GoSSIP Summer School, 2017.7.11

Android应用程序
第三方支付安全

杨文博

密码与计算机安全实验室 (LoCCS)

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

自我介绍

- GoSSIP @ LoCCS
- 移动安全
- NDSS, RAID, AsiaCCS

提纲

- 第三方支付整体模型及流程
- 安全最佳实践及威胁模型
- 第三方支付中的可行攻击
- 安全漏洞及检测方法
- 实例及现状分析

移动平台第三方支付

- 移动支付市场巨大
- 改变生活方式，带来巨大便利
- 中国科技进步的标志之一

移动平台第三方支付

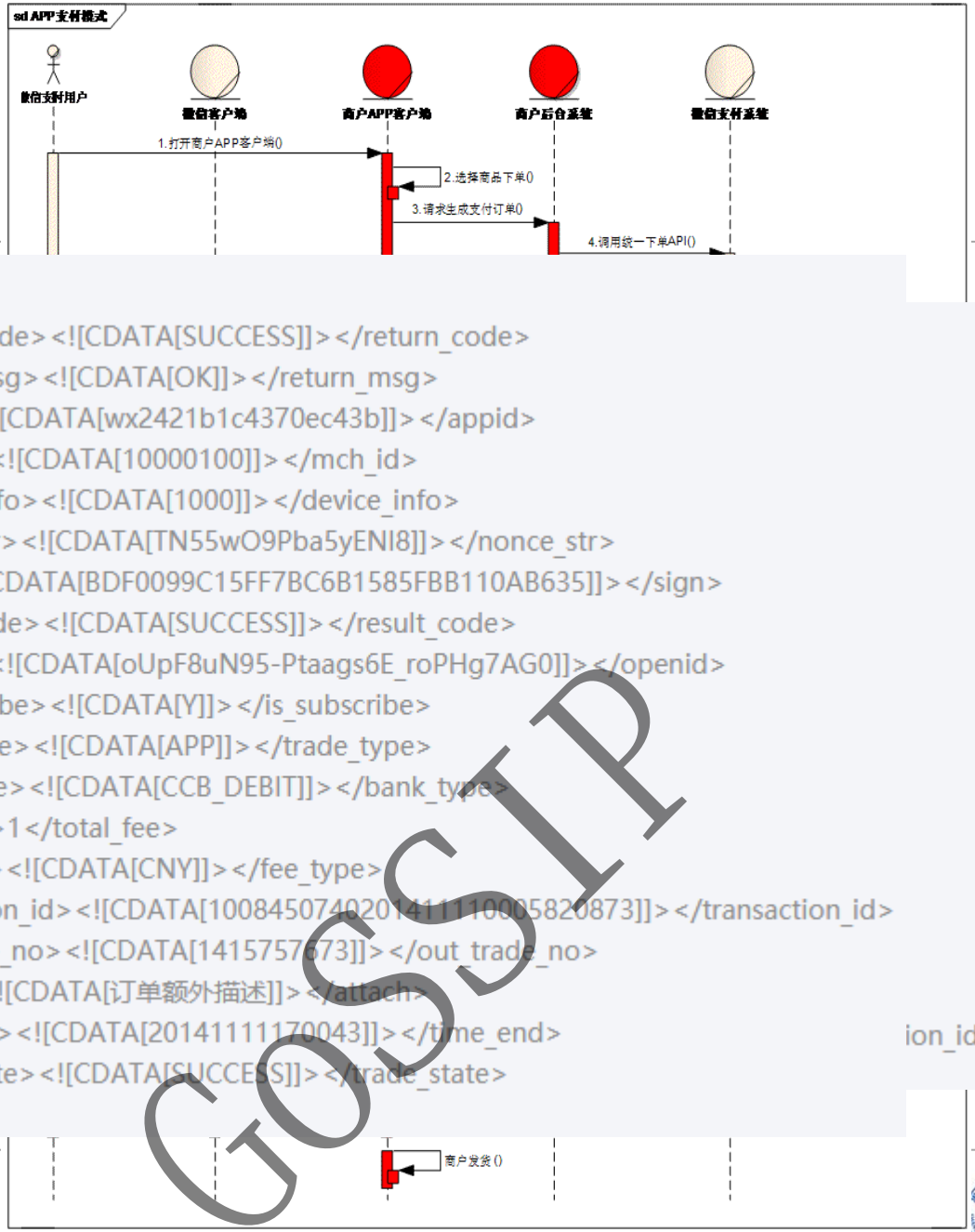
- APP应用内支付
- 电商、餐饮、出行、娱乐...
- 支付宝/微信/银联/百度...
- 纯在线过程
- 攻击成本低，隐蔽

APP第三方支付——用户

APP第三方支付——商户

- 选择第三方支付平台
- 注册/申请
- 获取文档/SDK
- 开发/测试
- 正式发布上线

微信支付




```
<xml>
  <appid>wx242
  <attach>支付测
  <body>APP支
  <mch_id>1000
  <nonce_str>1a
  <notify_url>htt
  <out_trade_no
  <spbill_create_
  <to  <xml>
  <tra    <return_
  <sig    <return_
</xm      <appid>
          <mch_id
          <nonce_
          <sign>
          <result_
          <prepay
          <trade_
        </xml>
```

```
<xml>
  <return_code><![CDATA[SUCCESS]]></return_code>
  <return_msg><![CDATA[OK]]></return_msg>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <mch_id><![CDATA[10000100]]></mch_id>
  <device_info><![CDATA[1000]]></device_info>
  <nonce_str><![CDATA[TN55wO9Pba5yENl8]]></nonce_str>
  <sign><![CDATA[BDF0099C15FF7BC6B1585FBB110AB635]]></sign>
  <result_code><![CDATA[SUCCESS]]></result_code>
  <openid><![CDATA[oUpF8uN95-Ptaags6E_roPHg7AG0]]></openid>
  <is_subscribe><![CDATA[Y]]></is_subscribe>
  <trade_type><![CDATA[APP]]></trade_type>
  <bank_type><![CDATA[CCB_DEBIT]]></bank_type>
  <total_fee>1</total_fee>
  <fee_type><![CDATA[CNY]]></fee_type>
  <transaction_id><![CDATA[1008450740201411110005820873]]></transaction_id>
  <out_trade_no><![CDATA[1415757673]]></out_trade_no>
  <attach><![CDATA[订单额外描述]]></attach>
  <time_end><![CDATA[20141111170043]]></time_end>
  <trade_state><![CDATA[SUCCESS]]></trade_state>
</xml>
```

微信支付

第一步：对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

```
stringA="appid=wxd930ea5d5a258f4f&body=test&device_info=1000&mch_id=10000100&nonce_str=ibuaiVcKdpRxkhJA";
```

第二步：拼接API密钥：

```
stringSignTemp=stringA+"&key=192006250b4c09247ec02edce69f6a2d" //注：key为商户平台设置的密钥key
sign=MD5(stringSignTemp).toUpperCase()="9A0A8659F005D6984697E2CA0A9CF3B7" //注：MD5签名方式
sign=hash_hmac("sha256",stringSignTemp,key) //注：HMAC-SHA256签名方式
```

最终得到最终发送的数据：

```
<xml>
<appid>wxd930ea5d5a258f4f</appid>
<mch_id>10000100</mch_id>
<device_info>1000</device_info>
<body>test</body>
<nonce_str>ibuaiVcKdpRxkhJA</nonce_str>
<sign>9A0A8659F005D6984697E2CA0A9CF3B7</sign>
</xml>
```

支付宝支付


```php
$aop = new AopClient;
$aop->gatewayUrl = "https://openapi.alipay.com/gateway.do";
$aop->appId = "app_id";
$aop->rsaPrivateKey = '请填写开发者私钥去头去尾去回车，一行字符串';
$aop->format = "json";
$aop->charset = "UTF-8";
$aop->signType = "RSA2";
$aop->alipayrsaPublicKey = '请填写支付宝公钥，一行字符串';
//实例化具体API对应的request类,类名称和接口名称对应,当前调用接口名称：alipay.trade.app.pay
$request = new AlipayTradeAppPayRequest();
//SDK已经封装掉了公共参数，这里只需要传入业务参数
$bizcontent = "{\"body\":\"我是测试数据\","
            . "\"subject\": \"App支付测试\","
            . "\"out_trade_no\": \"20170125test01\","
            . "\"timeout_express\": \"30m\","
            . "\"total_amount\": \"0.01\","
            . "\"product_code\":\"QUICK_MSECURITY_PAY\""
            . "}";
$request->setNotifyUrl("商户外网可以访问的异步地址");
$request->setBizContent($bizcontent);
//这里和普通的接口调用不同，使用的是sdkExecute
$response = $aop->sdkExecute($request);
//htmlspecialchars是为了输出到
传输不会有这个问题
echo htmlspecialchars($respo
```

```php
$aop = new AopClient;
$aop->alipayrsaPublicKey = '请填写支付宝公钥，一行字符串';
$flag = $aop->rsaCheckV1($_POST, NULL, "RSA");
```

```
    }
};

Thread payThread = new Thread(payRunnable);
payThread.start();
```

支付宝支付

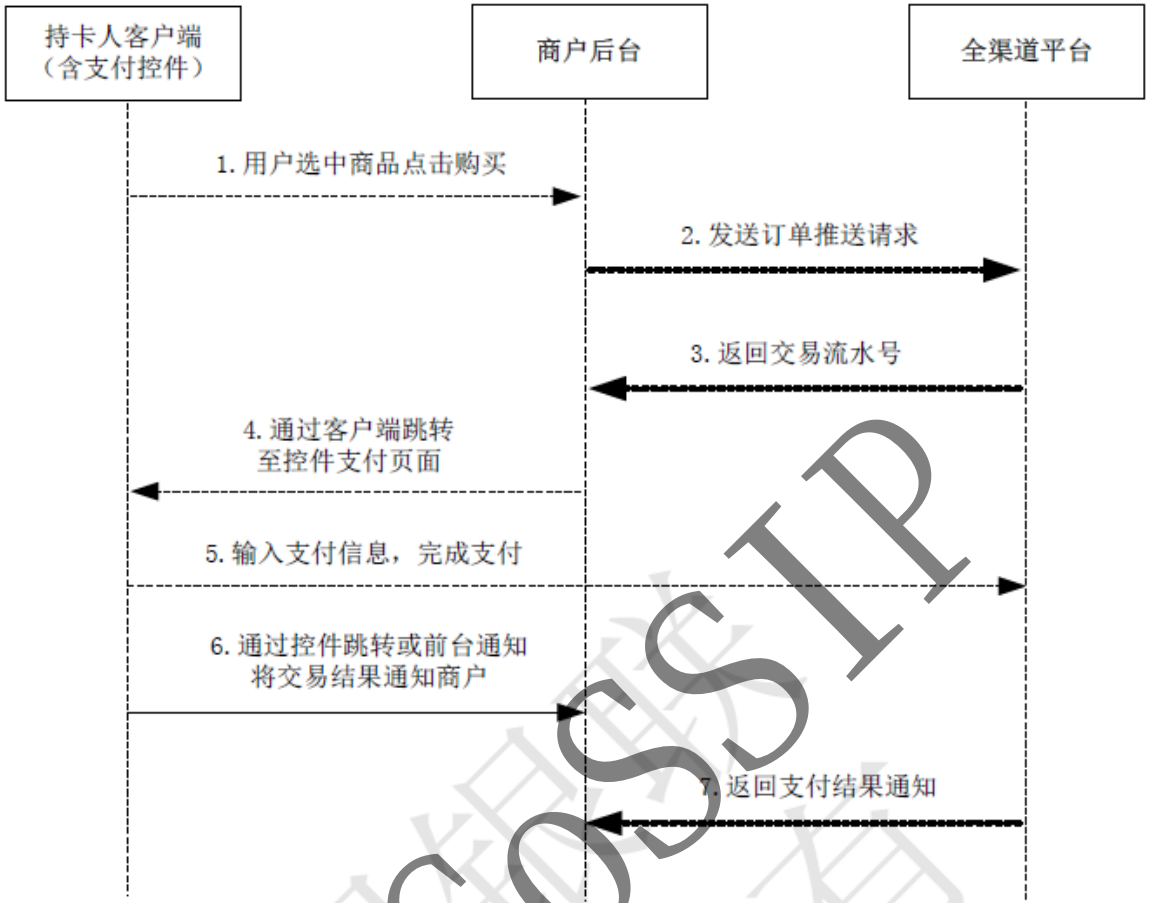

```
public class SignUtils {

    private static final String ALGORITHM = "RSA";

    private static final String SIGN_ALGORITHMS = "SHA1WithRSA";

    private static final String SIGN_SHA256RSA_ALGORITHMS = "SHA256WithRSA";

    private static final String DEFAULT_CHARSET = "UTF-8";

    private static String getAlgorithms(boolean rsa2) {
        return rsa2 ? SIGN_SHA256RSA_ALGORITHMS : SIGN_ALGORITHMS;
    }

    public static String sign(String content, String privateKey, boolean rsa2) {
        try {
            PKCS8EncodedKeySpec priPKCS8 = new PKCS8EncodedKeySpec(
                    Base64.decode(privateKey));
            KeyFactory keyf = KeyFactory.getInstance(ALGORITHM);
            PrivateKey priKey = keyf.generatePrivate(priPKCS8);

            java.security.Signature signature = java.security.Signature
                    .getInstance(getAlgorithms(rsa2));

            signature.initSign(priKey);
            signature.update(content.getBytes(DEFAULT_CHARSET));

            byte[] signed = signature.sign();

            return Base64.encode(signed);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return null;
    }
}
```

百度支付

通知参数：

参数名	参数值	参数值的说明
sp_no	1234567890	10 位数字组成的字符串
order_no	20080808123456123456	订单号
bfb_order_no	20080808BFB20080808123456123456	百度钱包支付交易号
bfb_order_create_time	20080808080808	百度钱包支付交易创建时间
pay_time	20080808090909	支付时间
pay_type	2	默认支付方式
unit_amount	1000	商品单价为 10 元
unit_count	2	订单包含 2 件相同的商品
transport_amount	500	运费为 5 元
total_amount	2500	买家需要支付 25 元
fee_amount	0	手续费 0 元
currency	1	支付的币种是人民币
buyer_sp_username	jarfield	卖家在商户网站的用户名
pay_result	1	支付成功
input_charset	1	中文编码是 GBK
version	2	版本号为 2
sign	B219D1A2784C1F12868FEE887374AFB2	签名结果
sign_method	1	签名算法为 MD5

银联支付


持卡人客户端
（含支付控件）
商户后台
全渠道平台
1. 用户选中商品点击购买
2. 发送订单推送请求
3. 返回交易流水号
4. 通过客户端跳转
至控件支付页面
5. 输入支付信息，完成支付
6. 通过控件跳转或前台通知
将交易结果通知商户
7. 返回支付结果通知

简化模型

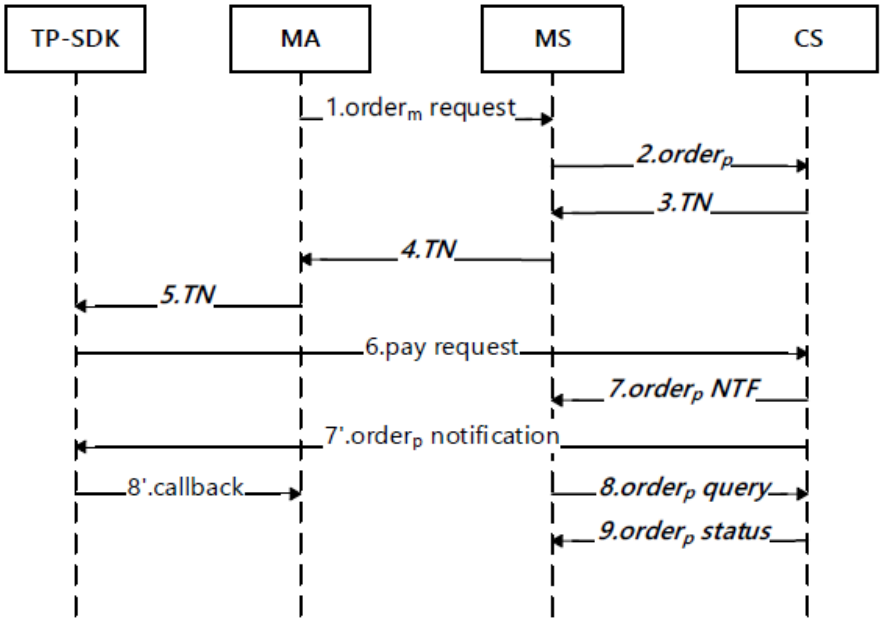


Fig. 1: In-app Payment Process Model I adopted by WexPay and UniPay

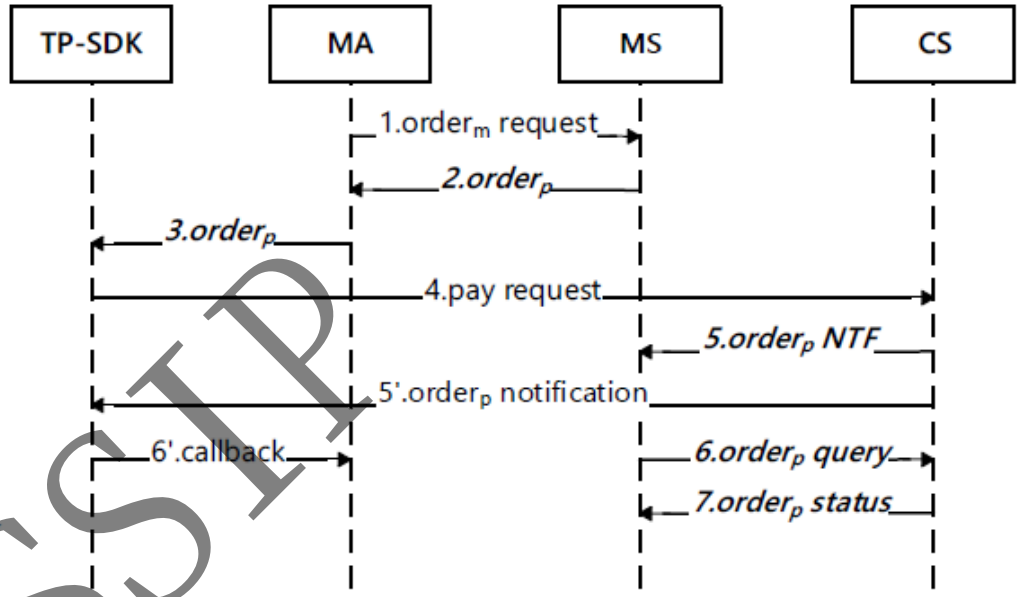


Fig. 2: In-app Payment Process Model II adopted by AliPay and BadPay

安全分析

- 威胁模型
- 最佳安全实践
- 可行攻击

威胁模型

- 攻击者可以逆向商户APP，SDK
- 向商户/支付商服务器发送请求或数据
- 若攻击针对商户/支付商
  - 恶意用户
  - 控制设备及APP的运行
- 若攻击针对普通用户
  - 控制网络数据
  - 可进行中间人攻击（如ARP欺骗，钓鱼WiFi）

安全实践

- 支付订单必须由商户服务器生成或签名
- 不应暴露任何秘密信息（如签名密钥）
- 支付商SDK因显示支付订单所有详情
- 支付商SDK需校验交易所属APP
- 使用安全的网络通信
- 服务器必须发货前做进一步的确认查询
- 服务器必须校验每次接受信息的签名

订单篡改攻击

- 没有在服务器生成（签名）支付订单
- 没有进行后续确认

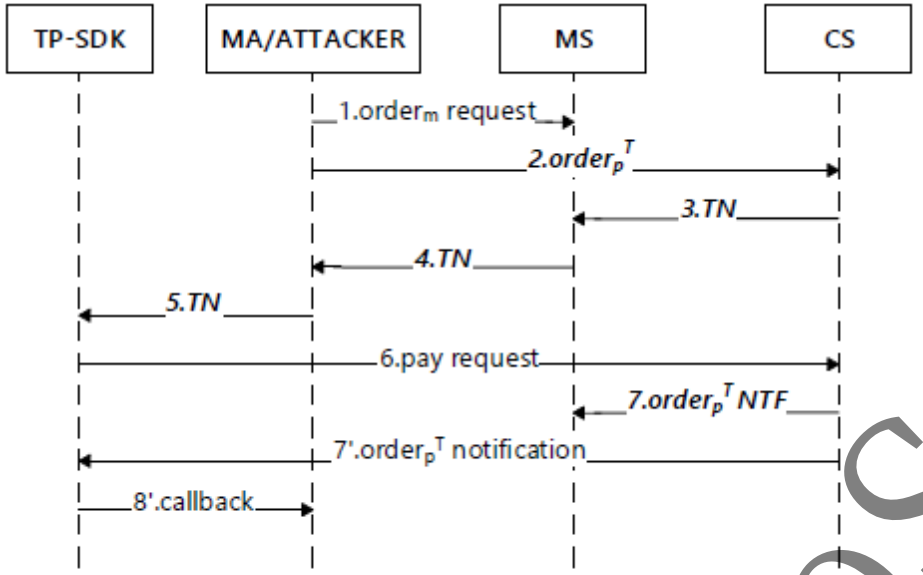


Fig. 3: Order Tampering Attack to Process Model I

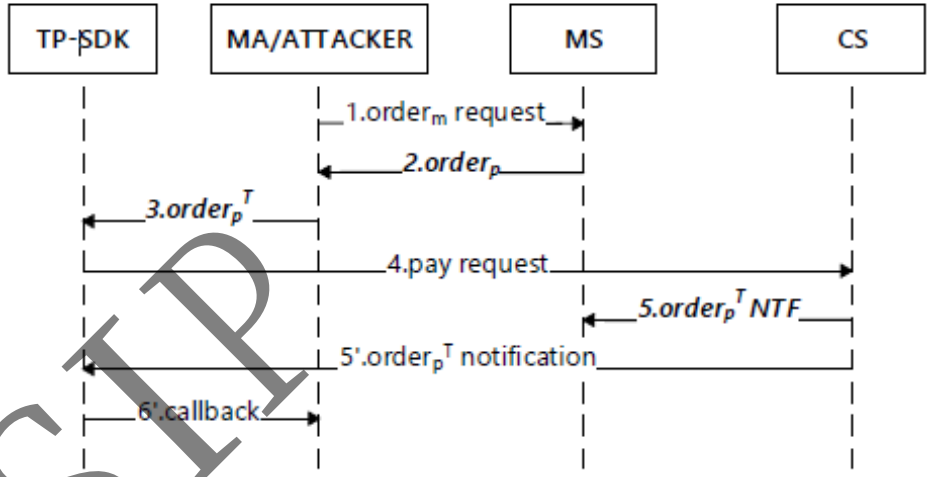


Fig. 4: Order Tampering Attack to Process Model II

支付结果伪造攻击

- 服务器没有校验消息签名（或泄露了密钥）
- 没有进行后续订单确认
- 不花钱购买商品

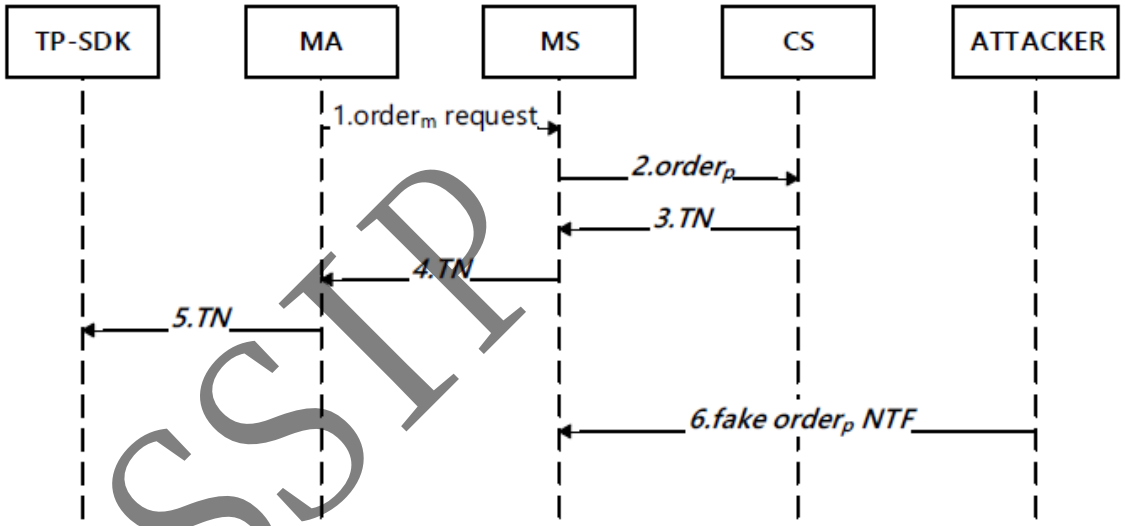


Fig. 5: Notification Forging Attack to Process Model I

支付订单替换攻击

- 针对普通用户的攻击
- 商户APP和服务器之间不安全的网络通信
- 支付商SDK支付订单信息显示不全
- 支付商SDK没有校验订单所属APP
- 将普通用户正常支付订单替换成攻击者订单，让用户支付攻击者订单

支付订单替换攻击

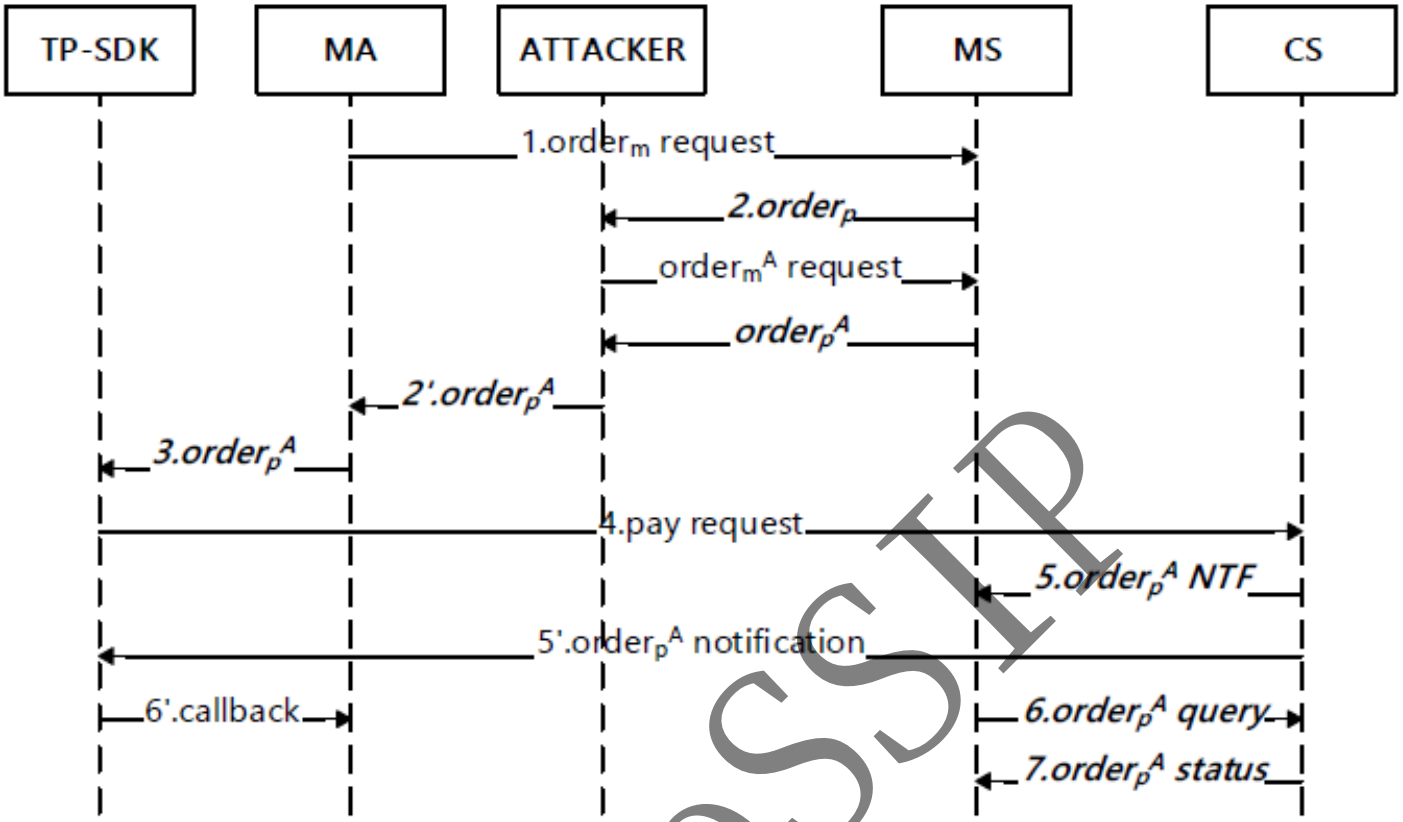


Fig. 6: Order Substituting Attack to Process Model II

越权查询攻击

- 泄露签名密钥
- 伪装成商户向支付商查询信息
- 订单详情，历史账单等

漏洞分析

- 识别
- 漏洞模型及检测
- 攻击实践

识别

```
if(GlobalConstants.n) {
    if(this.c.startsWith("https://wappaygw.alipay.com/home/exterfaceAssign.htm?")) {
        this.c = this.c.substring(this.c.indexOf("https://wappaygw.alipay.com/home/exterfaceAssign.htm?")
                + 53);
    }
    else if(this.c.startsWith("https://mclient.alipay.com/home/exterfaceAssign.htm?")) {
        this.c = this.c.substring(this.c.indexOf("https://mclient.alipay.com/home/exterfaceAssign.htm?")
                + 52);
    }
    else if(this.c.startsWith("http://mcashier.stable.alipay.net/home/exterfaceAssign.htm?")
            ) {
        this.c = this.c.substring(this.c.indexOf("http://mcashier.stable.alipay.net/home/exterfaceAssign.htm?")
                + 59);
    }
    else if(this.c.startsWith("http://mobileclientgw.stable.alipay.net/home/exterfaceAssign.htm?")
            ) {
        this.c = this.c.substring(this.c.indexOf("http://mobileclientgw.stable.alipay.net/home/exterfaceAssign.htm?")
                + 65);
    }
    this.b(arg3);
    this.startPage(arg3, "http://m.baidu.com/lightapp/3345414?page=" + arg4);
}
else {
    GlobalUtils.toast(arg3, ResUtils.getString(arg3, "bd_wallet_o2o_error"));
}
}
else {
    v0_1 = v0.a(this.c + "&bizcontext={\"appkey\":\"2014052600006128\"}");
}
}
```

漏洞检测

统一下单

应用场景

商户系统先调用该接口在微信支付服务后台生成预支付交易单，返回正确的预支付交易回话标识后再在APP里面调起支付。

接口链接

URL地址：https://api.mch.weixin.qq.com/pay/unifiedorder

是否需要证书

不需要

漏洞检测


```
public
    St
    try
        <xml>
         <appid>wx2421b1c4370ec43b</appid>
         <bill_date>20141110</bill_date>
         <bill_type>ALL</bill_type>
         <mch_id>10000100</mch_id>
         <nonce_str>21df7dc9cd8616b56919f20d9f679233</nonce_str>
         <sign>332F17B766FC787203EBE9D6E40457A1</sign>
        </xml>
    }
    ca
    }
    re
}
```

接口链接

https://api.mch.weixin.qq.com/pay/downloadbill

是否需要证书

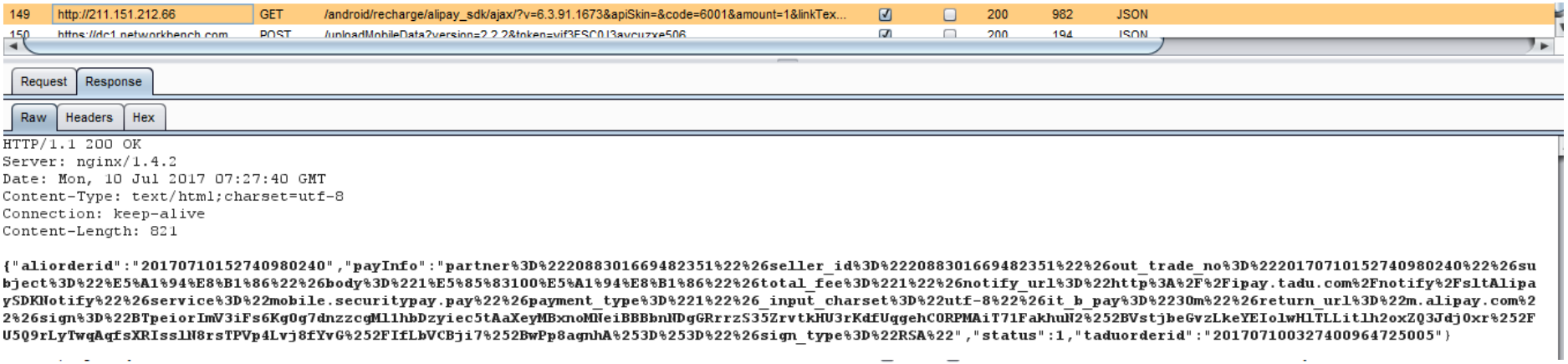
名称	描述	原因	解决方案
SYSTEMERROR	下载失败	系统超时	请尝试再次查询。
invalid bill_type	参数错误	请求参数未按指引进行填写	参数错误，请重新检查
data format error			
missing parameter			
SIGN ERROR			
NO Bill Exist	账单不存在	当前商户号没有已成交的订单，不生成对账单	请检查当前商户号在指定日期内是否有成功的交易。
Bill Creating	账单未生成	当前商户号没有已成交的订单或对账单尚未生成	请先检查当前商户号在指定日期内是否有成功的交易，如指定日期有交易则表示账单正在生成中，请在上午10点以后再下载。
CompressGZip Error	账单压缩失败	账单压缩失败，请稍后重试	账单压缩失败，请稍后重试
UnCompressGZip Error	账单解压失败	账单解压失败，请稍后重试	账单解压失败，请稍后重试

MIICdwIE…JNFmhT2jsz05IkxOgf
wsRZgBs/…8uAB4TpmrarOMUGtX
6cL+Mjgr…+ogUOYu1nPJlnMBQi9
pnne/SVC…vOxgPTEVXL/FI+baDb
i9xF3/acc…4sY0Um7Ydqd8Na9Tz
DE7H+ml…QDB72suLtIlRdymfdb
RP0WgY+…/FAkEA65poQA6ALzoh
dV4+dbF…fKQJAPl+YYzQf1mkPv
GSAsK/e3…CCWQJBAIT4094BpLf
+0Qma0/…/DV2IlM=

漏洞检测


确认付款
¥5.00
收款方
百度
订单信息
5阅读币
付款方式
余额宝
请验证指纹或者 输入密码
关闭
中文
确认交易
5阅读币
¥5.00
收款方
北京百度网讯科技有限公司
立即支付
返回
百度钱包
需付款
¥5.00
余额支付
请输入手机支付密码
忘记密码?
百度钱包安全键盘
1
2
3
4
5
6
7
8
9
0

漏洞检测



149	http://211.151.212.66	GET	/android/recharge/alipay_sdk/ajax/?v=6.3.91.1673&apiSkin=&code=6001&amount=1&linkTex...	☑	☐	200	982	JSON
150	https://dc1.networkbench.com	POST	/uploadMobileData?version=2.2.2&token=vif3ESC0J3avcuzxe506	☑	☐	200	194	JSON

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Server: nginx/1.4.2
Date: Mon, 10 Jul 2017 07:27:40 GMT
Content-Type: text/html;charset=utf-8
Connection: keep-alive
Content-Length: 821

{"aliorderid":"20170710152740980240","payInfo":"partner%3D%2220883016694823 51%22%26seller_id%3D%2220883016694823 51%22%26out_trade_no%3D%22017071015274098024 0%22%26subject%3D%22%E5%A1%94%E8%B1%86%22%26body%3D%221%E5%85%83100%E5%A1%94%E8%B1%86%22%26total_fee%3D%221%22%26notify_url%3D%22http%3A%2F%2Fipay.tadu.com%2Fnotify%2FsltAlipaySDKNotify%22%26service%3D%22mobile.securitypay.pay%22%26payment_type%3D%221%22%26_input_charset%3D%22utf-8%22%26it_b_pay%3D%2230m%22%26return_url%3D%22m.alipay.com%22%26sign%3D%22BTpeiorImV3iFs6Kq0q7dnzzcgMl1hbDzyiec5tAaXeyMBxnoMNeiBBBbnNDgGRrrzS35ZrvtkHU3rKdfUqgehCORPMAiT71FakhuN2%252BVstjbeGvzLkeYEIolwHlTLLitlh2oxZQ3Jdj0xr%252FU5Q9rLyTwqAqfsXRIsslN8rsTPVp4Lvj8fYvG%252FIfLbVCBji7%252BwPp8agnhA%253D%253D%22%26sign_type%3D%22RSA%22","status":1,"taduorderid":"20170710032740096472 5005"}
```

278	https://cashier.95516.com	POST	/b2c/api/Pay.action	☑	☐	302	733	text	
279	https://www.google.com	POST	/loc/m/api	☑	☐				
280	https://mcashier.95516.com	GET	/mobile/zh_CN/index.action?sign=499588679cefffaebaec1e3093c6f0ce&transNumber=856...	☑	☐	200	8606	HTML	ä¸-å□½é□¶è□□

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 302 Moved Temporarily
Date: Mon, 10 Jul 2017 07:40:21 GMT
Server: UPJAS
Cache-Control: no-store
P3P: CP='IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT'
Pragma: no-cache
Expires: Wed, 31 Dec 1969 23:59:59 GMT
Location:
https://mcashier.95516.com/mobile/zh_CN/index.action?sign=499588679cefffaebaec1e3093c6f0ce&transNumber=856869222994163639504&internalLogId=AC10170710154015970c0093024339&locale=zh_CN
Content-Language: zh-CN
Set-Cookie: default-cookie-name=TVdKZOLcAqutd4byb+EIUKUo; Path=/b2c
Set-Cookie: up_b4=AC10170710154015970c0093024339; Version=1; Path=/; Secure; HttpOnly
Content-Length: 0
Content-Type: text/plain
X-Via: 1.1 jf251:0 (Cdn Cache Server V2.0)
Connection: keep-alive
```

漏洞检测

，查看

漏洞检测

```
if (lpparam.packageName.equals("com.edaixi.activity"))
{

    Class spiderOrder = XposedHelpers.findClass("com.edaixi.baidupay.BaiDuPayUtil", lpparam.classLoader);
    XposedBridge.hookAllMethods(spiderOrder, "createOrderInfo", new XC_MethodHook() {
        @Override
        protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
            XposedBridge.log((String)param.args[1]);
            XposedBridge.log((String)param.args[2]);
            param.args[1] = "1";



        }

        @Override
        protected void afterHookedMethod(MethodHookParam param) throws Throwable {
            //XposedBridge.log("signParamsNative: " + (String)param.getResult());

        }

    });
}
```

漏洞检测

- 服务器
- 违反安

```
<xml>
  <appid><![CDATA[wx2421b1c4370ec43b]]></appid>
  <attach><![CDATA[支付测试]]></attach>
  <bank_type><![CDATA[CFT]]></bank_type>
  <fee_type><![CDATA[CNY]]></fee_type>
  <is_subscribe><![CDATA[Y]]></is_subscribe>
```

某商户设置的通知地址为https://api.xx.com/receive_notify.htm，对应接收到通知的示例如下：

注：以下示例报文仅供参考，实际返回的详细报文请以实际返回为准。

```
1  https://api.xx.com/receive_notify.htm?total_amount=2.00&buyer_id=2088102116773037&body=
   大乐透2.1&trade_no=2016071921001003030200089909&refund_fee=0.00&notify_time=2016-07
   -19 14:10:49&subject=大乐透2.1&sign_type=RSA2&charset=utf-8&notify_type=trade_status_syn
   c&out_trade_no=0719141034-6418&gmt_close=2016-07-19 14:10:46&gmt_payment=2016-07-1
   9 14:10:47&trade_status=TRADE_SUCCESS&version=1.0&sign=kPbQIjX+xQc8F0/A6/AocELIjhhZ
   nGbcBN6G4MM/FmfWL4ZiHM6fWI5NQhzXJusakIZ1LFuMo+IHQUELAYeugH8LYFvxnNajOvZhux
   NFbN2LhF0I/KL8ANtj8oyPM4NN7Qft2kWJTDJUpQOzCzNnV9hDxh5AaT9FPqRS6ZKxnzM=&gmt_
   create=2016-07-19 14:10:44&app_id=2015102700040153&seller_id=2088102119685838&notify
   _id=4a91b7a78a503640467525113fb7d8bg8e
```

```
</xml>
```

批量检测

```python
def tencent_vul(dx):

    #paths1 = dx.get_tainted_packages().search_methods("Lcom/tencent/m
    constStr = dx.get_strings()
    if "https://api.mch.weixin.qq.com/pay/unifiedorder" in constStr:
        #print "uppayuri"
        return 1
    return 0
```

```python
a, d, dx = session.get_objects_apk(APKFi
paths1 = dx.get_methods_descriptor("Lcom,
if len(paths1) > 0:
    return 1
```

```python
def get_xml_strings(a):
    st = []
    arscobj = a.get_android_resources()
    arscobj._analyse()
    for package_name in arscobj.get_packages_names():
        #print package_name
        for locale in arscobj.get_locales(package_name):
            try:
                for i in arscobj.values[package_name][locale]["string"]:
                    st.append(i[1])
            except KeyError:
                pass
    return st
```

```python
def alipay_vul(a, dx):

    constStr = get_all_strings(a,dx)

    for x in constStr:
        if len(x) > 500 and x.startswith("M"):
            r = r + x+"\t"
    return r
```

现状分析

- 第三方支付的流行度
- 三分之一

TABLE I: TP-SDK Distribution

Cashier	Number
WexPay	2260
AliPay	1299
UniPay	574
BadPay	34
Total	2679
Sample	7145

现状分析

- 应用程序中的安全问题
- APP & SDK

Cashier	KEY leakage	Local Ordering
WexPay	155	104
AliPay	398	/
UniPay	0	0
BadPay	7	/

TABLE II: Flaws in Merchant Apps

Cashier	Transaction Verification	Information Prompt					Network Communication
		orderID	commodity	owner	merchant	money	
WexPay	✓	×	✓	×	✓	✓	secure private protocal
AliPay	×	×	✓	×	×	✓	HTTPS pinning
UniPay	×	✓	✓	×	✓	✓	HTTPS pinning
BadPay	×	×	×	×	×	✓	HTTPS validation

TABLE III: flaws in TP-SDKs

现状分析

特别注意：

- 构造交易数据并签名必须在商户服务端完成，商户的应用私钥绝对不能保存在商户APP客户端中，也不能从服务端下发。
- 同步返回的数据，只是一个简单的结果通知，商户确定该笔交易付款是否成功需要依赖服务端收到支付宝异步通知的结果进行判断。
- 商户系统接收到通知以后，必须通过验签（验证通知中的sign参数）来确保支付通知是由支付宝发送的。建议使用支付宝提供的SDK来完成，详细验签规则参考异步通知验签。

泄漏导致出现“假

接口规则

3.返回签名后的订单信息

3. 商户支付请求参数的安全注意点：

a）请求参数的sign字段请务必在服务端完成签名生成（不要在客户端本地签名）；

b）支付请求中的订单金额total_amount，请务必依赖服务端，不要轻信客户端上行的数据（客户端本地上行数据在用户手机环境中无法确保一定安全）。

9.同步支付结果返回商户服务端，验签，解析支付结果

第五步： 在第四步签名验证通过后，必须严格按照如下的描述校验通知参数的合法性：

1、商户需要验证该通知数据中的out_trade_no是否为商户系统中创建的订单号；2、判断total_amount是否确实为该订单的实际金额（即商户订单创建时的金额）；3、校验通知中的seller_id（或者seller_email) 是否为out_trade_no这笔单据对应的操作方（有的时候，一个商户可能有多个seller_id/seller_email）；4、验证app_id是否为该商户本身。上述1、2、3、4有任何一个验证不通过，则表明同步校验结果是无效的，只有全部验证通过后，才可以认定买家付款成功。

联系我们

现状分析

- 不包含代码漏洞
- 集成第三方支付（多方）
- Demo
- 实例