个人资料

Jack_Jia

Android监控程序本身被卸载方法汇总

标签： Android应用 卸载提示 Android卸载提示

2013-09-03 16:55 7845人阅读 评论(8) 收藏 举报

分类：

Android开发调试（9）

本文章由Jack_Jia编写，转载请注明出处。

文章链接：http://blog.csdn.net/jiazhijun/article/details/10157901

作者：Jack_Jia 邮箱： 309zhijun@163.com

一般开发者都有这样的业务需求：统计自己应用的卸载量或在用户卸载应用后提供反馈信息以便更好的改进软件。

积分： 7428

等级： BLOG 5

排名： 第2011名

原创： 46篇 转载： 50篇

译文： 3篇 评论： 556条

应用开发者可以通过注册“Android.intent.action.PACKAGE_REMOVED”广播获取卸载其它应用的信息，但该广播不能用于应用本身被卸载。如何获取自己被卸载的信息呢？

目前有两种方式可以做到应用卸载提示：

第一种通过监控Android日志实现：

启动一个服务监控android系统的打印日志，当监控到"android.intent.action.DELETE"并且包含自己应用的包名时，提示给用户。代码摘至 （http://blog.csdn.net/xyz_lmn/article/details/8330710）

缺点：耗电问题，且程序必须在启动的情况下才可以监控。

[java]

```
01. public class AndroidLogcatScannerThread extends Thread {
02.     private LogcatObserver observer;
03.     public AndroidLogcatScannerThread(LogcatObserver observer) {
04.             this.observer = observer;
05.     }
06. 
07.     public void run() {
08.             String[] cmds = { "logcat", "-c" };
09.             String shellCmd = "logcat";
10.             Process process = null;
11.             InputStream is = null;
12.             DataInputStream dis = null;
13.             String line = "";
14.             Runtime runtime = Runtime.getRuntime();
15.             try {
16.                     observer.handleLog(line);
17.                     int waitValue;
18.                     waitValue = runtime.exec(cmds).waitFor();
19.                     observer.handleLog("waitValue=" + waitValue + "\n Has do Clear logcat
20.                     process = runtime.exec(shellCmd);
21.                     is = process.getInputStream();
22.                     dis = new DataInputStream(is);
23.                     while ((line = dis.readLine()) != null) {
```

文章存档

展开

阅读排行


```
24.                         //Log.d("Log","Log.Bestpay:"+line);
25.
26.                         if(observer!=null)
27.                             observer.handleLog(line);
28.
29.                 }
30.             } catch (InterruptedException e) {
31.                 e.printStackTrace();
32.             } catch (IOException ie) {
33.                 ie.printStackTrace();
34.             } finally {
35.                 try {
36.                     if (dis != null) {
37.                         dis.close();
38.                     }
39.                     if (is != null) {
40.                         is.close();
41.                     }
42.                     if (process != null) {
43.                         process.destroy();
44.                     }
45.                 } catch (Exception e) {
46.                     e.printStackTrace();
47.                 }
48.             }
49.     }
50. }
```

监控服务：

```
[java]
01. public class AndroidLogcatScannerService extends Service implements LogcatObserver{
02.
03.     @Override
04.     public void onCreate() {
```

【Android病毒分析报告】
(14124)
Android DEX安全攻防战
(13818)

最新评论

Android APK加壳技术方案【1】
snzang: 1、加壳程序：加密源程序为解壳数据、组装解壳程序和解壳数据 2、解壳程序：解密解...

Android APK加壳技术方案【1】
GeekKevin: 请问各位 另一种比较复杂的加壳方式 有谁具体实践吗能成功吗？

Android APK加壳技术方案【2】
wangzaieee: @beyond296089727:傻逼，想钱想疯了吧。

Android APK加壳技术方案【2】
往事随风慢慢飘散: 有什么用呢，完全没有实用性

Android APK加壳技术方案【2】
往事随风慢慢飘散: 加解密算法还是得自己写

【Android开发技巧】- 如何获取
Mirhunana: 有用

Android APK加壳技术方案【1】
iewwc: 为何一会儿说apk加壳一会儿说dex加壳，不能只对dex加壳吗？

Android APK加壳技术方案【2】
KingCallMe: 哈哈首先多谢分享，不过这种技术已经没有使用价值啦，只能对简单的demo进行加壳，一旦稍微复杂一点点的...

Android安全分析挑战：运行时篡
KingCallMe: @difcareer:用了伪加密但是去掉加密后，还是不能安装啊，提示Failure

```
05.         // TODO Auto-generated method stub
06.         super.onCreate();
07.     }
08.
09.     @Override
10.     public void onDestroy() {
11.         // TODO Auto-generated method stub
12.         super.onDestroy();
13.     }
14.
15.     @Override
16.     public void onStart(Intent intent, int startId) {
17.         // TODO Auto-generated method stub
18.         super.onStart(intent, startId);
19.
20.         AndroidLogcatScannerThread scannerThread=new AndroidLogcatScannerThread(AndroidLo
21.         scannerThread.start();
22.     }
23.
24.     @Override
25.     public IBinder onBind(Intent intent) {
26.         // TODO Auto-generated method stub
27.         return null;
28.     }
29.
30.     @Override
31.     public void handleLog(String info) {
32.         // TODO Auto-generated method stub
33.         if (info.contains("android.intent.action.DELETE") && info.contains(getPackageName
34.
35.                 Intent intent = new Intent();
36.                 intent.setClass(AndroidLogcatScannerService.this, UninstallActivity.class
37.                 intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
38.                 startActivity(intent);
39.         }
40.     }
41.
```

```
42.  }
```

第二种通过Fork子进程，然后在子进程中通过监控/data/data/{package_name}目录变化实现应用卸载提示，当私有目录被删除时，本应用即被卸载。”豌豆荚“即时使用该中方式。下面我们通过逆向分析“豌豆荚”来看看该功能具体实现。

豌豆荚被卸载后，总是调用浏览器打开如下页面：

在逆向过程中发现，即使在设置->应用程序中强制停止豌豆荚相关应用，当卸载豌豆荚是仍然可以调起浏览器访问反馈页面。豌豆荚是如何做到的呢？

豌豆荚启动后，主要涉及以下三个进程：

```
app_4     15503 12598 516780 53420 ffffffff 400e4440 S com.wandoujia.phoenix2
app_4     15575 15503 732    228   c0249128 400104d8 S uuids_sys
app_44    15610 12598 466736 27924 ffffffff 400e4440 S com.wandoujia.phoenix2.usbproxy
```

当在设置->应用程序中强制关闭豌豆荚相关应用后，我们观察进程变化：

```
app_4     15575 15503 732    228    c0249128 400104d8 S uuids_sys
```

发现在设置->应用程序中强制关闭并不能关闭uuid_sys进程，那么卸载反馈逻辑肯定就在uuid_sys进程中完成了！

在豌豆荚APK安装包存在lib\armeabi\libuuid.so文件，该文件并不是共享库文件而是一个可执行文件。

使用IDA逆向libuuid.so文件，实现关键代码如下：

1、使用Linux inotify_init、inotify_add_watch监控文件系统目录/data/data/com.wandoujia.phoenix2的变化。

```
.text:00008B24 loc_8B24                                ; CODE XREF: sub_8AA4+74↑j
.text:00008B24                                         ; sub_8AA4+78↑j
.text:00008B24                 LDMIA   R4!, {R0}
.text:00008B26                 CMP     R0, #0
.text:00008B28                 BNE     loc_8B1A
.text:00008B2A                 BLX     inotify_init
.text:00008B2E                 LDR     R1, =(aDataDataCom_wa - 0x8B38)
.text:00008B30                 MOVS    R2, 0x200
.text:00008B34                 ADD     R1, PC          ; "/data/data/com.wandoujia.phoenix2"
.text:00008B36                 STR     R0, [SP,#0x8A08+var_8A00]
.text:00008B38                 BLX     inotify_add_watch
.text:00008B3C                 STR     R0, [SP,#0x8A08+var_89FC]
.text:00008B3E
```


2、当监控目录被删除时说明程序被卸载，通过am命令启动浏览器打开指定网页。

```
.text:00008ACC                LDR     R1, =(aAmStartUser0AA - 0x8AD6)
.text:00008ACE                ADD     R0, SP, #0x8A08+var_87EC ; char *
.text:00008AD0                LDR     R7, =0x604
.text:00008AD2                ADD     R1, PC          ; "am start --user 0 -a android.intent.act"...
.text:00008AD4                BLX     strcpy
.text:00008AD8                LDR     R4, [R5,#4]
.text:00008ADA                ADD     R0, SP, #0x8A08+var_87EC ; char *
.text:00008ADC                ADD     R7, SP
.text:00008ADE                MOVS    R1, R4          ; char *
.text:00008AE0                BLX     strcat
.text:00008AE4                LDR     R1, =(aAmStartAAndroi - 0x8AEE)
.text:00008AE6                LDR     R0, =0x604
.text:00008AE8                ADD     R6, SP, #0x8A08+var_87EC
.text:00008AEA                ADD     R1, PC          ; "am start -a android.intent.action.VIEW "...
.text:00008AEC                ADD     R0, SP          ; char *
.text:00008AEE                BLX     strcpy
.text:00008AF2                LDR     R0, =0x604
.text:00008AF4                MOVS    R1, R4          ; char *
.text:00008AF6                ADD     R0, SP          ; char *
.text:00008AF8                BLX     strcat
.text:00008AFC                B       loc_8B06
```


上一篇 Android第二个绕过签名认证漏洞原理

下一篇 Android WebView挂马漏洞--各大厂商纷纷落马

我的同类文章

Android开发调试（9）

- Android Accessibility(辅助... 2014-12-12 阅读 7358
- 【Android开发技巧】- 如... 2013-05-03 阅读 4338
- Android优秀开源项目大全 2013-04-18 阅读 14543
- Android软件安全开发实践 2013-03-12 阅读 3952
- 捕获Android运行时改变 2013-03-05 阅读 1725
- Android4.0内存Dex数据动... 2013-08-01 阅读 16338
- GDB+gdbserver 远程调试... 2013-04-27 阅读 5459
- Android对system_server... 2013-04-16 阅读 3385
- Android Bander设计与实现... 2013-03-05 阅读 1444

参考知识库

.NET知识库

780 关注 | 635 收录

Linux知识库

4001 关注 | 3050 收录

Android知识库

19275 关注 | 1644 收录

猜你在找

反编译Android应用

Android应用的调试

Android应用更新实现策略

Android应用开发流程及友盟统计集成

android 卸载程序清除数据停止服务使用方法

Android模拟器中APK文件的安装和卸载方法

android监听自身被卸载的方法

Android安装卸载程序具体操作方法解析

零基础玩转Linux+Ubuntu(嵌入式开发基础课程)　　Android 启动 卸载 安装 apk 的方法

查看评论

6楼 rooneyGG 2016-03-09 18:29发表

这个方案不可能在所有机器或者场景都运行，原因：.必须启动豌豆荚间接启动监控进程，安装后立刻卸载这种情况下就无法打开反馈页面。 另外，在5.0机器上，使用FileObserve监控trace.txt文件好像并不成功，而FileObserve也是包装的inotify，不知楼主遇到过这个问题没有？ 能否解答一下原因，感激不尽。

5楼 余龙飞 2014-06-22 19:50发表

豌豆荚监控进程的uid和主进程的uid一样，强行停止进程会调用forcestoppackage函数，这个不会把2个进程都杀死吗？

4楼 皮鲁 2014-02-12 13:48发表

用daemon进程也可以，更安全

3楼 hackill2 2014-02-10 17:43发表

哥们，怎么查看一个程序的相关进程呢？你用的是什么工具？还请明示。谢谢

2楼 djyy1987 2013-09-26 16:13发表

第二个果然啊
不过感觉豌豆荚有bug，libuuid.so有时不启动，测试了半天。。。

Re: rooneyGG 2016-03-09 18:25发表

这个方案不可能在所有机器或者场景都运行，原因：.必须启动豌豆荚间接启动监控进程，安装后立刻卸载这种情况下就无法打开反馈页面。 另外，在5.0机器上，使用FileObserve监控trace.txt文件好像并不成功，而FileObserve也是包装的inotify，不知楼主遇到过这个问题没有？ 能否解答一下原因，感激不尽。

1楼 tom540066931 2013-09-05 09:07发表

求第二种的源码，楼主能发到我邮箱吗？540066931@qq.com谢了

发表评论

用 户 名： u014231159

评论内容：

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide
Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase
Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap