# **一、文件读写**

**1、demo：**两个按钮分别通过JAVA层和C层对一个txt文件进行读写操作。

**2、关键库：**

（JAVA层）

java.io.FileInputStream

java.io.FileOutputStream

（C层）

<fcntl.h>

**3、hook记录**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **open**(const char \*pathname, int oflag,int perms) | **read**(int fd,void \* buf ,size\_t count) | **write**(int fd, const void \*buf, size\_t count) |
| **libjavacore.so** | 有输出。  path输出.txt文件的绝对路径。  后两个参数输出不变化。 | 有输出。  %s buf显示读取内容，%d buf随着点击持续增加。  count显示的是读取长度。 | 有输出。  %s buf为内容，%d buf随着点击持续增加。  count为内容长度。 |
| **libcutils.so** | 无输出。 | 无输出。 | 无输出。 |
| **libutils.so** | 无输出。 | 有输出，输出不改变。  fd=22.  %s buf输出为乱码，%d buf=-1092987580，都不变。  count=16。 | 有输出，输出不改变。  %s buf=W，%d buf=1074556549，都不变。  count=1。 |
| **libsqlite.so** | 无输出。 | 无输出。 | 无输出。 |
| **libandroid\_runtime.so** | 无输出。 | 无输出。 | 无输出。 |

**libbinder.so**

**ioctl**(int \_\_fd, unsigned long int \_\_request, void \* arg)

有输出。

调用次数都为3的倍数，

fd=9，

request=-1072143871，

arg变化。

**结论：**通过对libjavacore.so的open函数拦截可以观察到应用程序的操作对象，操作内容与内容长度能够完全解读。

# **二、Shared Preferences(.xml)**

**1、demo：**自己随便在输入框中输入数据，点击button会存入数据库并读取数据库中的数显示到界面。

**2、关键库：**

android.content.SharedPreferences

android.content.SharedPreferences.Editor

（追踪源码实际上用到的库依然是java.io.(File)InputStream）

**3、hook记录**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **open**(const char \*pathname, int oflag,int perms) | **read**(int fd,void \* buf ,size\_t count) | **write**(int fd, const void \*buf, size\_t count) |
| **libjavacore.so** | 有输出。  path输出.xml文件的绝对路径。  后两个参数输出不变化。  oflag=577  perms=384 | 无输出。 | 有输出。  fd=41  %s buf=.xml文件内容  count=10x，x=写入数的长度。 |
| **libcutils.so** | 无输出。 | 无输出。 | 只输入框中的内容改变的时候有输出，一次输出6条。  fd=23  %s buf=7292或7291  count=4 |
| **libutils.so** | 无输出。 | 有持续输出，输出不改变。  fd=22.  %s buf输出为乱码，  count=16。 | 有持续输出，输出不改变。  fd=29  %s buf=W，  count=1。 |
| **libsqlite.so** | 无输出。 | 无输出。 | 无输出。 |
| **libandroid\_runtime.so** | 无输出。 | 无输出。 | 无输出。 |

**libbinder.so**

**ioctl**(int \_\_fd, unsigned long int \_\_request, void \* arg)

有输出。

不随用户行为每秒持续输出，

fd=9，

request=-1072143871（同文件读写），

arg变化，但每秒变化的数相同。%s arg=，， 4， （乱码）。

**结论：**通过对libjavacore.so的open函数拦截可以观察到应用程序的操作对象，通过对libjavacore.so的write函数拦截可以观察到xml文件的操作内容。

# **三、Content Provider(Contacts,SMS)**

**1、demo：**三个按钮，分别获得联系人、短信收件箱和通话记录，并以Log显示。

**2、关键库：**

android.database.Cursor

android.content.ContentResolver

**3、hook记录**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **open**(const char \*pathname, int oflag,int perms) | **read**(int fd,void \* buf ,size\_t count) | **write**(int fd, const void \*buf, size\_t count) |
| **libjavacore.so** | 无输出。 | 无输出。 | 无输出。 |
| **libcutils.so** | 无输出。 | 无输出。 | 只有Contacts有输出。输出一次10条相同的  fd=23  %s buf=29536  count=5 |
| **libutils.so** | 无输出。 | 在Log输出前和后都有输出。  fd=22  %s buf=乱码，不一样  count=16 | 在Log输出前和后都有输出。  fd=29  %s buf=W  count=1 |
| **libsqlite.so** | 无输出。 | 无输出。 | 无输出。 |
| **libandroid\_runtime.so** | 无输出。 | 无输出。 | 无输出。 |

**libbinder.so**

**ioctl**(int \_\_fd, unsigned long int \_\_request, void \* arg)

有输出。

fd=9，

request=-1072143871（同文件读写），

arg变化。%s arg=，， 4 ，（乱码）。

**结论：**ioctl函数称为最有可能的捕捉对象，行为捕捉不明确。

# **四、SQLite**

**1、demo：**自建数据库，可以对数据库进行增删改查。

**2、关键库：**

android.database.Cursor

android.database.SQLiteDatabase

android.database.SQLiteDatabase.CursorFactory

android.database.SQLiteOpenHelper

**3、hook记录**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **open**(const char \*pathname, int oflag,int perms) | **read**(int fd,void \* buf ,size\_t count) | **write**(int fd, const void \*buf, size\_t count) |
| **libjavacore.so** | 无输出。 | 无输出。 | 无输出。 |
| **libcutils.so** | 每次操作有一条相同输出。  pathname=/dev/ashmem  oflag=2  perms=98 | 无输出。 | 插入、更新、删除后均有两条相同输出。输入框内容一旦改变立即有输出。  fd=23  %s buf=2175和2176交替出现  count=4 |
| **libutils.so** | 无输出。 | 有持续输出，输入框内容改变和按钮动作都会增加输出量。  fd=22  %s buf=乱码，会变  count=16 | 有持续输出，输入框内容改变和按钮动作都会增加输出量。但输出全部一样。  fd=29  %s buf=W  count=1 |
| **libsqlite.so**  **（依赖于libutils.so）** | 均有输出。输出量与操作有关。  pathname=.db和.db-journal和database文件夹绝对路径  oflage=131072（.db-journal），131138（.db），0（文件夹）  perms=384,432 | 无输出。 | 无输出。 |
| **libandroid\_runtime.so** | 无输出。 | 无输出。 | 无输出。 |

**libbinder.so**

**ioctl**(int \_\_fd, unsigned long int \_\_request, void \* arg)

有输出。

持续每秒输出，按钮时输出增加。

fd=9，

request=-1072143871（同文件读写），

arg变化。%s arg=，， 4 ，（乱码）。

**结论：**通过对libsqlite.so的open函数拦截可以观察到操作对象，但内容无法捕捉。