APP安全面临的新挑战与解决方案

APP安全问题严峻

手机改变我们的生活

- 智能手机使用者占比超过9成。
- 每天人均使用手机时间近3个小时。
- 每个手机平均安装50个APP。

APP安全问题严峻

当手机应用给我们带来便利的时候.....

A. 你是否遭遇过扣费？

B. 你是否遭遇过偷跑流量？

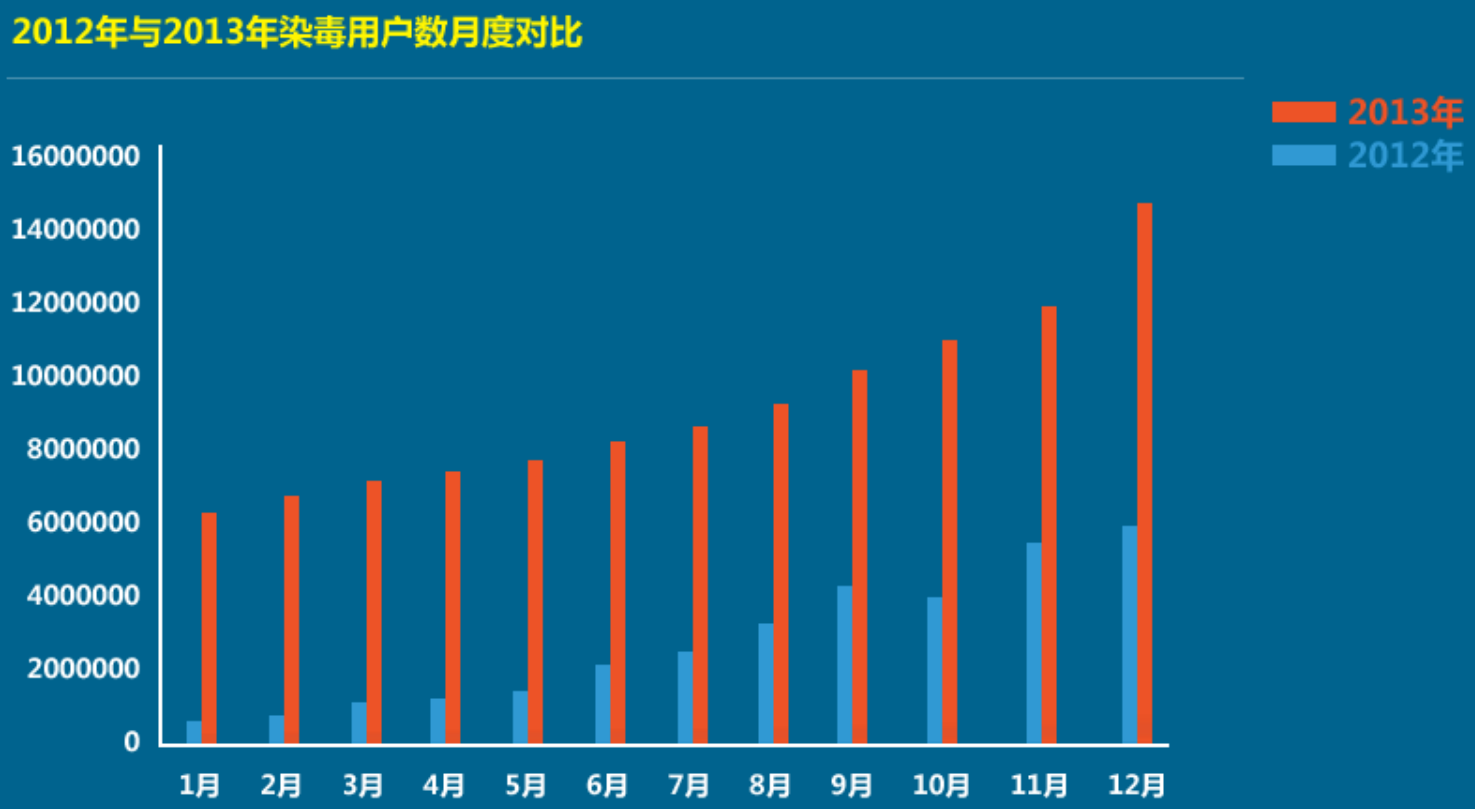
C. 你是否遭到骚扰短信和电话？

D. 你是否正在玩游戏却弹出恶意广告？

APP安全问题严峻

2013年97%的移动端恶意应用来自安卓

2013年共出现 1400 万恶意高风险 Android App



APP安全问题严峻

APK静态破解：四个工具简单破解APK，植入恶意代码：

① apktool ：将其解压缩文件和编译成APK
② dex2jar ：编码出其java源代码,用于了解其逻辑便于修改其文件
③ jd-gui ：查看java源码
④ eclipse：通过查看系统log输出 便于定位修改源代码
⑤ 签名工具：给编译成的二次apk签名 用于安装签名

APP安全问题严峻

APK动态破解：对正在运行时的APK进行动态破解

① 启动手机上面的apk，进行正常使用
② 使用ps命令找到运行apk对应的进程ID
③ 使用gdb命令连接运行时apk的进程
④ 使用gore命令对apk内存dex进行dump

APP安全问题严峻

破解应用的黑客都做了什么：

01 嵌入广告SDK赚取广告费用

02 加入恶意代码如：扣费，自动下载APK等

03 替换支付链接，导致用户向私人账户付费

04 收集用户隐私，贩卖用户信息

05 窃取金融APP账号密码，盗取资金

APP安全问题严峻

打包党恶意推广背后操作利益链

游戏APP安全挑战

黑客会对移动Android游戏应用做什么？

01 内购破解

02 游戏资源文件篡改

03 WPE刷量

04 内存修改

05 存档修改

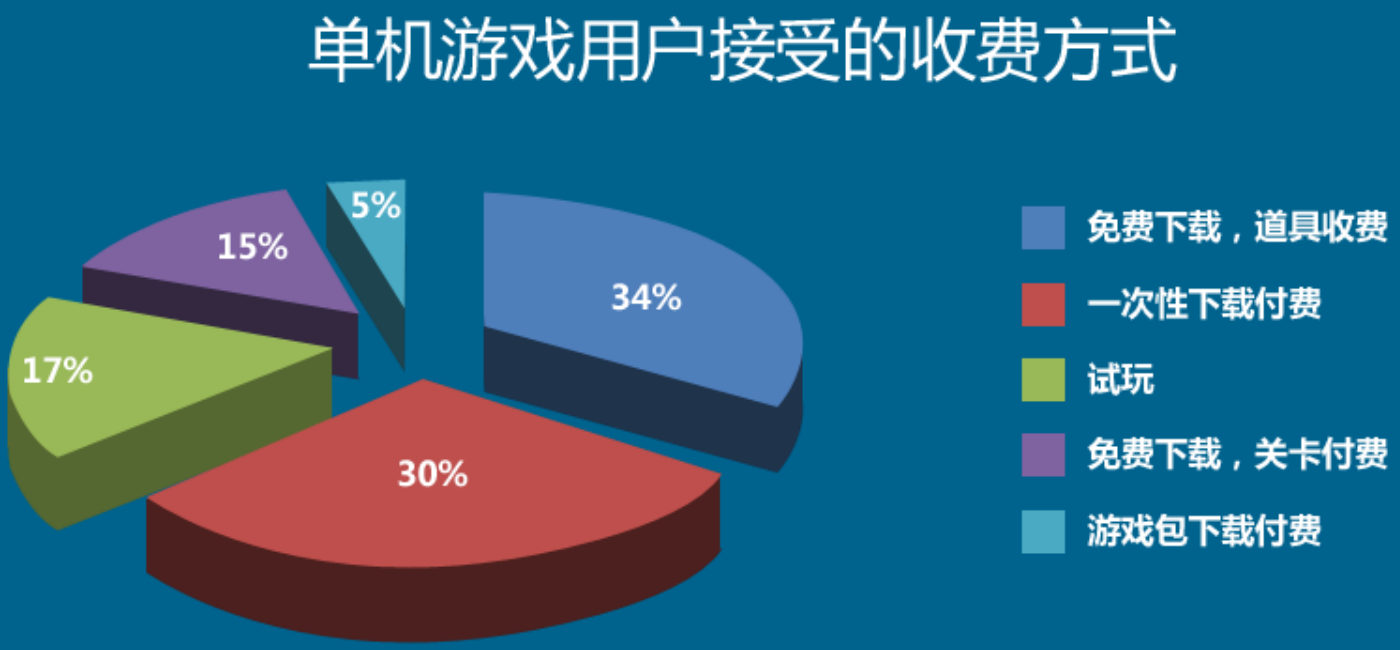
06 广告修改

07 注入恶意代码

08 篡改游戏规则

内购破解：

描述：所谓的内购指的是一些游戏通关内购道具或其他的方式收取玩家的费用，顾名思义，内购破解就是破解一个游戏的内购机制，从而达到免费玩游戏的目的。



内购破解：

常见的内购破解方式一：支付SDK破解

内购破解：

常见的内购破解方式一：短信支付破解

游戏资源文件修改：

描述：通过对APK中资源文件的修改，改变游戏背景音乐、图片、人物、地图等信息。

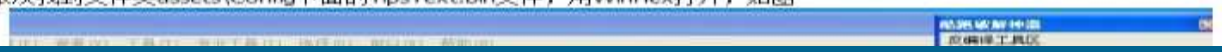
游戏资源文件修改：

3.下载完成之后，将我们自己的mp3音乐，用改软件转换成.Ogg格式就行了，替换文件夹里面的文件，重新打包安装！
u_bgm.ogg：游戏开始后的背景音乐
r_bgm_01.ogg -------r_bgm_04.ogg：游戏排行榜背景音乐，循环播放的！

五，讲解改人物特效，比如人物骑人物
1.依次找到文件夹 assets\Action\Character，如图

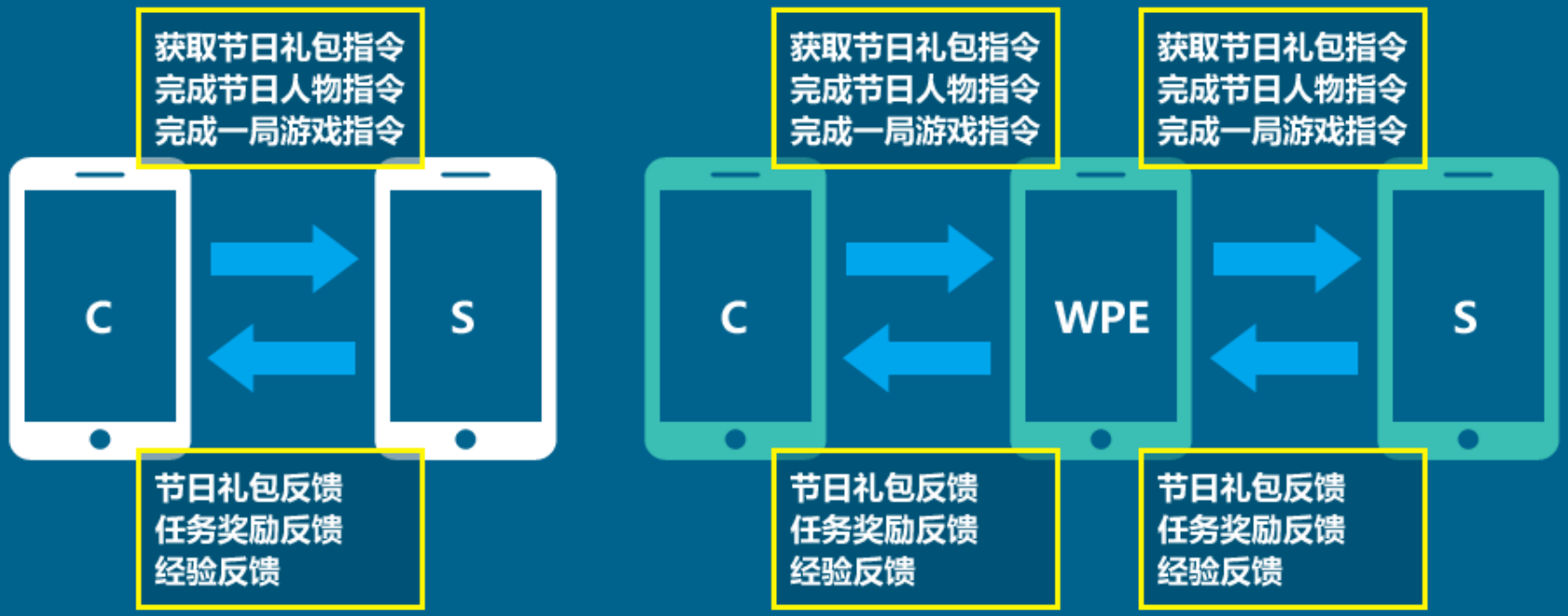
2.C开头的是人物文件，M开头的是坐骑文件，这具体我也没试验过，还是那句老话，改完之后需要重新以ZIP格式压缩签名，之后才能放到手机里面安装，不然会提示程序未安装！

六：讲解酷跑登陆后，tips提示内容的修改！需要用到工具C32ASM，和WinHex
1.依次找到文件夹assets\Config下面的TipsText.bin文件，用WinHex打开，如图



WPE刷量：

描述：WPE（Winsock Packet Editor）它的中文名称是：网络封包编辑器
在大多数的编程工具中winsock已经封装成一个控件，成为网络编程的控件，是非常方便的，利用这个控件，编程工具就可以编写外挂工具。



内存修改：

描述：通过工具修改运行游戏的内存数据。（修改金币、体力、法力、攻击等）

门户 论坛 新帖 悬赏 排行榜 站内搜索 吾爱云盘 站点帮助

查看: 2529 | 回复: 5

[Android] [redacted]修改金币=========攻略 [复制链接]

发表于 2014-4-5 21:16 | 只看该作者

[redacted]昨日正式登录安卓平台，不少的玩家已经加入游戏体验卡牌乐趣了。游戏中英雄主要通过金币或者钻石来抽取，下面教你如何通过八门神器修改[redacted]刷金币刷钻石！

[redacted]修改准备事项：

1、下载八门神器软件

2、手机需要ROOT

3、下载刀塔传奇游戏客户端

不提供下载地址了，自己百度吧

[redacted]八门神器刷金币刷钻石教程：

1、先下载安装八门神器到你的手机上，然后打开八门神器在后台运行，进入[redacted]游戏中。

2、打开八门神器在其中输入你游戏当前的金币数量，然后点击放大镜自动搜索。

3、再返回到游戏，消耗点金币或者钻石，让金币或者钻石数值发生变化。

4、然后回到八门神奇中搜索你之前变动的这个数，点击搜索后你可以看到结果变少了很多

5、接下来重复第四步、第五步，继续搜索得到的数值也就只剩下一个了，多个也是可以的，全部进行修改吧！

6、然后修改金币或者钻石数值。

7、返回游戏你就可以看到你的金币或者钻石修改成功了。

特别说明：

小编提醒玩家，本游戏修改数据可能有一定的风险，如被封号请勿喷！

通过以上内容，玩家你修改刷金币和钻石吧！

内存修改：

常用的内存修改器：

1）烧饼修改器
2）烧饼加速器
3）八门神器
4）葫芦侠修改器
5）叉叉修改器
6）安卓游戏加速器

存档修改：

描述：对游戏存档进行修改。（解锁关卡、装备等）

Android《口袋侦探》修改存档

2013年06月20日 来源：玩游戏网 编辑：猪猪 [已有0人评论][我要评论]

《口袋侦探》是一款ios及Android双平台解密游戏大作，在此分享的是安卓版修改存档。

口袋侦探修改说明：

1、四个人物共计20套服装全部解锁了；

2、侦探点的点数修改为了九千万。关于侦探点数，有三个作用，第一就是购买服装，第二就是让无法询问的嫌疑人立刻变为可询问状态，第三就是让失败的游戏从失败处重新尝试，避免从头再来的悲剧。

使用方法：

1、首先确保你的安卓设备已经root并安装了钛备份工具；

2、下载并解压口袋侦探安卓破解存档，可以得到一个TitaniumBackup文件夹；

3、将你的安卓设备和电脑连接，将第二步中解压出来的TitaniumBackup文件夹放置到手机SD卡根目录下并选择覆盖；

4、打开手机上的钛备份工具，找到口袋侦探安卓破解游戏程序，选择【恢复数据】即可。

广告修改：

描述：删除游戏中广告或替换游戏中广告的识别ID

[Android] 【全民汉化】Android软件汉化教程之图文第七课————Android去广告教程【已附视...

相信很多Android用户都特别讨厌广告，我也不例外，我是个看到广告就会心烦的人，所以学会了点去广告的方法，现在与大家分享，希望能帮助小白们找到相关的方法要靠大家去摸索。如果你有什么经验或者新方法可以跟帖回复与我们分享，我会更新进这篇教程里汇总，请大家踊跃参与，谢谢！

今天就以最常见的广告作为教程讲解下去广告的简单过程。首先，我们要了解广告的添加方法，才能更好的去除广告，达到完美去除广告的目的。每家公司的广告简单的广告还是很容易去除的。

广告通常都是有其固定的代码形式，修改了代码以后就会失去原本的功能，因此我们就能利用此方法来去除广告。通常情况下，去广告要做到两步就够了。

1.阻止广告请求，即广告错误，广告代码一般存在于res资源文件夹和smali文件夹中，去广告的原理是把广告代码修改去掉，使其不能链接到广告源

2.去除广告界面，大部分软件去除了下载源以后还会有个空白的界面留在屏幕上，这时候就要去除了。

常见广告代码（简单却也最常见的去除教程）：

Admob广告代码为：

```
<com.admob.android.ads.AdView
android:id="@+id/ad"
android:layout_width="fill_parent"

android:layout_height="wrap_content"/>
```

Google Ads广告代码为：

```
<com.google.ads.GoogleAdView
android:id="@+id/adview"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
```

Admob广告下载源：

http://r.admob.com/ad_source.php
http://mm.admob.com
http://api.admob.com

广告修改：

修改广告识别ID

```
import android.app.Activity
import net.youmi.android.AdManager;

/**
 *  这是您的应用的主 Activity
 */
public class YourMainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        // 初始化应用的发布 ID 和密钥，以及设置测试模式
        AdManager.getInstance(this).init("您的应用发布ID", "您的应用密钥", false);
    }
}
```

注入恶意代码：

描述：破解者可以通过静态注入的方式向APK中添加恶意代码或广告。

篡改游戏规则：

描述：通过修改反编译获取的smali代码，去除游戏中的种种限制。（如必须下载推广应用或达到某些条件才能继续游戏等）

支付APP安全挑战

黑客会对支付应用做什么？

01 系统使用键盘和输入法攻击

02 界面截取

03 储存本地数据窃取、用户隐私窃取

04 反编译源码

05 网络交互协议抓取

系统使用键盘和输入法攻击：

用户软件界面内部的输入框属于APK，整体大小、颜色都归属于APK内部配置。 但是输入法是属于系统，也可能为第三方输输入法软件。

通过对系统输入法的攻击，达到对支付应用内部输入框数据的窃取。

系统使用键盘和输入法攻击：

易付宝自定义键盘

财付通自定义键盘

拉卡拉自定义键盘

界面截取：

键盘与密码输入框之间的关系：

1）键盘项点击事件影响输入框内容改变

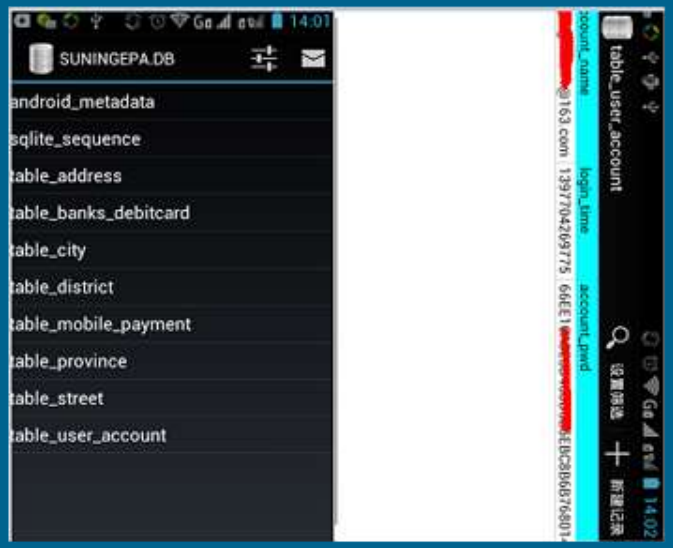
2）密码输入框信息为”*”，但在键盘项事件触发会在界面做项信息提示。(约1s左右)

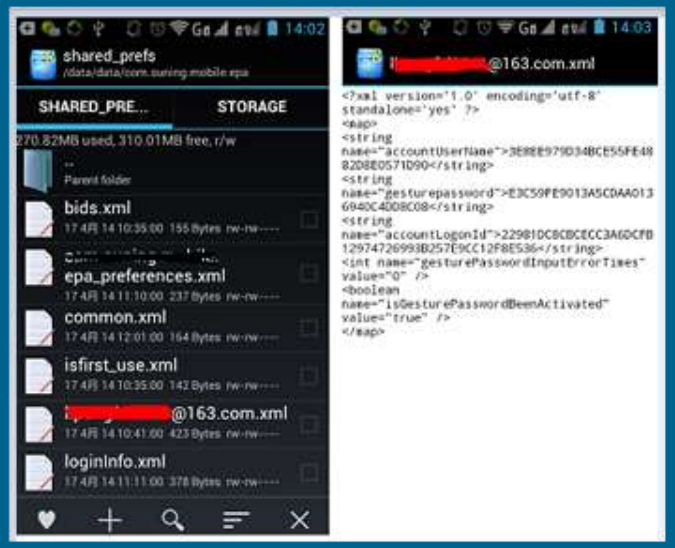
支付APP安全挑战

界面截取：

第三方黑软件可以通过对顶层界面进行实时监控，当界面运行在支付应用安全界面时对系统屏幕事件日志再进行监控，当监控到划屏时对手机界面进行截取，达到如上效果。

储存本地数据窃取、用户隐私窃取：





Root手机获取SharedPreferences和sqlite数据。

反编译源码：

```
E:\aijiami\apktool2.0>apktool.bat d easybuy.apk
I: Using Apktool 2.0.0-Beta7 on easybuy.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\Administrator\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
Cleaning up unclosed ZipFile for archive C:\Users\Administrator\apktool\framework\1.apk
W: Cant find 9patch chunk in file: "drawable-hdpi/navbar.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "drawable-hdpi/btn_sure_normal.9.png". Renaming it to *.png.
W: Cant find 9patch chunk in file: "drawable-hdpi/btn_sure_pressed.9.png". Renaming it to *.png.
I: Decoding values */* XMLs...
I: Loading resource table...
I: Baksmaling...
I: Copying assets and libs...
I: Copying unknown files/dir...
I: Copying original files...
```

```
E:\aijiami\apktool2.0>apktool.bat b easybuy
I: Using Apktool 2.0.0-Beta7 on easybuy
I: Checking whether sources has changed...
I: Smaling...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs...
I: Building apk file...
I: Copying unknown files/dir...
```

通过APKTool.bat指令对apk进行解包、打包

反编译源码—二次打包：

通过对APK进行反编译，获取源码、修改源码、嵌入病毒等行为再打包签名得到的病毒APK与原APK界面一模一样，用户是根本发觉不了。

反编译源码—二次打包：

二次打包的支付应用背后的发展链

反编译源码—查看系统日志：

一、系统日志查看：

Android系统控件和系统方法的使用和操控都会有相应的系统日志提示，黑客可以对系统日志进行观察来进行源码获取源码和源码的研究找到破解突破口。

二、显示自定义日志查看：

Android应用程序开发中开发者往往会对关键功能和数据进行日志输出进行功能测试，在应用正式发布时可能出现疏忽忘记关闭日志导致黑客对定义在日志查询找到破解突破口。

三、隐示自定义日志查看：

大部分有经验的开发者会对自定义日志统一管理，在应用开发时开启日志输出，应用发布时关闭日志输出。但是黑客通过对android应用的二次打包并打开自定义日志的输出再签名运行，此时暴露的日志就非常危险了。

反编译源码—显示自定义日志：

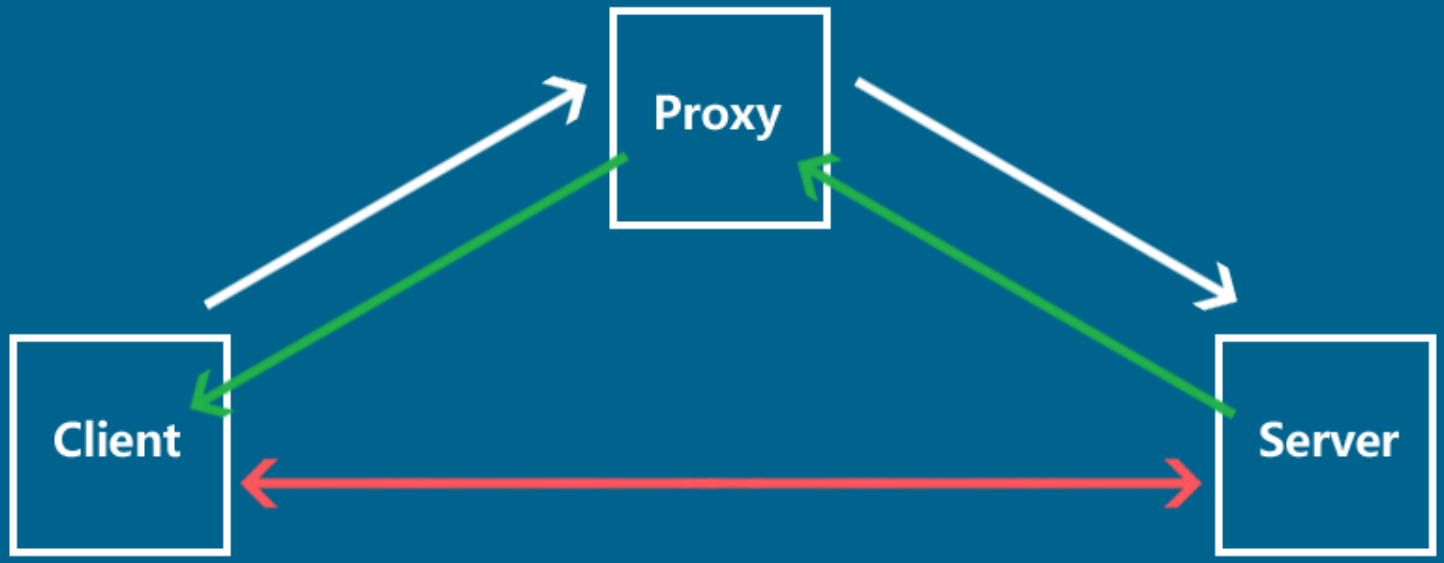
```
public static int UnZipFolder(String paramString1, String paramString2)
{
  ZipInputStream localZipInputStream;
  while (true)
  {
    FileOutputStream localFileOutputStream;
    try
    {
      android.util.Log.v("XZip", "UnZipFolder(String, String)");
      localZipInputStream = new ZipInputStream(new FileInputStream(paramString1));
      ZipEntry localZipEntry = localZipInputStream.getNextEntry();
```

```
    localStatFs = new StatFs(Environment.getExternalStorageDirectory().getPath());
    l1 = localStatFs.getBlockCount();
    l2 = localStatFs.getAvailableBlocks();
  }
  while ((l1 <= 0L) || (l1 - l2 < 0L));
  i = (int)(100L * (l1 - l2) / l1);
  l3 = localStatFs.getBlockSize() * localStatFs.getFreeBlocks();
  Log.d("MicroMsg.Util", "checkSDCardFull per:" + i + " blockCount:" + l1 + " availCount:" + l2 + " availSize:" + l3);
}
while ((95 > i) || (l3 > 52428800L));
```

反编译源码—隐示自定义日志：

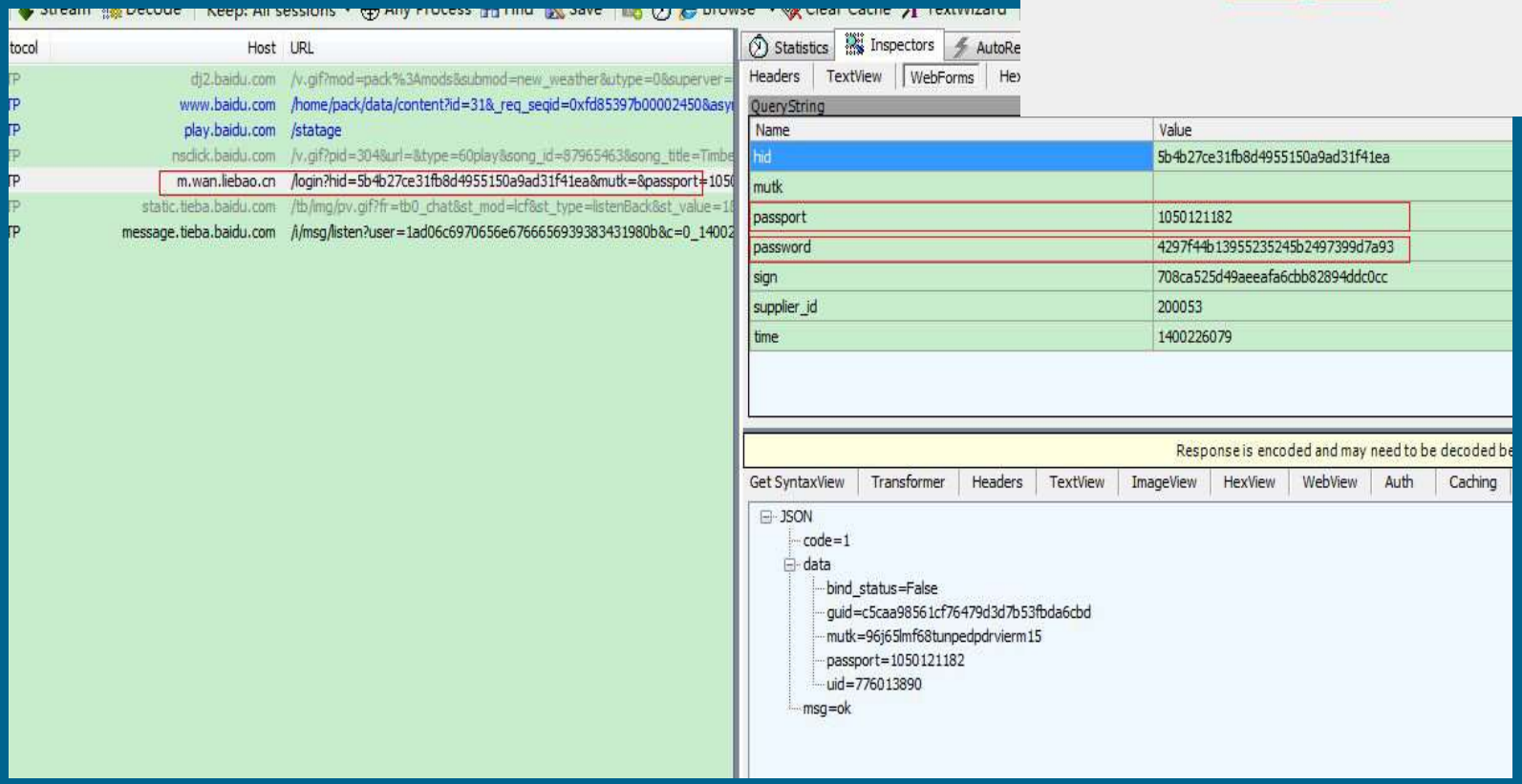
```
package com.joboevan.push.tool;

import android.util.Log;

public final class n
{
  private static boolean a = true;
  private static boolean b = false;

  public static void a(String paramString1, String paramString2)
  {
    if (a)
      Log.w(paramString1, paramString2);
    if (b)
      p.a(paramString2);
  }

  public static void a(boolean paramBoolean)
  {
    a = false;
  }

  public static void b(String paramString1, String paramString2)
  {
    if (a)
      Log.d(paramString1, paramString2);
    if (b)
      p.a(paramString2);
  }

  public static void c(String paramString1, String paramString2)
  {
    if (a.e)
      Log.d(paramString1, paramString2);
  }

  public static void d(String paramString1, String paramString2)
  {
    if (a)
      Log.e(paramString1, paramString2);
    if (b)
      p.a(paramString2);
  }
}
```

网络协议的抓取：



代理抓包：就是将网络传输发送与接收的数据包进行截获、重发、编辑、转存等操作。

网络协议的抓取-抓取案例



APP安全解决方案

三重安全服务

爱加密为APP提供**三重安全服务**：

安全检测服务

为APP提供安全漏洞检测平台。

应用保护服务

保护APP不被反编译和盗版。

渠道监测服务

为APP提供最快的盗版检查服务。

安全检测服务


爱加密APP漏洞分析平台
免费
APK
APP漏洞直接 损害 开发者利益
及时检测 快速补救
马上检测
（目前支持apk文件，软件大小限制为150M内。）
漏洞分析四大特色
文件检查
检查dex、res文件是否存在源代码、资源文件被窃取、替换等安全问题。
漏洞扫描
扫描签名、XML文件是否存在安全漏洞、存在被注入、嵌入代码等风险。
后门检测
检测App是否存在被二次打包，然后植入后门程序或第三方代码等风险。
一键生成
一键生成App关于源码、文件、权限、关键字等方面的安全风险分析报告。

安全检测服务

内嵌sdk检测：			
内嵌广告接入检测	安全	项目说明	漏洞解析
内嵌支付接入检测	安全	项目说明	漏洞解析
内嵌推送接入检测	不安全	项目说明	漏洞解析
内嵌统计接入检测	安全	项目说明	漏洞解析
数据存储检测：			
数据存储检测	不安全	项目说明	漏洞解析
源码检测：			
敏感权限检测	不安全	项目说明	漏洞解析
字符串初始化检测	不安全	项目说明	漏洞解析
资源索引引用检测	不安全	项目说明	漏洞解析
敏感信息检测：			
敏感信息检测	安全	项目说明	漏洞解析
其他项检测：			
So库使用	安全	项目说明	漏洞解析
调试信息	不安全	项目说明	漏洞解析

安全检测服务

APK基本信息

应用名称	好多铃声	版本号	2.4.2
包名	com.haoduolingsheng.RingMore	安全等级	★★☆☆☆

漏洞分析结果报告

检测项		数据存储检测->SharedPreferences
分析方案		对APK进行破解后，获取apk源码进行关键词搜索和人工查询
分析结果		危险
漏洞详情	结果说明	可以获取通过SharedPreferences这种存储方式的存储数据，或者修改获取的默认值。
	敏感代码	包名：cn.jpush.android.c　类名：q.java

代码片段

```
private static SharedPreferences c(Context paramContext)
{
  if (a == null)
    a = paramContext.getSharedPreferences(z[0], 0);
  return a;
}
```

解决方案	**通用标准版：** 爱加密基本版本提供：DEX加密，防二次打包，防止黑客破解等服务，有效保护APK安全。 **企业定制版：** 爱加密定制版本提供：源码混淆，对apk的包名，类名，字符串，资源文件等进行加密。开发者破解后看不到apk的任何信息。还对本地SharedPreferences做有一系列自定义的加密措施。

检测项		数据存储检测->[illegible]
分析方案		对APK进行破解后，获取apk源码进行关键词搜索和人工查询
分析结果		危险
	结果说明	可以获取源码，通过代码逻辑找到创建数据库、创建表，对表的操作以及保存的数据等信息，可以找到私有数据等信息。
		包名：com.haoduolingsheng.RingMore.activity　类名：SelectContactActivity.java

DEX三重保护

So库加密

DEX三重保护：

- **Dex加壳保护**

  技术来源：Android APK包安全级别太低

  缺陷：内存保护级别低

- **Dex指令动态加载**

  技术来源：Dex加壳无法保证源码内存安全

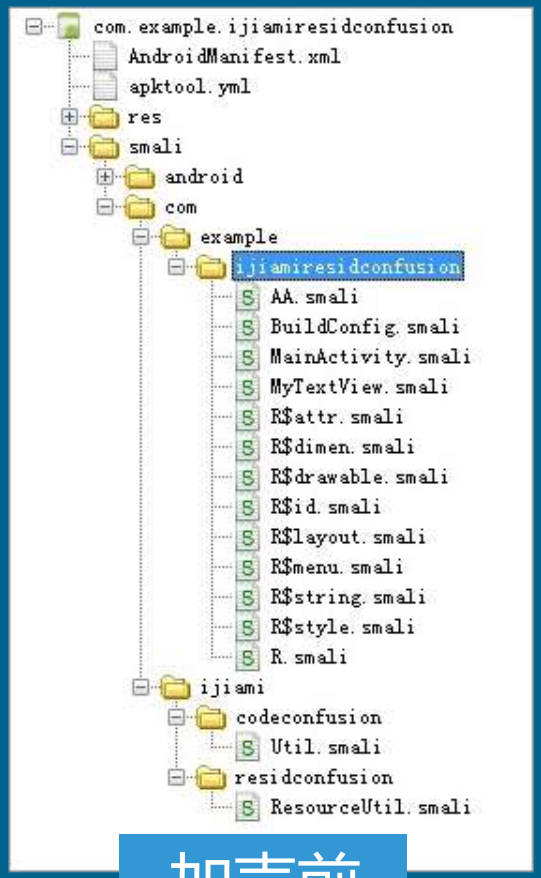
  缺陷：不能100%保证内存安全

- **源码混淆保护**

  技术来源：Dex分离依然无法保证内存中的源码100%安全

DEX三重保护

Dex加壳保护



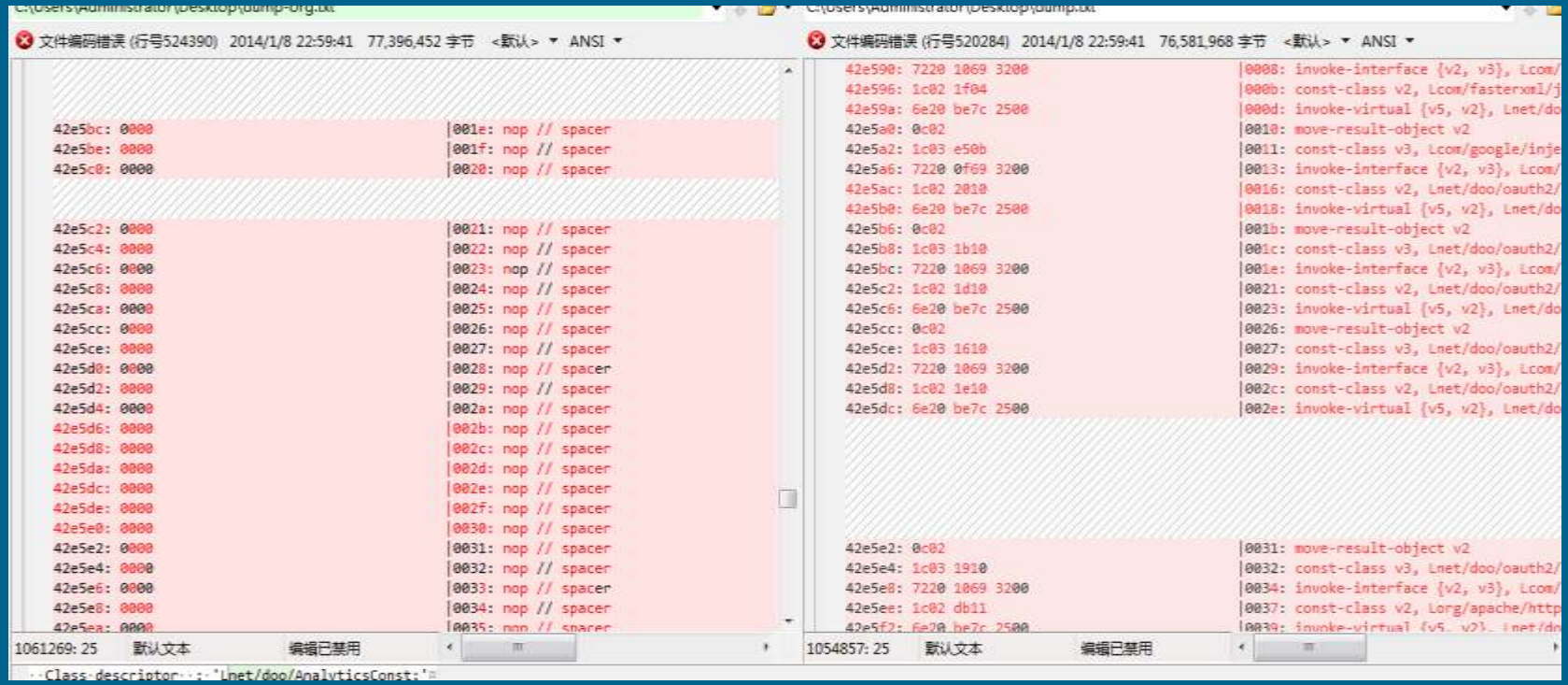
加壳前

VS

加壳后

DEX三重保护

Dex指令动态加载对比图



DEX三重保护

Dex指令动态加载

```
protected ContactPhotoLoader mPhotoLoader = null;
ArrayList<Runnable> n = null;
public boolean noSync = false;
ArrayList<Runnable> o = null;
ArrayList<Runnable> p = null;
private OnKeyboardListener q = null;
private ViewTreeObserver.OnGlobalLayoutListener r = new g(this);

public void addKeyboardHideRunnable(boolean paramBoolean, Runnable paramRunnable)
{
}

public void addKeyboardShowRunnable(boolean paramBoolean, Runnable paramRunnable)
{
}

public void dismissProgressDialog()
{
}

public void finish()
{
}

public ContactPhotoLoader getContactPhotoLoader()
{
  return null;
}

protected int getThemeResId()
{
  return 0;
}

protected void initXlip(Bundle paramBundle)
{
}
```

```
    return;
  }
}

public void addKeyboardShowRunnable(boolean paramBoolean, Runnable paramRunnable)
{
  if (paramBoolean);
  for (ArrayList localArrayList = this.keyboardShowRunnables; ; localArrayList = this.keyboardShowRunnablesPermanent)
  {
    localArrayList.add(paramRunnable);
    return;
  }
}

public void dismissProgressDialog()
{
  AsyncLog.v(getClass().getSimpleName(), "ProgressDialog> [" + hashCode() + "] dismissProgressDialog");
  if (this.isDialogShown)
  try
  {
    dismissDialog(0);
    label49: this.isDialogShown = false;
    return;
  }
  catch (IllegalArgumentException localIllegalArgumentException)
  {
    break label49;
  }
}

public void finish()
{
  Main.goingBack = true;
  super.finish();
}

public ContactPhotoLoader getContactPhotoLoader()
{
  if (this.mPhotoLoader == null)
    this.mPhotoLoader = new ContactPhotoLoader(this, 2130837659);
  return this.mPhotoLoader;
}
```

Dex高级混淆加密：

01 Android常用控件对象名进行混淆

02 Android常用方法名进行混淆

03 对java与Android部分常用对象名进行混淆

04 对源码中出现的字符串进行加密

05 对源码与资源文件ID的引用进行加密

Dex高级混淆加密：

```
private ImageView iv;

private void clear()
{
  this.fadein = null;
  this.fadeout = null;
  this.iv = null;
}

private void init()
{
  this.iv = ((ImageView)findViewById(2131427362));
  this.fadein = AnimationUtils.loadAnimation(this, 2130968576);
  this.fadeout = AnimationUtils.loadAnimation(this, 2130968577);
  new Thread(this).start();
}

protected void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  requestWindowFeature(1);
  setContentView(2130903056);
  init();
}
```

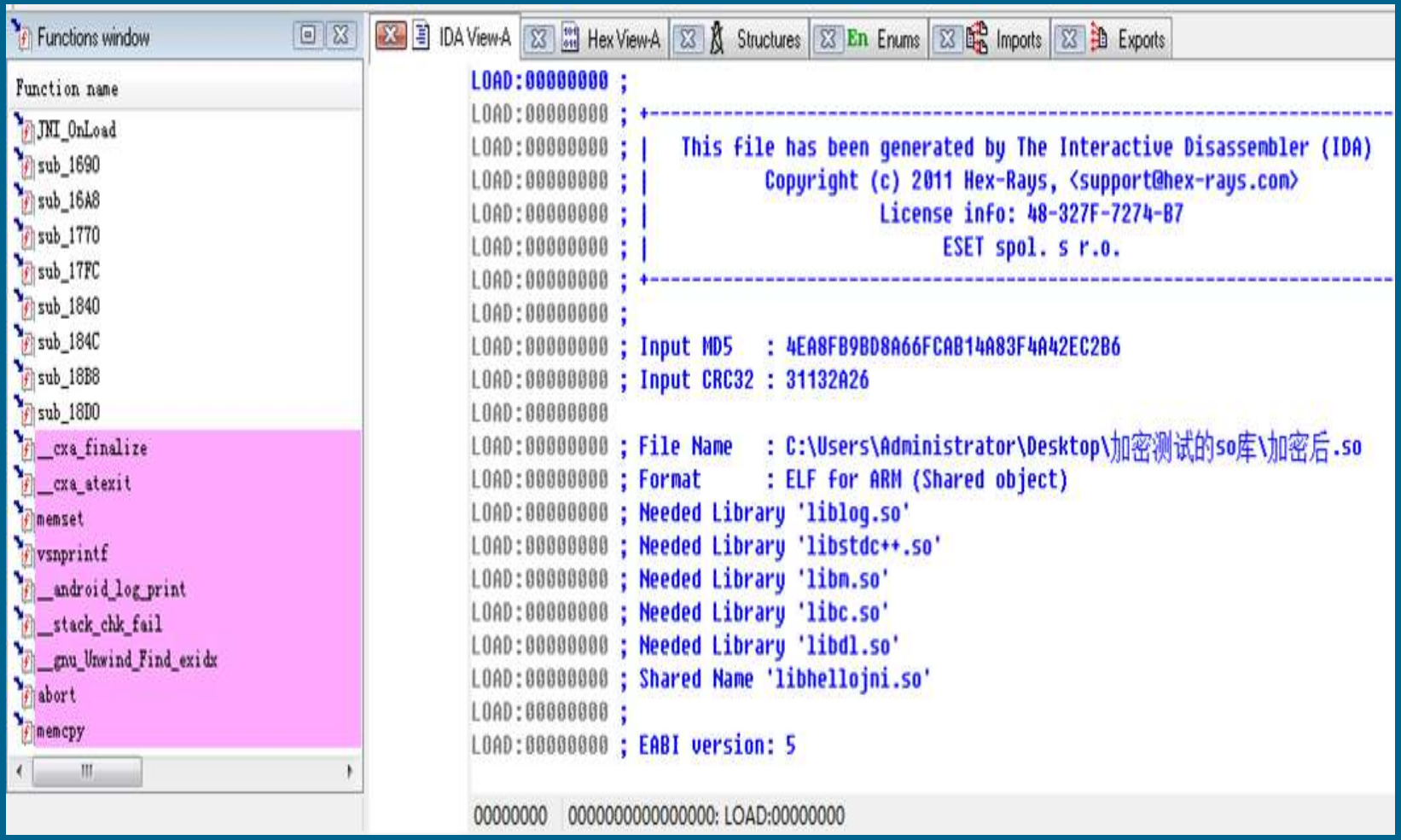
加密前

```
private IJM_B ijma;

private void clear()
{
  this.fadein = null;
  this.fadeout = null;
  this.ijma = null;
}

private void init()
{
  this.ijma = ((IJM_B)findViewById(ResourceUtil.getId(this, "fNULXVNtRVPERNf")));
  this.fadein = AnimationUtils.loadAnimation(this, 2130968576);
  this.fadeout = AnimationUtils.loadAnimation(this, ResourceUtil.getAnimId(this, "OJMNXdc"));
  new Thread(this).start();
}

protected void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  requestWindowFeature(1);
  setContentView(ResourceUtil.getLayoutId(this, "fNULXVN"));
  init();
}
```

加密后

SO库加密

ida反编译加密后的so库



渠道监测

- 精准识别正盗版
- 一站监测渠道信息
- 详情分析盗版APP
- 24小时即时更新渠道数据

渠道监测

数据挖掘：

渠道数据24小时及时更新（包括渠道名、版本信息、版本个数、下载次数、盗版详情等）开发者第一时间通过更新数据对正盗版APP做出正确评估及处理。

深度分析：

从系统和人工两方面对盗版APP做出详情分析，包名、版本、渠道、下载次数以及正盗版对比一目了然，及时进行安全保护。

全面覆盖：

精准获取国内外428个APP推广渠道信息，包括应用商店、下载站、论坛等，快速识别分布在各大渠道的正盗版APP。

渠道监测

爱加密为“史上最坑爹游戏”提供渠道监测服务

渠道监测

爱加密为“史上最坑爹游戏”提供渠道监测服务

监测记录

应用名称	史上最坑爹的游戏	证书	None
包名	com.ipeaksoft.pitDadGame		

监测渠道 | 渠道全部版本信息 | 渠道盗版版本 | 版本的正盗版个数 | 正版信息详情 | 盗版信息详情

总计apk：39

应用名称	包名	版本	渠道	下载次数	下载地址
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	1mobile	92	http://t.1mobile.com/mobile_software/brain/com.ipeaksoft.pitDadGame_1006.apk
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	叶子猪	50000	http://app.yzz.cn/api/download?id=86934&site=9&an&imei=&mac=&token=5e2f765b452f67e859e760d89027ed9f&sign=71c318852523b74.
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	叶子猪	50000	http://app.yzz.cn/api/download?id=87945&site=9&an&imei=&mac=&token=ac5e350383a878d0e2a0056c170f596a&sign=b488f4d03c8dc2a.
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	第风网	4245	http://an.sfeny.com/download.php?id=405052
史上最坑爹的游戏1.0.06安卓版	com.ipeaksoft.	1.0.06	QQ下载网	–	http://www.qqdn.com/down.asp?id=50409
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	TopBer安卓市场	4256	http://www.topber.com/download.php?id=405052
史上最坑爹的游戏安卓版1.0.06下载	com.ipeaksoft.	1.0.06	天畅游戏	106	http://down.tcgame.com.cn/tcgame01/an/sszkdxy_1.0.06_www.tcgame.com.cn.apk
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	斗蟹手游	81	http://shouji.douxie.com/app/download/9048
史上最坑爹的游戏(1.0.06)	com.ipeaksoft.	1.0.06	安卓应用部落	4033	http://www.app30.net/download.php?app_id=75649&id=405052
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	喜爱APP	4247	http://www.xaapp.com/download.php?id=405052
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	啦啦网	4245	http://www.niapk.com/download.php?app_id=78093&id=405052
史上最坑爹的游戏	com.ipeaksoft.	1.0.06	yeah就	4249	http://apk.yeah2.com/download.php?id=405052

« Prev 1 / 4 Next »

渠道监测

爱加密为“史上最坑爹游戏”提供渠道监测服务

渠道监测

爱加密为“史上最坑爹游戏”提供渠道监测服务

【史上最坑爹的游戏】盗版分析结果

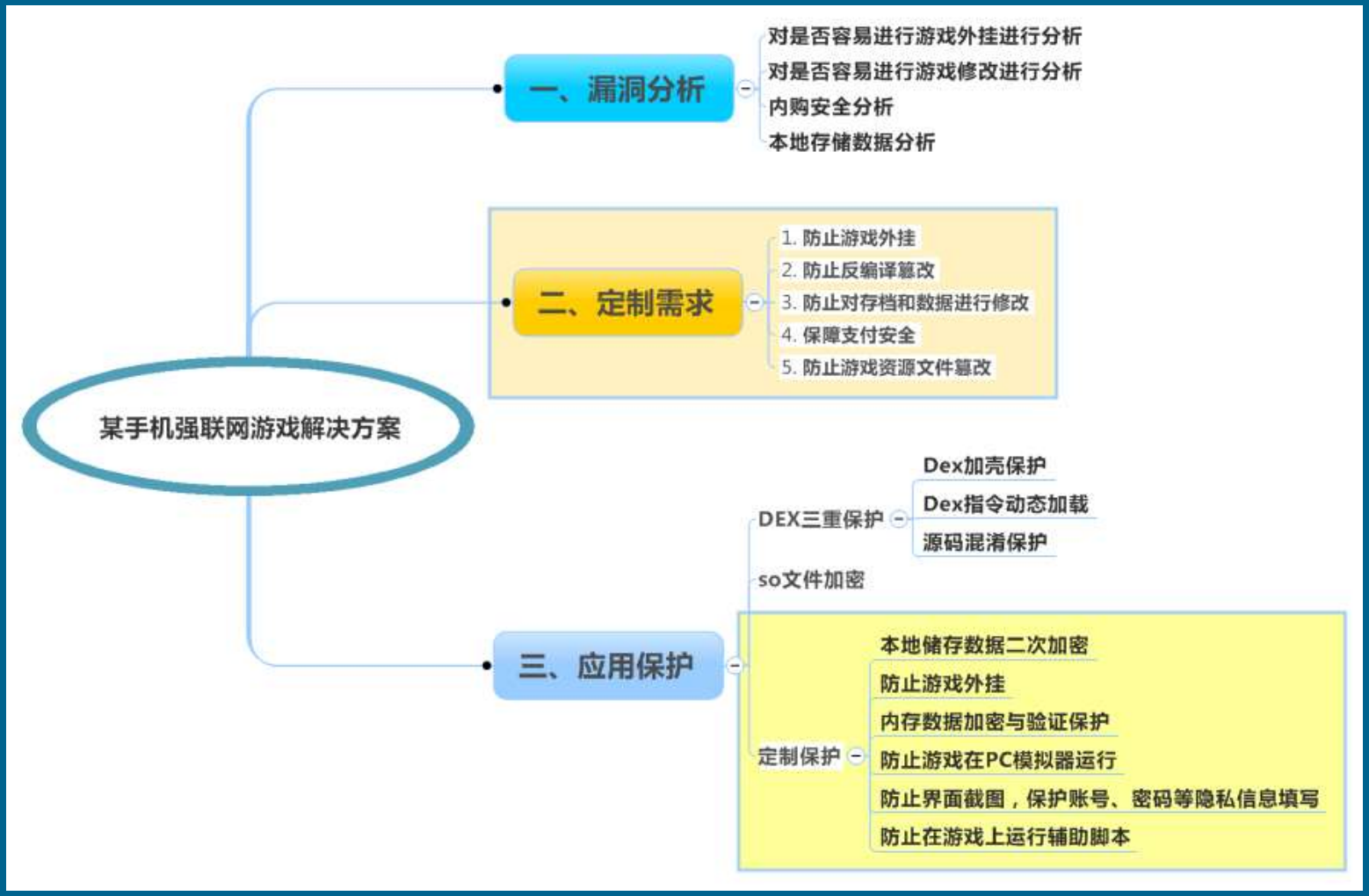
APK的基本信息对比		
	正版	盗版
包名	com.ipeaksoft.pitDadGame	com.ipeaksoft.pitDadGame.mivefo
版本信息	1.0.06	1.0.06
版本号	1006	1006
Application		
签名md5值	3EB744BA52E0CAD126E2E941B8538EB2 正版	F7A3EA7F2EA6F6880CC7A2DD6678116B

盗版APK详细信息		
Activity	cn.domob.android.ads.DomobActivity	删除
	com.baidu.mobads.AppActivity	删除
	com.ipeaksoft.pitDadGame.mi.tfyoff.x	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.b	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.f	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.aa	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.j	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.p	增加
	com.ipeaksoft.pitDadGame.mi.tfyoff.t	增加

服务案例

一款日系动作类游戏，游戏定位为玩家提供填补碎片时间的爽快动作射击体验

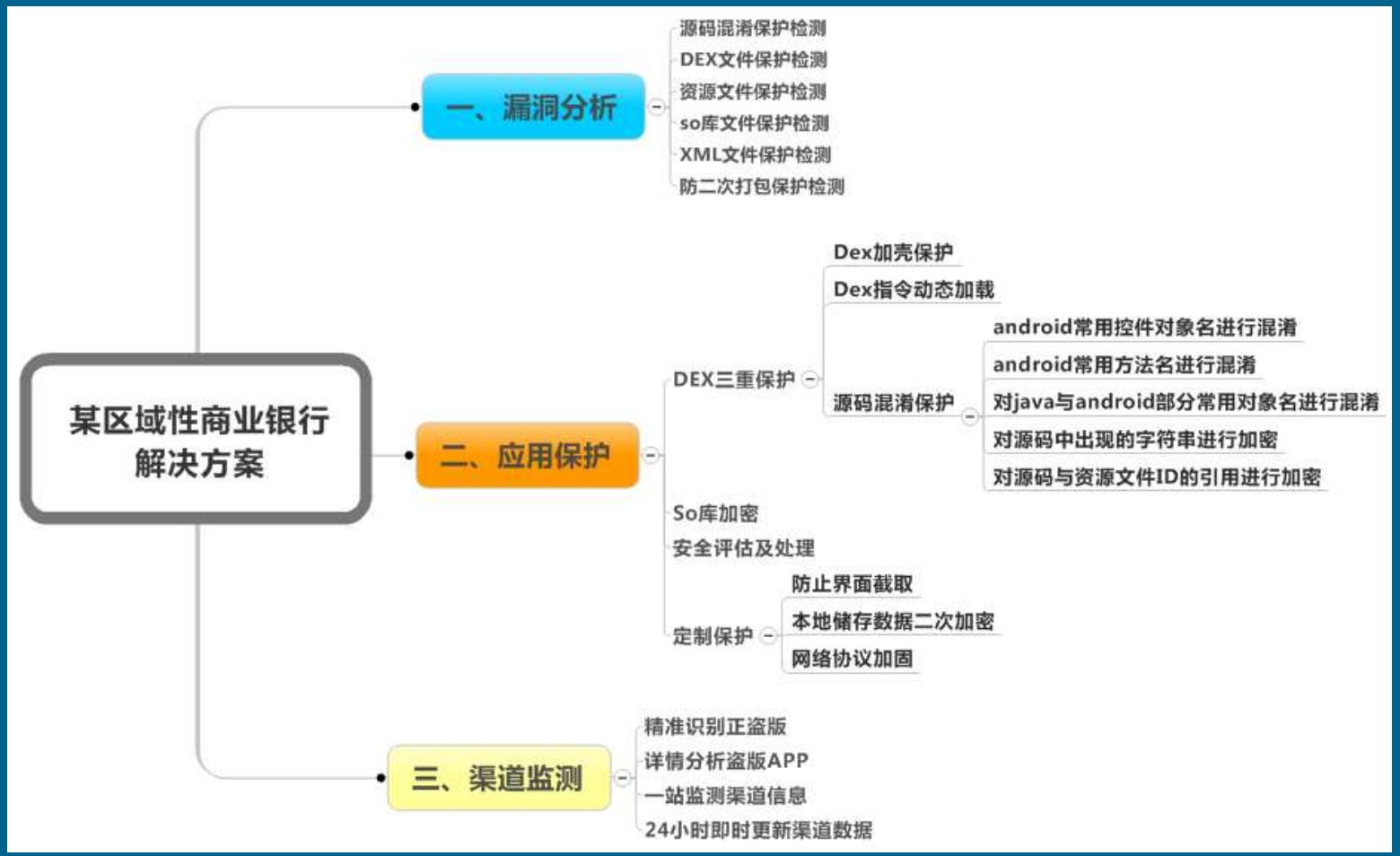
使用服务：
漏洞分析+定制保护+so文件加密



服务案例

该省规模最大、实力最强的一家本土银行机构，资产超过2000亿。

使用服务：
漏洞分析+应用保护+渠道监测



服务案例


首信易支付
PayEase


中国电信
CHINA TELECOM


郑州银行
BANK OF ZHENGZHOU


厨


崩坏学园


坑2


中国平安


平安银行
PING AN BANK


Suning.com
苏宁易购


BUS

THANKS

谢谢观看