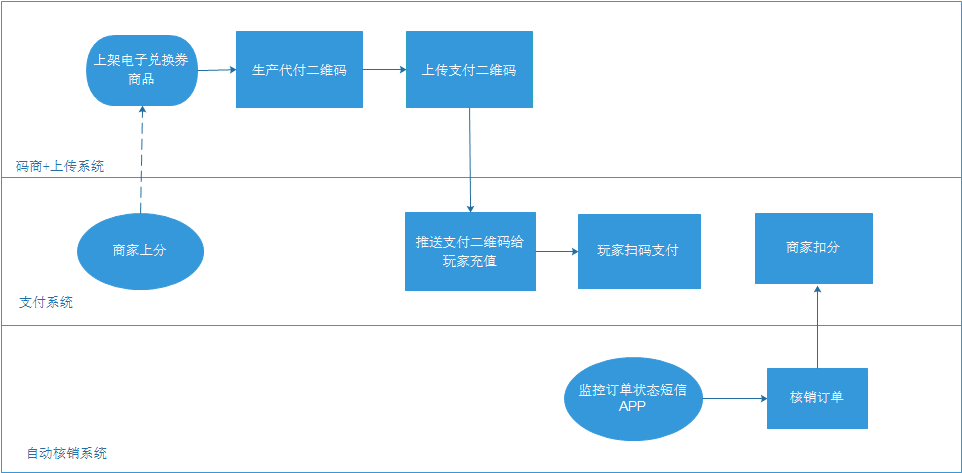
## 竞品淘宝代付原理和实现调研

## 1综述

### 1.1 淘宝代付流程图



### 1.2 核心功能

1. 上传支付二维码时自动识别金额

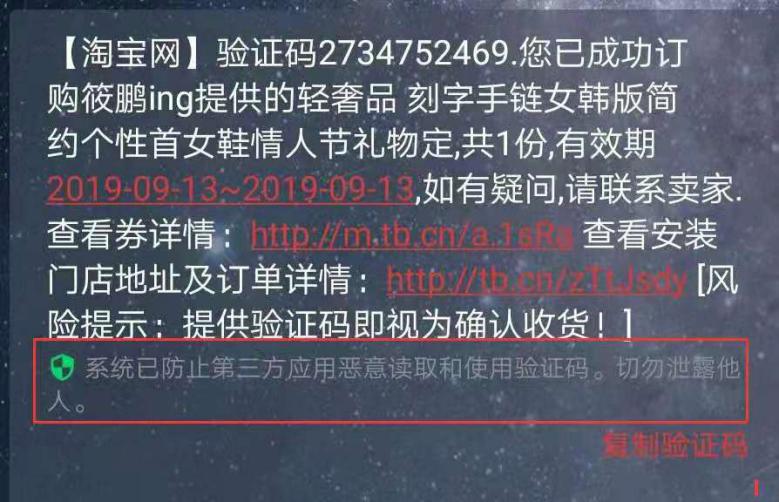
Android APP版：采用Xposed框架拦截淘宝APP 接口。

PC版：通过记录淘宝账号登录cookies模拟接口请求，获取二维码和金额。

1. 核销订单自动实现

Android APP版主要功能分成两部分，核心是为了将短信验证码和订单号对应起来，实现核销和商户扣分。

1. 监控手机短信，上报短信内容到管理后台。



1. 采用Xposed框架拦截阿里旺旺卖家版千牛APP 接口（注，此部分非必须）。

## 2 APP功能实现

### 2.1 上传支付二维码时自动识别金额

##### 1）APP功能部分

* 拦截用户购买记录

XposedHelpers.findAndHookMethod("com.taobao.tao.purchase.activity.PurchaseActivity", "onCreate", new Object[] { Bundle.class, new XC\_MethodHook()

* 生成好友代付码

XposedHelpers.findAndHookMethod("com.taobao.tao.purchase.activity.PurchaseActivity", "initView", new Object[] { new XC\_MethodHook()

{

protected void afterHookedMethod(final XC\_MethodHook.MethodHookParam paramAnonymousMethodHookParam)

throws Throwable

{

super.afterHookedMethod(paramAnonymousMethodHookParam);

Object localObject = (Activity)paramAnonymousMethodHookParam.thisObject;

paramAnonymousMethodHookParam = (ListView)TaobaoHook.this.findViewByResName((Activity)localObject, "com.taobao.taobao:id/lv\_content", new String[0]);

localObject = (ViewGroup)((Activity)localObject).findViewById(16908290);

PayHelperUtils.findTextViewByText((ViewGroup)localObject, "购买数量", 10).flatMap(new Function()

{

public ObservableSource<TextView> apply(TextView paramAnonymous2TextView)

throws Exception

{

paramAnonymousMethodHookParam.setSelection(paramAnonymousMethodHookParam.getAdapter().getCount() - 1);

return PayHelperUtils.findTextViewByText(this.val$content, "朋友代付", 10);

}

}).subscribe(new Consumer()new Consumer

{

public void accept(TextView paramAnonymous2TextView)

throws Exception

{

TaobaoHook.access$002(TaobaoHook.this, true);

((ViewGroup)paramAnonymous2TextView.getParent()).performClick();

}

}, new Consumer()

{

public void accept(Throwable paramAnonymous2Throwable)

throws Exception

{

LogUtil.log(paramAnonymous2Throwable.getMessage());

}

});

}

} });

* 获取代付详情

XposedHelpers.findAndHookMethod(XposedHelpers.findClass("com.taobao.order.detail.ui.OrderCoreDetailActivity", paramClassLoader), "onCreate", new Object[] { Bundle.class, new XC\_MethodHook()

##### 后台接口

/channel/common/plugin/app/alipaytbs/class/uppayurl?tid=&id=&loginId=&codeurl=&\_bizNo=&money=

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| sign | 淘宝返回的codeurl的sign，接口外层又做了一次md5签名。 |
| tid |  |
| id |  |
| loginId |  |
| codeurl |  |
| \_bizNo | 主键，订单号 |
| money |  |

### 2.2 核销订单部分实现

##### 2.3.1 短信监控后台接口

/channel/common/plugin/app/alipaytbs/class/uppayurl?mobile=&content=&time=

|  |  |
| --- | --- |
| **请求参数名称** | **说明** |
| mobile | 发送人的手机号 |
| content | 短信内容 |
| time | 发送时间 |

#### 2.3.1 千牛监控订单（非必须）

##### 1）APP功能部分

// 获取买家已付款订单

XposedHelpers.findAndHookMethod("c8.Qfi", paramClassLoader, "a", new Object[] { XposedHelpers.findClass("c8.Sfi", paramClassLoader), new XC\_MethodHook()

{

protected void afterHookedMethod(XC\_MethodHook.MethodHookParam paramAnonymousMethodHookParam)

throws Throwable

{

LogUtil.log("after");

super.afterHookedMethod(paramAnonymousMethodHookParam);

paramAnonymousMethodHookParam = JSON.parseObject(JSON.toJSONString(((List)XposedHelpers.getObjectField(paramAnonymousMethodHookParam.getResult(), "c")).get(0)));

LogUtil.log(paramAnonymousMethodHookParam.toString());

try

{

if (paramAnonymousMethodHookParam.getString("subMsgTypeName") != null)

{

if (!"买家已付款".equals(paramAnonymousMethodHookParam.getString("subMsgTypeName"))) {

return;

}

String str = SPUtils.getInstance(QianNiuHook.userId).getString("recentBizId", "-");

Object localObject = paramAnonymousMethodHookParam.getString("bizId");

if (str.equals(localObject)) {

return;

}

SPUtils.getInstance(QianNiuHook.userId).put("recentBizId", (String)localObject);

str = paramAnonymousMethodHookParam.getString("msgContent");

localObject = new Intent();

((Intent)localObject).setAction(ActConst.TAOBAO\_GETORDER);

((Intent)localObject).putExtra("bizId", paramAnonymousMethodHookParam.getString("bizId"));

((Intent)localObject).putExtra("payer", StringUtils.getTextCenter(str, "买家ID:", "\","));

((Intent)localObject).putExtra("money", StringUtils.getTextCenter(str, "金额: ", "元"));

paramContext.sendBroadcast((Intent)localObject);

LogUtil.log("sendOrder");

return;

}

return;

}

catch (Exception paramAnonymousMethodHookParam)

{

LogUtil.log(paramAnonymousMethodHookParam.getMessage());

}

}

##### 后台接口

/channel/common/plugin/app/alipaytbs/class/upnotify?bizId=&payer=&money=

|  |  |
| --- | --- |
| **请求参数名称** | **说明** |
| bizId | 订单号 |
| payer | 买家ID |
| money | 金额 |

### 2.3 其他功能

* 淘宝订单自动确认收货
* 云闪付、支付宝、拼多多、农信支付收款回调

## 3 淘宝核销工具V1.1.190913.exe的实现

### 3.1 功能实现

#### 3.1.1 调用支付宝接口获取二维码金额

##### 1）APP功能部分

长传支付二维码前，需要登录支付宝账号，获取支付宝账号cookie，

以下面二维码为例：



该二维码解码后的内容为：

<https://qr.alipay.com/_d?_b=peerpay&enableWK=YES&biz_no=2019091304200325251044028642_b27d02f293fdc6bea61e697897fdd22c&app_name=tb&sc=qr_code&v=20190920&sign=2ffde5&__webview_options__=pd%3dNO>

请求：

<https://mclient.alipay.com/h5/peerpay.htm?biz_no=2019091304200325251044028642_b27d02f293fdc6bea61e697897fdd22c&app_name=tb&sc=qr_code&v=20190920&sign=2ffde5&__webview_options__=pd%253dNO>

可获得包含金额的h5页面，通过解析h5获取金额。

##### 后台接口

/channel/jpay/tbuppayurl?team=&no=&payurl=&code=1&money=

|  |  |
| --- | --- |
| **参数名称** | **说明** |
| team | 团队名称 |
| no | 订单号，根据二维码图片的名称获取？ |
| payurl | 支付码连接 |
| code | 固定为1 |
| money | 金额 |

#### 3.1.2 调用淘宝接口核销订单

##### 1）APP功能部分

淘宝核销工具登录淘宝账户后，记录账户的cookie，接口公共参数如下：

**Headers：**

POST /consume/code.htm HTTP/1.1

Host: ma.taobao.com

Connection: keep-alive

Content-Length: 131

Cache-Control: max-age=0

Origin: http://ma.taobao.com

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36

Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

Referer: http://ma.taobao.com/consume/code.htm

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7

Cookie:

并调用以下接口：

1）校验验证码

**参数：**

\_tb\_token\_: 7bb63eb6b7e73

action: consume/code\_action

event\_submit\_do\_validate\_code: 提交

\_fm.cod.\_0.co: **核销码**

\_fm.cod.\_0.m:

**汇编源码：**



1. 确认消费

**参数：**

action: consume/confirm\_action

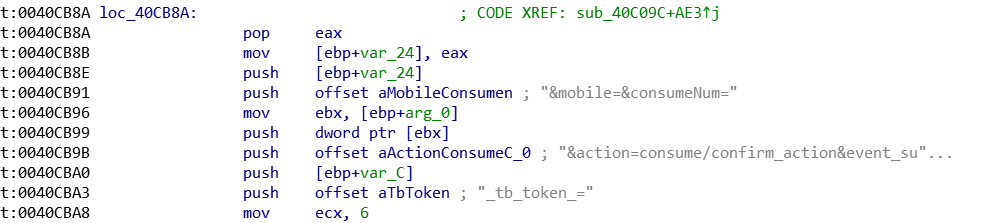
event\_submit\_do\_validate\_code: 提交

code:核销码

mobile:不用填

consumeNum: 1

**汇编源码：**



##### 后台接口

* 获取核销的验证码列表

/channel/jpay/tbpayurl?merNo=001&sellAccount=&sign=

merNo：商家序号，固定001

sellAccount：店铺名称

|  |  |
| --- | --- |
| **返回参数名称** | **说明** |
| respCode | 返回验证码 |
| respDesc | 描述 |
| totalCount | 总数，0000为正确 |
| data |  |
| verifyCode | 验证码 |

* 核销成功回调后台

/channel/jpay/tbnotify?sellAccount=&verifyCode=&respCode=&respDesc=&time=&tradeAmt=&merNo=001&payOrderNo=

|  |  |
| --- | --- |
| **请求参数名称** | **说明** |
| sellAccount | 店铺名称 |
| verifyCode | 验证码 |
| respCode | 接口返回值 |
| respDesc | 接口返回描述 |
| time | 核销时间 |
| tradeAmt | 不知道 |
| merNo | 商家序号，固定001 |
| payOrderNo | 订单号 |

* 查看核销码状态

/channel/jpay/tbstatus?payOrderNo=&merNo=&verifyCode=

|  |  |
| --- | --- |
| **请求参数名称** | **说明** |
| payOrderNo | 订单号 |
| merNo | 商家序号，固定001 |
| verifyCode | 验证码 |