

代码修改位置如上图。

结构分析：

参考资料

<http://www.cr173.com/html/24374_1.html>

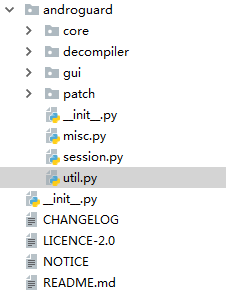
<http://blog.csdn.net/ybdesire/article/details/52629142>

<http://blog.csdn.net/jiangwei0910410003/article/details/50628894>

AndroBug 包名功能分析

Androguard 代码

目录树：



Utill.py 工具类：读取文件

**Misc.py**

Python类库：

cPickle 序列化 dump 序列化 load 反序列化

save\_session 序列化保存session对象

load\_session 反序列化session对象

AnalyzeAPK

AnalyzeDex

AnalyzeODex

RunDecompiler

AnalyzeElf

**Session.py**

序列化对象：file、digest、apk、dex

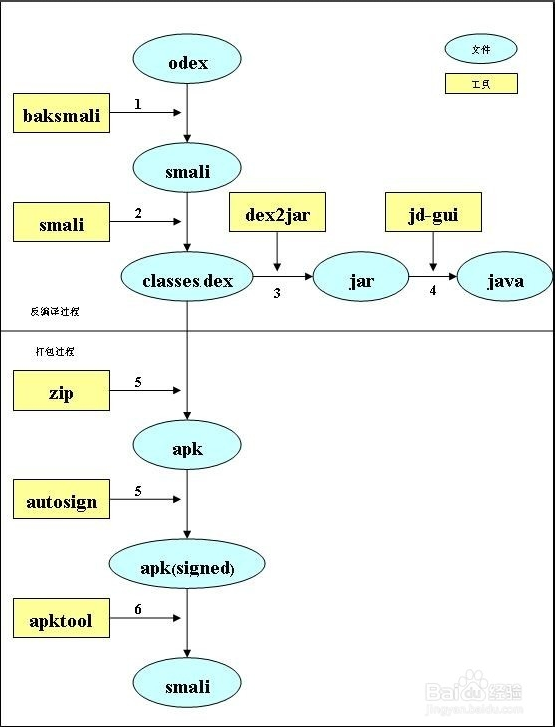
Patch包

Zipfile 读写ZIPfiles 解压与压缩文件。

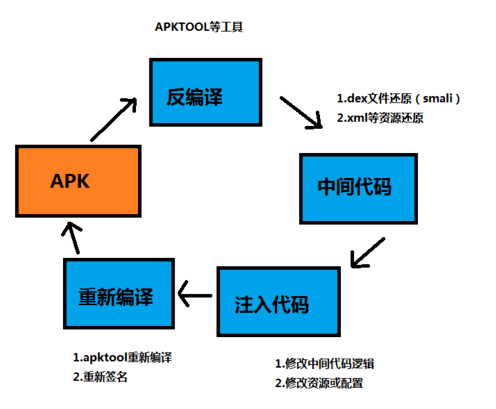
Gui包

图形化表示：具体后续再看

Android反编译and编译过程



反编译过程：



Decompiler

Python类库：

Subprocess 子线程

Tempfile 临时文件

Pygments 配置代码高亮

AndroGuard 支持：DEX, ODEX；

APK；

Android的二进制XML；

Android资源文件；

分解的DEX/ODEX 字节；

DEX/ODEX 文件反编译程序；

支持翻遍方式：dex2jar、dad、ded

<http://blog.csdn.net/working_harder/article/details/53287665>

工作流：

Dex对象 classes.dex

参考资料：<https://bbs.ichunqiu.com/thread-11244-1-1.html>

1.Androgen.py main函数开始类 分析过程开始

dvm.DalvikVMFormat( read(i) )

2.core/bytecodes dvm.py

def \_\_init\_\_(self, buff, decompiler=None, config=None, using\_api=None):

classmanager 初始化过程：

def \_\_init\_\_(self, vm, config):

解析dex头文件

def \_load(self, buff):  
 self.\_\_header = HeaderItem(0, self, ClassManager(None, self.config))

core/bytecodes/apk.py

apk文件的所有元素

analysis/analysis/VMAnalysis 分析dex文件对象，根据字符串搜索apk中的package

snov1ns6