**

Write down the tasks list you finished and the corresponding score
1.微信延时发送消息
2.微信自动回复消息
3.微信零钱余额显示修改

- **Project Demo Video**
  Baidu Netdisk share link (and verify code)

  链接：https://pan.baidu.com/s/1cRv7vZwZC0h553oWCBQOhw
  提取码：48o3

- **Task1**
- **Introduction**
  Introduce the task briefly

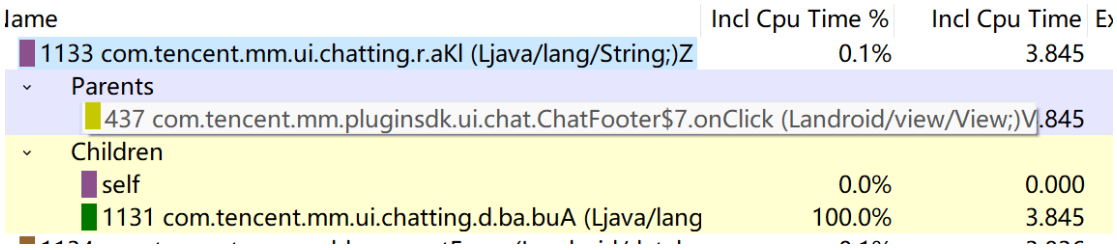
  微信延时发送消息，当输入消息满足：@Timer:XXs/min:examplemessage 的时候可以延迟发送 XXs/min。

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

1. **用 DDMS 的 Method Profiling 功能分析**

   a) **在模拟器中打卡微信聊天窗口，打开 Monitor，选中微信进程，start Method Profiling，然后在聊天窗口中发送一条消息，stop method profili，分析生成的分析文件：**

b) 搜索 onClick，发现了 chatFooter 类，观察 onClick()，发现其调用了一个 boolean chatting.r.aKl(String)方法，



2. 验证：

a) 从参数类型是和返回类型推测，应该是传入了消息文本，返回值判断是否发送成功。

用 AS hook 该函数验证，然后查看日志。发现确实调用了。

b) 进一步确定确实是发送消息的方法，拦截该方法，override beforeHookedMethod，在调用该方法前修改其参数 param.arg[0] = “1111“，无论发送什么消息，得到的都是”1111“，因此证实了猜想。

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

使用 Xposed 框架进行 hook。

1. Hook method：Hook com.tencent.mm.ui.chatting.r.aKl(String),

   重写 beforeHookedMethod

2. Modify args：根据输入文本的格式要求：@Timer:XXs/min:<message>,

   对第一步得到的参数，利用（(String)param.arg[0]）.split(“:”)将输入的文本消息分成 String[] str，然后对 str[0]和 str[1]分析，如果满足响应的格式要求，就 sleep 对应的时间，然后将最后的发送消息修改为 str[2]。

3. **改进：在测试的过程中发现虽然能延迟发送，但是在延迟发送期间，聊天框被阻塞了，哪怕对方再次发送消息，也要等到我的延迟消息发送出去之后才能收到。因此在 beforeHookedMethod 里新开一个线程，在这个线程里进行 sleep 延迟的时间并发送修改后的消息，同时将原本的线程的参数改成 null，这样就不会阻塞了。**

- **Task2**
- **Introduction**

Introduce the task briefly

微信自动回复消息，当接收到消息时可以自动回复相应的内容。

- **How you find the target function**

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

1. **在 task1 的基础之上，明确了发送消息的 api 是 chatting.r.aKl(String);但是现在还需要知道什么时候接收到消息，才能决定什么时候自动发送。**
2. **在 jadx 中查看 aKl 函数**

```
598     public final boolean aKl(String str) {
599         AppMethodBeat.i(34525);
599         hvK();
601         this.QSE.hwe();
601         boolean buA = ((aq) this.QSE.aY(aq.class)).buA(str);
601         AppMethodBeat.o(34525);
601         return buA;
        }
```

不断寻找分析其调用的方法，如 AppMethodBeat.i 和 aY 等方法，调用太多，有点混乱。

3. 考虑到当接收到新消息时，微信会往本地数据库插入聊天消息，将聊天消息保存到本地。因此只要 hook 住消息的插入动作，就能实时的获取到聊天消息。微信的数据库最终都调用了 com.tencent.wcdb.datatbase.SQLiteDatabase 类，插入方法是 insert，里面又调用了 insertWithOnConflict 方法，用 AS 尝试 hook 这两个方法，都 hook 到了。因此最终选择第二个方法作为 api。

4. 打印 insertWithOnConflict 方法的参数列表，查看日志，发现参数 str 是消息类型，contentValues 是消息的相关信息，内涵消息内容，发送者等。

```
public final long insertWithOnConflict(String str, String str2, ContentValues contentValues, int i) {
    SQLiteStatement sQLiteStatement;
    int i2 = 0;
    AppMethodBeat.m45755i(3184);
    acquireReference();
    try {
        StringBuilder sb = new StringBuilder();
        sb.append("INSERT");
        sb.append(CONFLICT_VALUES[i]);
        sb.append(" INTO ");
        sb.append(str);
        sb.append('(');
        Object[] objArr = null;
        int size = (contentValues == null || contentValues.size() <= 0) ? 0 : contentValues.size();
```

```
9): hook数据库：tablemessage
9): ;nullColumnHack:msgId
9): ;+CONFLICT_VALUES:;contentValues:msgId=346 status=1 createTime=165
9): hook数据库：tablemessage
9): ;nullColumnHack:msgId
9): ;+CONFLICT_VALUES:;contentValues:msgId=347 msgSvrId=555134320438
9):  flag=0 status=3 msgSeq=777104823 createTime=1650027729000 imgPa
```

```
//1：表示是自己发送的消息
int isSend = contentValues.getAsInteger( key: "isSend");
//消息内容
String strContent = contentValues.getAsString( key: "content");
//说话人ID
String strTalker = contentValues.getAsString( key: "talker");
XposedBridge.log( text: "isend="+isSend+"\tcontent="+strContent+"\ttalker="+st
```

5. 因此可通过该方法的参数直接得到发送者的 id，进而作为发送消息的接收者的 id。但是 api1 找到发送函数的传入参数只有 String str 发送的消息，不能传入接收者的 id，因此需要继续找接收者 id 的相关方法。

6. 通过打印栈和不断搜索，找到了 modelmulti.i 类，其构造器的参数列表

```
public C25689i(String str, String str2, int i, int i2, Object obj) {
    AppMethodBeat.m45755i(43024);
    if (Log.getLogLevel() <= 1) {
        Log.m50591d("MicroMsg.NetSceneSendMsg", "dktext :%s", Util.getStac
    }
    if (!Util.isNullOrNil(str)) {
        C32127ca caVar = new C32127ca();
        caVar.setStatus(1);
        caVar.mo25211GY(str);
        caVar.setCreateTime(C25593bp.m28583Pp(str));
        caVar.mo25248pU(1);
        caVar.setContent(str2);
        caVar.setType(i);
        String a = m28796a(((C27620w) C36539h.m46902U(C27620w.class)).mo25538ad(caVar), obj, i2);
        if (!Util.isNullOrNil(a)) {
            caVar.mo25213Gb(a);
            Log.m50597i("MicroMsg.NetSceneSendMsg", "NetSceneSendMsg:MsgSource:%s", caVar.fEf);
        }
        if ((i2 & 4) != 0 || (i2 & 8) != 0) {
            int i3 = (i2 & 4) != 0 ? 2 : 3;
            Log.m50597i("MicroMsg.NetSceneSendMsg", "has paste fully flag, %d", Integer.valueOf(i3));
            HashMap hashMap = new HashMap();
```

正在讲话: 包炜杰;

根据其代码，推测参数 str 为接收者 id，str2 为发送内容。Hook 该方法验证。确实调用了，是 api3.

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

使用 Xposed 框架进行 hook。

1. hook 数据库 insertWithOnConflict 方法，得到其发送者等相关消息。strTalker 即发送者 id 即自动回复的接收者 id。
2. 新建一个 modelmulti.i 类，传入第一步得到的接收者 id 作为第一个参数。这样创建的对象就可以识别接收者的 id。
3. 但是，虽然新建了一个类，但是试了很久没找到对应的调用其的方法，因此该自动回复消息发送有延迟。最后选择调用发送 task1 提到的发送方法来发送消息，尽而“自动调用”前面新建的 modelmulti.i 对象。
4. 因为 aKl 方法不是静态方法，在 chatFooter 中发现了一个字段 chat.b MgP，chat.b 是一个接口，chatting.r 正是对其的实现，在后者的 onClick 中调

用了 b.aKl(), 是回调。找到 MgP 被赋值的地方是 chatFooter.setFooterListener(), 这个方法可以实例化 MgP。可以 hook 它来获得 MgP 的引用尽而调用 aKl。

5. 为了避免自动回复自己发的消息陷入循环，同时防止在公众号和微信群中自动回复，根据 insertWithOnConflict 的参数 contentValues 得到的消息可以判断当 isSend=1 是表示自己发送，如果 strTalker 的结尾是 @chatroom，表示是群消息，如果其开头是 gh_ 表示公众号。因此加判断，以上条件不自动回复。

- **Task3**
- **Introduction**

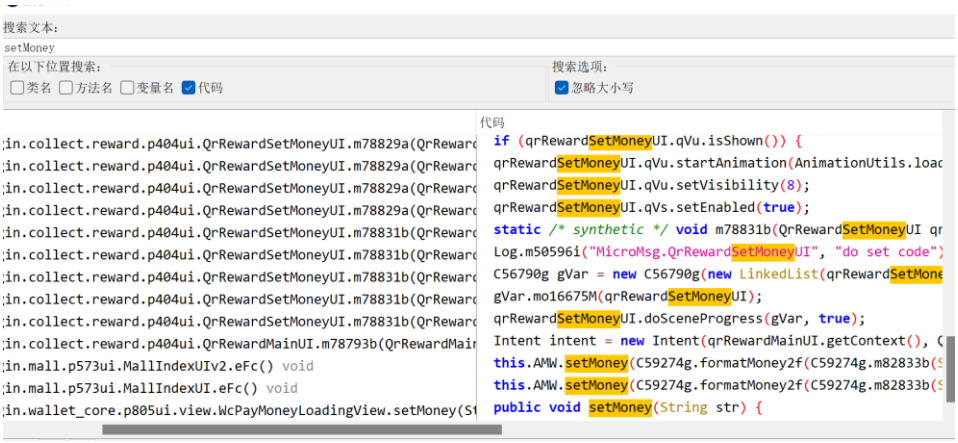
Introduce the task briefly

改变微信零钱余额显示。

- **How you find the target function**

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

1. 用 jadx 打开微信 apk，搜索 setMoney；发现了一个关于 wallet 的 WcPayMoneyLoadingView 类。根据名字推测应该与微信零钱余额显示有关。



2. 观察发现，该类中有三个跟 setMoney 相关的函数，分别是 setMoney，

setFirstMoney 和 setNewMoney，传入参数都是 String，推测是有关零钱变动的相关函数。确定了该三个方法作为 hook 对象

How you hooked the function

Introduce how you hook the target functions to realize your goal

1．直接用 frida hook 这三个函数，改变传入的参数为相同的目标值。即可得到想要的效果。