Android 动态逆向分析工具（二）

——Andbug 扩展功能

anbingchun@163.com

本文所述功能是在 andbug 原有功能的基础上实现的信息的功能，包括对 apk 运行时行为监控功能的实现，以及获取 apk 更详细信息的功能的实现。

Andbug 作为一个用于动态逆向分析 apk 的工具来说目前只实现了相对简单的功能，还不能算是一个完整功能的调试器。好在原作者将工具的所有代码都开源了，我们可以根据自己的实际需求增加信息功能，本文作者主要增加了对 apk 运行时的行为进行动态监控的功能，详细信息位于：https://github.com/anbc/AndBug/。还有很多功能需要我们进一步实现，本文后面部分也会列出一些，文本作者认为需要增加的功能点。

一、扩展功能实现

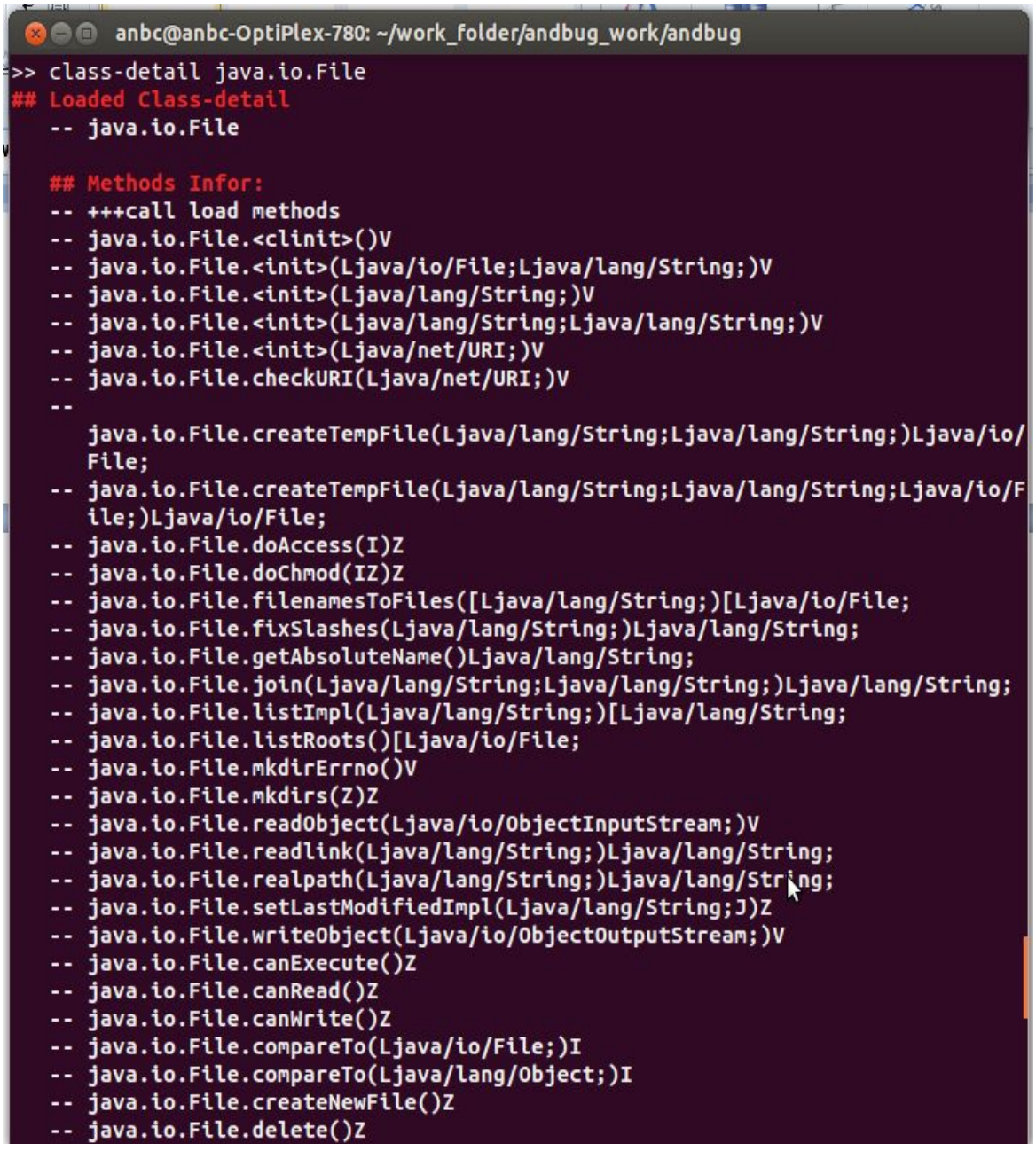
1、davlik 虚拟的对 java 的支持情况

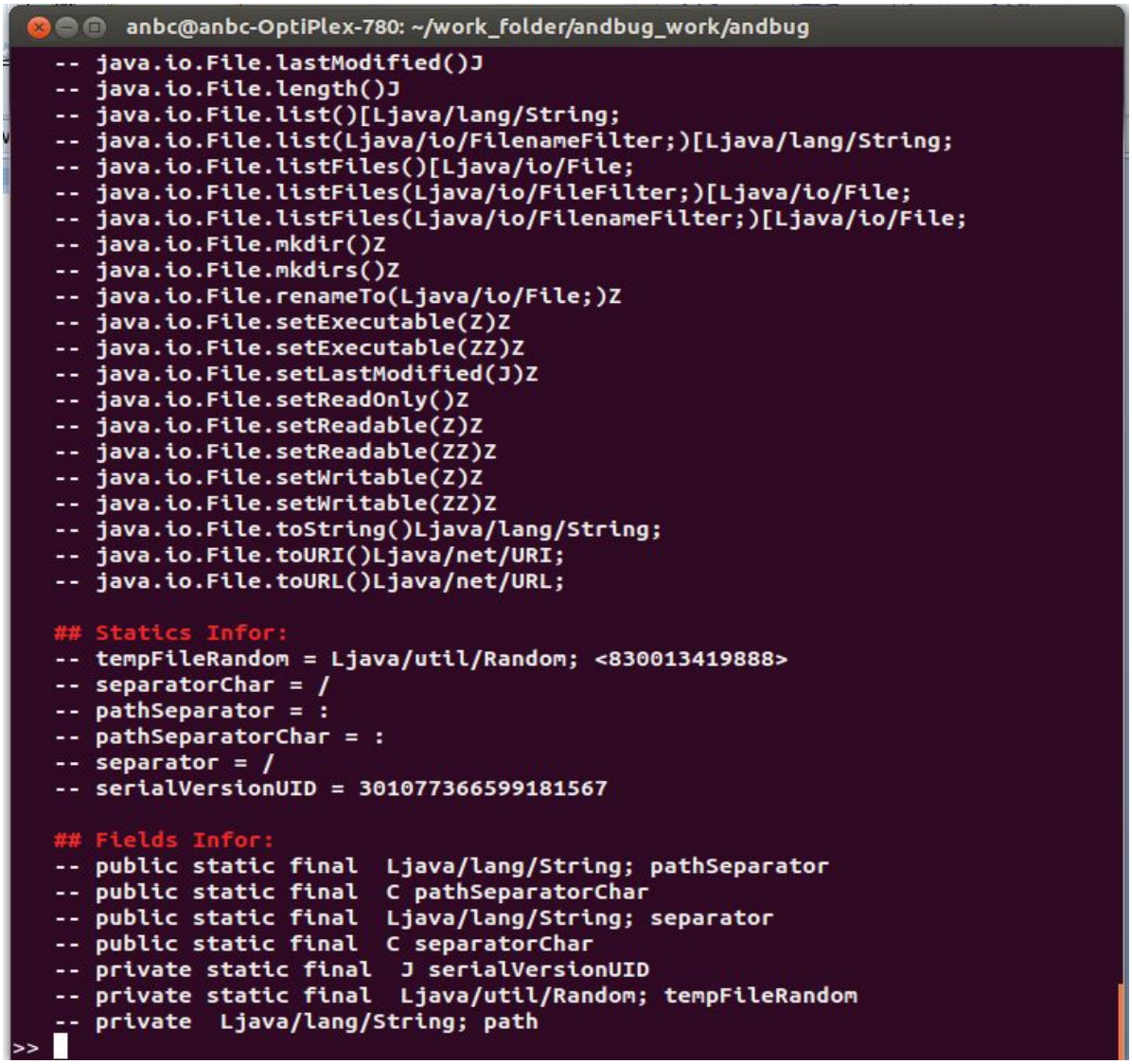
```
## AndBug (C) 2011 Scott W. Dunlop <swdunlop@gmail.com>
>> vm-cap
## java vm capability:
   -- canWatchFieldAccess:0
   -- canGetSourceDebugExtension:0
   -- canGetMonitorInfo:0
   -- canAddMethod:0
   -- canUnrestrictedlyRedefineClasses:0
   -- canPopFrames:0
   -- canUseInstanceFilters:0
   -- canGetSyntheticAttribute:1
   -- canRedefineClasses:0
   -- canGetOwnedMonitorInfo:0
   -- canGetCurrentContendedMonitor:0
   -- canSetDefaultStratum:0
   -- canWatchFieldModification:0
   -- canGetBytecodes:0
   -- canRequestVMDeathEvent:0
>>
```

具体含义请查阅相关资料

2、类的详情

命令：class-detail java.io.File
功能：展示出指定类的成员方法、成员变量和静态变量的信息





3、方法的详细内容

命令：method-detail java.io.File mkdir

```
## AndBug (C) 2011 Scott W. Dunlop <swdunlop@gmail.com>
>> method-detail java.io.File mkdir
## Methods Ljava/io/File;->mkdir
   -- +++call load methods

   ## Method Detail:
      java.io.File.mkdir()Z
   ## LOCATION:

   -- firstLoc=0  line=872
   -- lastLoc=8
   -- lineTable infor:
      -- loc=0  line=872
      -- loc=3  line=873
      -- loc=5  line=874
      -- loc=6  line=875

   ## ARGUMENT:
   -- java/io/File  this

   ## VARIABLE:
   -- libcore/io/ErrnoException  errnoException
>>
```

4、对 apk 的运行进行监控

4.1 执行监控命令

命令：./andbug monitor -p com.android.browser
功能：对浏览器的运行情况进行监控

```
anbc@anbc-OptiPlex-780:~/work_folder/andbug_work/andbug$ ./andbug monitor -p com
.android.browser
status=0
status=0
status=0
status=0
status=0
task_file_path:/home/anbc/work_folder/andbug_work/andbug/data/monitor_fun.conf
```

4.2、监控点在 monitor_fun.conf 文件中配置

```
in android.widget.Toast.makeText
####################################
#文件相关的监控项
####################################
#in out inout 监控的类型，包括：in函数调用时监控； out 函数调用完成时监控； inout函数调用时和调用完成后两个点都监控
in java.io.File.<init>
in java.io.File.createTempFile
out java.io.File.listFiles
out java.io.File.list
in java.io.File.delete()
in java.io.File.deleteOnExit()
out java.io.File.mkdir()
```

4.3、监控到的日志信息

```
2014-01-03 17:56:44,119 monitor.py[line:80] DEBUG {"args": {"this": {"fields_infor": {"mUser": "Landroid/os/UserHandle; <830016088360>",
"mMainThread": "Landroid/app/ActivityThread; <830015906008>"}, "object_type": "Landroid/app/ContextImpl$ApplicationContentResolver;", "object_id":
830016088376}, "selectionArgs": null, "where": null, "uri": {"fields_infor": {"fragment": "Landroid/net/Uri$Part$EmptyPart; <830014892976>", "ssp":
"None", "authority": "Landroid/net/Uri$Part; <830016188984>", "uriString": "content://com.android.browser/images", "query": "Landroid/net/Uri$Part
$EmptyPart; <830014892976>", "path": "Landroid/net/Uri$PathPart; <830017317368>", "scheme": "content"}, "object_type": "Landroid/net/Uri
$HierarchicalUri;", "object_id": 830017317392}, "values": {"fields_infor": {"mValues": "Ljava/util/HashMap; <830018531640>"}, "object_type": "Landroid/
content/ContentValues;", "object_id": 830018531624}}, "name": "android.content.ContentResolver.update(Landroid/net/Uri;Landroid/content/
ContentValues;Ljava/lang/String;[Ljava/lang/String;)I:0", "thread": "thread <16> AsyncTask #5\t(running suspended)", "is_native": false}
2014-01-03 17:56:44,564 monitor.py[line:80] DEBUG {"args": {"selectionArgs": {"array_type": "[Ljava/lang/String;", "array_data": "['http://
www.google.com.hk/url?sa=t&source=web&cd=1&ved=0CCoQFjAA&url=http%3A%2F%2Fzh.wikipedia.org%2Fzh-cn%2F%25E8%2587%25AA%25E7%2584%25B6%25E5%25B0%258D%
25E6%2595%25B8&ei=2mvGUrTjJo6IiQfdsoHQDA&usg=AFQjCNFMXIsQHUuXa0huLwJs4reAPEeYaQ']"}, "selection": "'url == ?'", "projection": {"array_type": "[Ljava/
lang/String;", "array_data": "['url']"}, "this": {"fields_infor": {"mUser": "Landroid/os/UserHandle; <830015940088>", "mMainThread": "Landroid/app/
ActivityThread; <830015906008>"}, "object_type": "Landroid/app/ContextImpl$ApplicationContentResolver;", "object_id": 830015940104}, "uri":
{"fields_infor": {"fragment": "Landroid/net/Uri$Part$EmptyPart; <830014892976>", "ssp": "None", "authority": "Landroid/net/Uri$Part; <830016188984>",
"uriString": "content://com.android.browser/bookmarks", "query": "Landroid/net/Uri$Part$EmptyPart; <830014892976>", "path": "Landroid/net/Uri
$PathPart; <830016189152>", "scheme": "content"}, "object_type": "Landroid/net/Uri$HierarchicalUri;", "object_id": 830016189176}, "sortOrder": null},
"name": "android.content.ContentResolver.query(Landroid/net/Uri;[Ljava/lang/String;Ljava/lang/String;[Ljava/lang/String;Ljava/lang/String;)Landroid/
database/Cursor;:0", "thread": "thread <18> DataControllerHandler\t(running suspended)", "is_native": false}
2014-01-03 17:56:44,661 monitor.py[line:80] DEBUG {"args": {"this": {"fields_infor": {"this$0": "Lcom/android/browser/provider/BrowserProvider2;
<830015973096>"}, "object_type": "Lcom/android/browser/provider/BrowserProvider2$DatabaseHelper;", "object_id": 830015976192}}, "name":
"android.database.sqlite.SQLiteOpenHelper.getWritableDatabase()Landroid/database/sqlite/SQLiteDatabase;:0", "thread": "thread <16> AsyncTask #5\t
(running suspended)", "is_native": false}
```

4.4、可视化处理后的监控数据

三、需要进一步完善的功能

1、断点设置功能

目前的断点设置功能，仅支持类和方法上设置断点，在代码的其他未知还不支持设置断点。要想实现一个完整的逆向调试器，需要实现更丰富的断点功能。

2、单步调试功能

目前调试器还不支持单步调试功能，只能在具体函数调用处设置断点对该处程序的运行情况进行查询，需要进一步增加单步调试功能，包括:step inito、step over、step return

3、调试时与代码的关联处理

目前虽然后关联源码的的功能，但是很不完善。关联是需要进一步分别支持汇编代码和java 代码不同源码形式的关联，并且将源码显示和断点设置，单步调试等功能联动起来，才是一个完整的逆向调试功能。

4、对运行中的 apk 应用的行为进行动态监控

这部分作者已经增加了一部分功能，但是还很不完善，很多功能需要细化。欢迎大家一起交流。

5、其他功能的丰富

目前就想到了这些待实现的功能，可以丰富的功能还有很多。