猿人学–Android脱壳

壳的种类非常多，根据其种类不同，使用的技术也不同，这里稍微简单分个类：

- 一代整体型壳：采用 Dex 整体加密，动态加载运行的机制；
- 二代函数抽取型壳：粒度更细，将方法单独抽取出来，加密保存，解密执行；
- 三代 VMP、Dex2C 壳：独立虚拟机解释执行、语义等价语法迁移，强度最高。

先说最难的 `Dex2C` 目前是没有办法还原的，只能跟踪进行分析； `VMP` 虚拟机解释执行保护的是映射表，只要心思细、功夫深，是可以将映射表还原的；二代壳函数抽取目前是可以从根本上进行还原的， `dump` 出所有的运行时的方法体，填充到 `dump` 下来的 `dex` 中去的，这也是 `fart` 的核心原理。

frida_unpack

找到 DexFile 对象(art 虚拟机下是 DexFile 对象，dalvik 虚拟机下是 DexFile 结构体)，获取到 DEX 文件的起始地址和大小，然后 `dump` 下来。常见能够找到 `DexFile对象` 的函数有 `LoadMethod` 、 `ResolveMethod` 函数等，能直接获取到 `DEX起始地址和大小` 的常见函数有 `openMemory` 、 `dexparse` 、 `dexFileParse` 、 `dvmDexFileOpenPartial` 等函数。 `frida_unpack` 便是其中的代表作。

纯文本

```
https://github.com/dstmath/frida-unpack
https://github.com/GuoQiang1993/Frida-Apk-Unpack
https://github.com/lasting-yang/frida_dump
```

frida-Dexdump

利用 frida 的搜索内存，通过匹配 `DEX` 文件的特征，例如 `DEX` 文件的 `文件头` 中的 `magic` ---- `dex.035` 这个特征。 `frida-Dexdump` 便是这种脱壳方法的代表作。

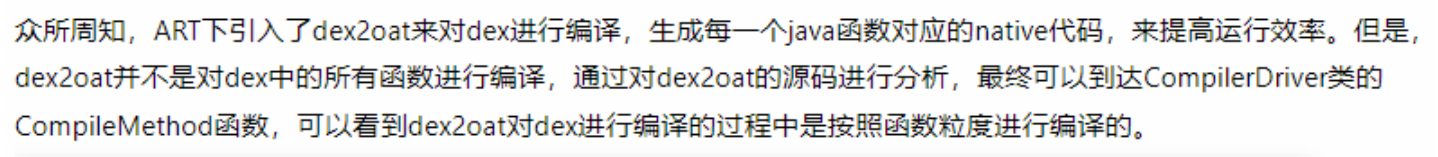
- 对于于完整的 dex，采用暴力搜索 dex035 即可找到。
- 而对于抹头的 dex，通过匹配一些特征来找到，然后自动修复文件头。

纯文本

```
https://github.com/hluwa/FRIDA-DEXDump

抽取    invoke    还原    再抽取
```

fart+youpk/fartext



可以看到在进行编译前进行了判断,最终可以发现，dex2oat对类的初始化函数并没有进行编译，那么也就是说类的初始化函数始终运行在ART下的inpterpreter模式，那么最终必然进入到interpreter.cc文件中的Execute函数，进而进入ART下的解释器解释执行。因此，我们便可以选择在Execute或者其他解释执行流程中的函数中进行dex的dump操作。事实上，当前一些壳通过阻断dex2oat的编译过程，导致了不只是类的初始化函数在解释模式下执行，也让类中的其他函数也运行在解释模式下。

关于Fart的脱壳点：

有两点，一个点是Execute函数，另一个点就是送到主动调用链的时候。

> ①初始化函数<clinit> - Execute => dumpDexFileByExecute
> ②其他正常函数=> DexFile_dumpMethodCode => myfartInvoke => Invoke => dumpArtMethod

关于主动调用链：

> ①启动fart线程-(getClassloader来获取ClassLoader)>
> ②fartwithClassLoader-(反射获取mCookie)>
> ③loadClassAndInvoke-(dumpMethodCode将各种函数转化成ArtMethod类型并送入我们的fake_Invoke参数包装)>
> ④送入系统的Invoke-(调用dumpArtMethod实现第二个脱壳点)。

Fart主动调用前提：

> ①获取appClassLoader;
> ②通过ClassLoader加载到所有类；
> ③通过每个类获取到该类下的所有方法【包括构造函数和普通函数】。

Java

```
[原创]拨云见日：安卓APP脱壳的本质以及如何快速发现ART下的脱壳点
https://bbs.pediy.com/thread-254555.htm

[原创]将FART和Youpk结合来做一次针对函数抽取壳的全面提升
https://bbs.pediy.com/thread-260052.htm
```


拓展阅读

纯文本

```
[原创]Android加壳脱壳学习（1）——动态加载和类加载机制详解
https://bbs.pediy.com/thread-271538.htm

[原创]一个Android壳简单实现
https://bbs.pediy.com/thread-248733.htm
https://github.com/huaerxiela/AndroidShell

[原创]Android第二代加固（support 4.4-8.1）
---就是一代不落地方式，叫法不一样
https://bbs.pediy.com/thread-225303.htm

Android逆向之Dalvik下一代壳通用解决方案(学习快记)
https://mp.weixin.qq.com/s/v_IHmJPvPryDJnsnxYcEfg

App逆向|Art下整体壳的解决方案(一)
https://mp.weixin.qq.com/s/FSRIEr9pgyXSImjfUD3t_Q

深入 FRIDA-DEXDump 中的矛与盾
https://www.anquanke.com/post/id/221905

[原创]分享一个自己做的函数抽取壳
```

```
https://bbs.pediy.com/thread-271139.htm
https://github.com/luoyesiqiu/dpt-shell
1，入口application替换，获取执行权
2，对dex中的codeitem进行抽取并保存文件
1，hook mmap 获取写权限
2，hook loadmethod ，当loadclass的时候会经过loadmethod，此时进行指令还原
3，load dex，merge dexelements

App逆向|ART下抽取壳的解决方案(二)
https://mp.weixin.qq.com/s/Gfpcwbpp3wtT3Pw-rwte7g
https://github.com/daisixuan/FridaUnpack

寒冰大佬文章
https://bbs.pediy.com/user-home-632473.htm

[原创]AUPK：基于Art虚拟机的脱壳机
https://bbs.pediy.com/thread-266716.htm

[原创]FartExt之优化更深主动调用的FART10
https://bbs.pediy.com/thread-268760.htm
https://github.com/dqzg12300/FartExt

课下问题：
1，Fartext实现主动调用函数逻辑的大致流程?
2，函数抽取恢复的过程是怎么和fart 执行 artmethod->invoke 方法关联起来的，
是在执行 artmethod->invoke 的时候壳会自己恢复函数吗？
回答：
https://bbs.pediy.com/thread-252630-5.htm

刷机前提：
自己去下载Android-sdk-platform-tools
配置好adb 和 fastboot以供使用
λ adb --version
Android Debug Bridge version 1.0.41
Version 33.0.1-8253317
Installed as D:\tools\Android\Sdk\platform-tools\adb.exe
λ fastboot --version
fastboot version 33.0.1-8253317
Installed as D:\tools\Android\Sdk\platform-tools\fastboot.exe

java 环境
λ java -version
openjdk version "1.8.0_332"
OpenJDK Runtime Environment (build 1.8.0_332-b09)
OpenJDK 64-Bit Server VM (build 25.332-b09, mixed mode)

抹头dex处理:https://blog.csdn.net/weixin_42453905/article/details/109185452
脱壳机适用手机：pixel 1
课程脱壳机压缩包解压密码：gebilaohuaajghalheu
脱壳机后续会保持更新
  主要更新检测对抗bypass方向
    最新版可在公众号[妄为写代码]回复[脱壳机]获取
当然，不管对后续更新感不感兴趣，本次课程的脱壳机都会上传猿人学课程网盘一份


echo "packageName" > /data/local/tmp/hext.config
记得给目标app打开储存空间权限

adb shell su -c "/data/local/tmp/fs_15.1.12_arm64"
```

引用文献

纯文本

```
实用FRIDA进阶：脱壳、自动化、高频问题
```

```
https://www.anquanke.com/post/id/197670

[原创]对脱壳脚本的一些改进--识别出目标DEX
https://bbs.pediy.com/thread-272076.htm
```