

密级 ★

项目组公开

# Imkey 安全审计报告



主测人：孙浩然@knownsec.com

## 文档信息

文档名称	文档版本号	保密级别
Imkey安全审计报告	V2.1	公开

## 版权声明

本文件中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明，版权均属北京知道创宇信息技术有限公司所有，受到有关产权及版权法保护。任何个人、机构未经北京知道创宇信息技术有限公司的书面授权许可，不得以任何方式复制或引用本文件的任何片断。

## 目录

1. 综述.....	1
2. 代码漏洞分析.....	2
2.1 漏洞等级分布.....	2
2.2 审计结果汇总.....	2
3. 复测结果分析.....	6
软件部分： .....	6
3.1 调试信息泄露【安全】 .....	6
3.2 SHA-1 弱信息摘要算法【安全】 .....	6
3.3 CBC 模式下的 IV 值【安全】 .....	7
3.4 APP 代码未做混淆【安全】 .....	7
3.5 OLED 安全性测试【安全】 .....	8
测试方法： .....	8
3.6 输入 PIN 码时电磁干扰测试【安全】 .....	9
3.7 菜单界面电磁干扰测试【安全】 .....	10
3.8 控制界面电磁干扰测试【安全】 .....	11
4. 附录 A: 硬件钱包.....	13
5. 附录 B: 漏洞测试工具简介 .....	14
5.1 特斯拉线圈 .....	14
5.2 Burp Suite.....	14
5.3 jeb.....	14
5.4 jadx .....	15
5.5 IDAPro .....	15
5.6 知道创宇渗透测试人员专用工具包 .....	15

## 1. 综述

本次报告有效复测时间是从 2019 年 02 月 12 日开始到 2019 年 3 月 19 日结束，在此期间针对 Imkey硬件钱包及App 的安全性和规范性进行安全审计并以此作为报告统计依据。

此次测试中，知道创宇工程师对 Imkey硬件钱包、Android SDK（见第三章节）进行了全面的复测分析，以此来发现目标代码存在的安全问题。

### 本次 Imkey硬件钱包 安全审计结果： 安全

由于本次复测过程在非生产环境下进行，所有代码均由相关人员提供，检测过程均与相关接口人进行沟通，并在操作风险可控的情况下进行相关复测操作，以规避复测过程中的生产运营风险、代码安全风险。

#### 本次复测的目标信息：

模块名称
Imkey 硬件钱包
Android SDK

#### 本次项目复测人员信息：

姓名	邮箱/联系方式	职务
孙浩然	sunhr@knownsec.com	安全工程师
田林生	tianls@knownsec.com	安全工程师
韩志伟	hanzw@knownsec.com	安全工程师

## 2. 代码漏洞分析

### 2.1 漏洞等级分布

本次漏洞风险按等级统计：

漏洞风险等级个数统计表			
严重	高危	中危	低危
0	0	0	0

### 2.2 审计结果汇总

钱包安全渗透测试服务内容			
测试项目	序号	测试内容	状态
钱包客户端-通信	1.	通信加密流程	安全
	2.	设备加密安全性	安全
	3.	会话密钥更新机制	安全
	4.	入侵行为检测	安全
	5.	可信加密模块	安全
钱包客户端-API接口	6.	加密过程	安全
	7.	调试信息	安全
	8.	组件通信	安全
	9.	通信过程	安全
	10.	中间人攻击	安全

钱包客户端-升级功能	11.	升级认证	安全
	12.	钱包客户端升级通道的安全性	安全
	13.	权限控制	安全
	14.	升级固件的可篡改性	安全
	15.	固件升级校验	安全
	16.	固件安全保护	安全
硬件钱包安全测试	17.	发行机制	安全
	18.	硬件钱包固件检测	安全
	19.	硬件钱包SDK渗透测试	安全
	20.	硬件钱包安全风险分析	安全
	21.	硬件钱包脚本代码分析	安全
	22.	钱包硬件电路板设计检测	安全
	23.	钱包硬件核心元件渗透测试	安全
	24.	钱包硬件通信模组渗透测试	安全
	25.	钱包硬件存储数据提取渗透测试	安全
	26.	硬件钱包生产测试遗留接口测试	安全
	27.	硬件钱包整体设计的安全性及稳定性测试	安全
	28.	硬件钱包内存重要信息读取	安全
	29.	硬件钱包抗电磁干扰测试	安全
	30.	非加密更新	安全

钱包硬件-更新机制	31.	更新位置为可写	安全
	32.	无手动更新机制	安全
	33.	缺乏更新机制	安全
	34.	更新过程中的安全防护	安全
	35.	API调用滥用	安全
钱包硬件-物理接口	36.	删除存储介质	安全
	37.	JTAG / SWD接口	安全
	38.	In-Situ dumping	安全
	39.	拦截OTA更新	安全
	40.	从制造商网页进行下载	安全
区块链相关安全测试	41.	真假随机数算法测试	安全
	42.	私钥生产及存放机制	安全
	43.	Bip39安全机制	安全
	44.	助记词安全机制	安全
钱包硬件-设备固件	45.	固件和存储提取	安全
	46.	数据存储安全	安全
	47.	内部通信数据校验	安全
	48.	固件加解密过程	安全
	49.	侧面渠道攻击- Glitching攻击	安全
	50.	开放端口检测	安全

钱包硬件操作系统	51.	系统级服务接口检测	安全
	52.	OLED安全性测试	安全
	53.	输入PIN码时电磁干扰测试	安全
	54.	菜单界面电磁干扰测试	安全
	55.	控制界面电磁干扰测试	安全
Applet检测	1.	Applet安全签名问题	安全
	2.	Applet资源调用安全问题	安全
	3.	JavaSmartCard安全问题	安全
	4.	Applet加固风险	安全
	5.	Applet权限控制风险	安全
	6.	Applet沙箱运行风险	安全
	7.	组件调用风险	安全
	8.	JavaRuntime安全风险	安全
	9.	Applet权限绕过风险	安全

### 3. 检测结果分析

软件部分：

#### 3.1 调试信息泄露【安全】

**测试原理：**

在APP的开发过程中，为了方便调试，通常会使用log函数输出一些信息，这会让攻击者更加容易了解APP内部结构，方便破解和攻击，甚至有可能直接获取到有价值的隐私敏感信息。

**测试方法：**

手机连接pc端，打开cmd窗口，输入 adb logcat 即可看到log信息。

```
03-14 14:40:52.466 10161 10298 D imkey : ble >>>> 80CA004400
03-14 14:40:52.519 4230 W ContentTaskController: Invalid task was provided to stopTracking.
03-14 14:40:52.523 10161 10298 D imkey : ble <<<< 4C5758A135383753444B38324B35344BA9000
03-14 14:40:52.523 10161 10298 D imkey : LWXA587SDK02K54J
03-14 14:40:52.524 10161 10298 D imkey : http >>>> https://imkey.online:1000/imkey/seInfoQuery
03-14 14:40:52.524 10161 10298 D imkey : http >>>>
{"seid": "1902000000000860001010000000713", "sn": "LWXA587SDK02K54J", "stepKey": "01", "commandID": "seInfoQuery", "cardRetDataList": []}
03-14 14:40:52.549 4230 E NetworkScheduler: Unable to resolve correct action against
com.google.android.apps.walletnfccel/com.google commerce.tapandpay.android.clientconfig.sync.ClientConfigSyncService to instantiate callback. Not executing task.
03-14 14:40:52.549 10161 10298 D imkey : http <<<<
{"_ReturnCode": "00000000", "_ReturnMsg": "操作成功", "_ReturnData": {"seid": "1902000000000860001010000000713", "appUpdateInfo": [], "installedAppList": [{"instanceAid": "695F627463", "appVersion": "1.2"}, {"instanceAid": "695F657468", "appVersion": "1.2"}, {"instanceAid": "695F696D6B", "appVersion": "1.2"}], "nextStepKey": "end"}
03-14 14:40:52.553 4230 4230 W ContentTaskController: Invalid task was provided to stopTracking.
03-14 14:40:52.555 10161 10298 D imkey : ble >>>>
03-14 14:40:52.555 4230 4230 E NetworkScheduler: Unable to resolve correct action against
com.google.android.youtube/com.google.android.libraries.youtube.common.gcore.gcoresclient.gcm.impl.GcmTaskServiceDelegator to instantiate callback. Not executing task.
03-14 14:40:52.577 4230 4230 W ContentTaskController: Invalid task was provided to stopTracking.
03-14 14:40:52.599 4230 4230 E NetworkScheduler: Unable to resolve correct action against
com.google.android.youtube/com.google.android.libraries.youtube.common.gcore.gcoresclient.gcm.impl.GcmTaskServiceDelegator to instantiate callback. Not executing task.
03-14 14:40:52.650 4230 4230 W ContentTaskController: Invalid newTask was provided to startTracking.
03-14 14:40:52.672 4230 4230 W ContentTaskController: Invalid newTask was provided to startTracking.
03-14 14:40:52.697 4230 4230 W ContentTaskController: Invalid newTask was provided to startTracking.
03-14 14:40:53.098 10161 10298 D imkey : ble <<<<
6F5C8408A000000003000000A550734A06072A864886FC6B016000C060A2A864886FC6B02020101630906072A864886FC6B041110650B06092B8510864864020103660C060A2B060104012A026E01029F
6501FF9000
03-14 14:40:53.098 10161 10298 D imkey : ble >>>> 80CB800005DFFF028101
03-14 14:40:53.153 10161 10298 D imkey : ble <<<< 1902000000000008600010100000007139000
03-14 14:40:53.153 10161 10298 D imkey : ble >>>> 00A4040000
03-14 14:40:53.668 10161 10298 D imkey : ble <<<<
6F5C8408A000000003000000A550734A06072A864886FC6B016000C060A2A864886FC6B02020101630906072A864886FC6B03640B06092A864886FC6B041110650B06092B8510864864020103660C060A2B060104012A026E01029F
6501FF9000
```

**检测结果：**经检测，android 应用在运行中存在一些调试信息，与客户沟通后确认调试信息对应用无影响。

#### 3.2 SHA-1弱信息摘要算法【安全】

**位置：**

im.imkey.imkeylibrary.core.foundation.crypto.Hash

**测试方法：**

代码审计，审查代码中是否有用到弱信息摘要算法。

```

private static byte[] sha1(byte[] input, int offset, int length) {
    try {
        MessageDigest md = MessageDigest.getInstance("SHA");
        md.update(input, offset, length);
        return md.digest();
    } catch (Exception e) {
        throw new ImkeyException(Messages.IMKEY_SHA_EXCEPTION);
    }
}

```

检测结果：此处确实使用了弱信息摘要算法，与客户沟通后确认该处对整体通信无重要影响。

### 3.3 CBC模式下的IV值【安全】

漏洞位置：

org.consenlabs.tokencore.wallet.keystore.HDMnemonicKeystore

测试方法：

代码审计，审计代码中用到的算法是否存在例如AES算法中CBC模式的IV值是否是随机。例如该位置的AES算法，追溯到IV定义的代码位置，可以看到IV值在代码中是固定的。

```

public String getEncryptXpub() {
    String plainText = this.xpub;
    try {
        return BaseEncoding.base64().encode(AES.encryptByCBC(plainText.getBytes(), Hex.decode(XPubCommonKey128), Hex.decode(XPubCommonIv)));
    } catch (Exception e) {
        throw new TokenException(Messages.ENCRYPT_XPUB_ERROR);
    }
}

public final class HDMnemonicKeystore extends IMTKeystore implements EncMnemonicKeystore {
    static int VERSION = 44;
    public static String XPubCommonIv = "9C0C30889CBCC5E01AB5B2BB88715799";
    public static String XPubCommonKey128 = "B888D25EC8C12BD5043777B1AC49F872";
    private EncPair encMnemonic;
    private Info info;
    private String mnemonicPath;
    private String xpub;
}

```

检测结果：此处AES算法IV值对整体加密存储无影响，可忽略危害。

### 3.4 APP 代码未做混淆【安全】

**测试方法:**

对App进行反编译，脱壳，可以看到代码未做混淆。

```

import im.imkey.imkeylibrary.common.Messages;
import im.imkey.imkeylibrary.exception.ImkeyException;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES {

    public enum AESType {
        CTR,
        CBC
    }

    public static byte[] encryptByCBC(byte[] data, byte[] key, byte[] iv) {
        return doAES(data, key, iv, 1, AESType.CBC, "PKCS5Padding");
    }

    public static byte[] decryptByCBC(byte[] ciphertext, byte[] key, byte[] iv) {
        return doAES(ciphertext, key, iv, 2, AESType.CBC, "PKCS5Padding");
    }

    private static byte[] doAES(byte[] data, byte[] key, byte[] iv, int cipherMode, AESType type, String paddingType) {
        String aesType;
        if (type == AESType.CBC) {
            aesType = "CBC";
        } else {
            aesType = "CTR";
        }
        try {
            IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
            SecretKeySpec secretKeySpec = new SecretKeySpec(key, "AES");
            Cipher cipher = Cipher.getInstance(String.format("AES/%s/%s", new Object[]{aesType, paddingType}));
            cipher.init(cipherMode, secretKeySpec, ivParameterSpec);
            return cipher.doFinal(data);
        } catch (Throwable th) {
            ImkeyException imkeyException = new ImkeyException(Messages.IMKEY_AES_EXCEPTION);
        }
    }
}

```

**建议:**

对代码进行混淆，防止逆向分析。

**硬件部分：****3.5 OLED安全性测试【安全】****测试方法:**

OLED显示通常使用SPI或IIC与MCU通信。在硬件电路上，这几根通信引脚通常无需外围电路，并且会直接和MCU相连，通过测试加密芯片与OLED间直接导通的线路可以发现。

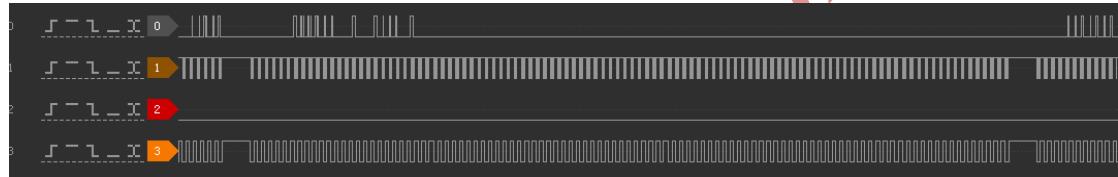
1. 分析电路走线。
2. 测试OLED的导通引脚。
3. 在引脚上飞线。
4. 使设备正常工作。
5. 使用逻辑分析仪捕捉数据。

**测试结果:**

经过测试有四根线是导通的，分别为pin6, pin7, pin10, pin11。先判断为SPI通信，通过捕捉波形，并进行数据分析，确认使用SPI通信。引脚功能分别为：

引脚	功能
PIN6	D/C
PIN7	CS
PIN10	CLK
PIN11	MOSI

在输入PIN码界面选择数据，捕捉这次波形并还原图像。以下为输入PIN界面时第一个数字输入7时捕捉到的波形：



波形图

但对数据进行分析时发现数据比较杂乱，未能还原。

### 3.6 输入PIN码时电磁干扰测试【安全】

说明：

正常设备在开启后会要求输入PIN码，输入正确进入菜单界面，如果在PIN码正在输入时放置干扰源就可以看到电磁干扰此时造成的影响。

测试方法：

1. 使设备正常工作。
2. 进入输入 PIN 码的界面。
3. 使用特斯拉线圈进行干扰，查看设备反应。

测试结果：

开始输入PIN码：



放置干扰源：



输入PIN码时进行电磁干扰，不影响正常启动

### 3.7 菜单界面电磁干扰测试【安全】

说明：

在进入菜单界面后，放置干扰源查看造成的影响。

测试方法：

1. 使设备正常工作。
2. 进入菜单界面。
3. 使用特斯拉线圈进行干扰，查看设备反应。

测试结果：

放置干扰源前：



放置干扰源之后：



在菜单界面时，进行电磁干扰会造成设备重启。

### 3.8 控制界面电磁干扰测试【安全】

说明：

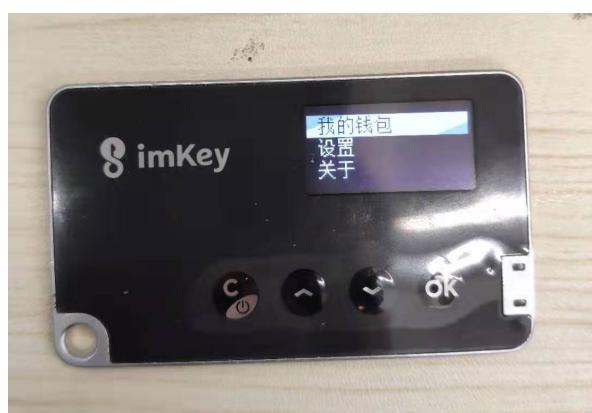
在进入控制界面后，放置干扰源查看造成的影响。

测试方法：

1. 使设备正常工作。
2. 进入控制界面。
3. 使用特斯拉线圈进行干扰，查看设备反应。

测试结果：

未放置干扰源前：

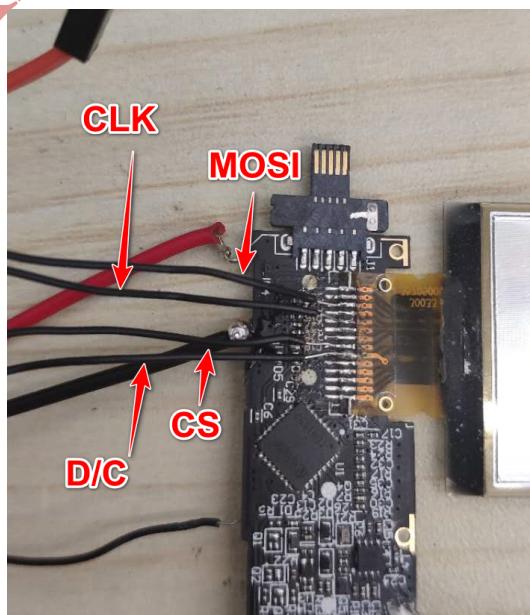
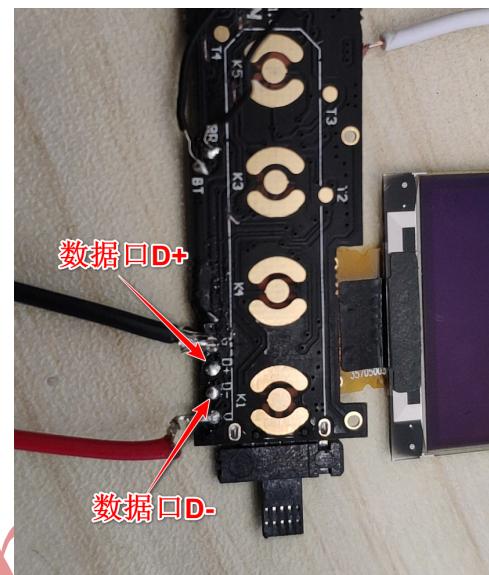


放置干扰源后：



在控制界面时，进行电磁干扰会造成OLED屏显示变暗

## 4. 附录 A：硬件钱包



## 5. 附录 B：漏洞测试工具简介

### 5. 1 特斯拉线圈

测试指标：场强3.1V/m，调制频率10Khz，在频段80~200MHZ进行扫频干扰测试。

### 5. 2 Burp Suite

Burp Suite 是用于攻击web 应用程序的集成平台，包含了许多工具。Burp Suite 为这些工具设计了许多接口，以加快攻击应用程序的过程。

Proxy：是一个拦截HTTP/S的代理服务器，作为一个在浏览器和目标应用程序之间的中间人，允许你拦截，查看，修改在两个方向上的原始数据流。

Spider：是一个应用智能感应的网络爬虫，它能完整的枚举应用程序的内容和功能。

Scanner[仅限专业版]：是一个高级的工具，执行后，它能自动地发现web 应用程序的安全漏洞。

Intruder：是一个定制的高度可配置的工具，对web应用程序进行自动化攻击，如：枚举标识符，收集有用的数据，以及使用fuzzing 技术探测常规漏洞。

Repeater：是一个靠手动操作来补发单独的HTTP 请求，并分析应用程序响应的工具。

Sequencer：是一个用来分析那些不可预知的应用程序会话令牌和重要数据项的随机性的工具。

Decoder：是一个进行手动执行或对应用程序数据者智能解码编码的工具。

Comparer：是一个实用的工具，通常是通过一些相关的请求和响应得到两项数据的一个可视化的“差异”。

### 5. 3 jeb

JEB is a reverse-engineering platform to perform disassembly, decompilation, debugging, and analysis of code and document files.

## 5. 4 jadx

<https://github.com/skylot/jadx> Command line and GUI tools for produce Java source code from Android Dex and Apk files.

## 5. 5 IDAPro

IDA is a Windows, Linux or Mac OS X hosted multi-processor disassembler and debugger that offers so many features it is hard to describe them all.

## 5. 6 知道创宇渗透测试人员专用工具包

知道创宇渗透测试人员专用工具包，由知道创宇渗透测试工程师研发，收集和使用，包含专用于测试人员的批量自动测试工具，自主研发的工具、脚本或利用工具等。



[ 咨询电话 ] +86(10)400 060 9587

[ 投诉电话 ] 13811527185

[ 邮 箱 ] sec@knownsec.com

[ 网 址 ] www.knownsec.com

[ 地 址 ] 北京市朝阳区望京SOHO T3 A座15层

