

目录

前言	1.1
重新打包apk概览	1.2
重新打包apk流程	1.3
apk解包	1.3.1
改动	1.3.2
重新打包apk	1.3.3
重签名	1.3.4
对齐	1.3.5
确认成功	1.3.6
相关工具	1.4
apktool	1.4.1
签名相关	1.4.2
keytool	1.4.2.1
jarsigner	1.4.2.2
apksigner	1.4.2.3
优化	1.4.3
zipalign	1.4.3.1
常见问题	1.5
心得	1.6
附录	1.7
参考资料	1.7.1

Android逆向：重新打包apk

- 最新版本： v0.7
- 更新时间： 20221029

简介

介绍Android逆向开发期间，如何重新打包apk。先是安卓apk重新打包的概览；然后详细介绍重新打包apk的典型流程，包括apk解包、改动、重新打包apk、重签名、对齐，最后确认成功；期间涉及常用工具apktool、签名相关的keytool、jarsigner、apksigner、优化相关的对齐工具zipalign；然后总结常见问题和相关心得；

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/android_re_repack_apk: Android逆向：重新打包apk](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [Android逆向：重新打包apk book.crifan.org](#)
- [Android逆向：重新打包apk crifan.github.io](#)

离线下载阅读

- [Android逆向：重新打包apk PDF](#)
- [Android逆向：重新打包apk ePub](#)
- [Android逆向：重新打包apk Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。
如发现有侵权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 `crifan` 还写了其他 150+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 12:04:57

重新打包apk概览

[Android逆向开发](#)期间，经常会涉及到，给安卓的apk重新打包。

- Android重新打包apk
 - 主要流程：用逆向工具(Apktool 等)反编译apk，得到各种资源和文件，编辑相关的内容，再用工具(Apktool 等)重新打包成apk。
 - 期间涉及
 - 逆向工具： Apktool 等
 - 重签名：签名和证书等
 - 优化：对齐等
 - 典型用途
 - 正向
 - 汉化
 - 技术研究和学习
 - 逆向
 - 破解
 - 免会员
 - 去广告
 - 黑灰产
 - 加上广告 -> 重新分发 -> 广告引流，挣钱
 - 加上病毒 -> 恶意事件：盗取数据，勒索等

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:51:00

重新打包apk流程

TODO:

- 【记录】下载和安装试用迅雷安卓app
 - 【已解决】Mac中下载和安装Thunder迅雷
 - 【未解决】Mac中尝试逆向或破解迅雷Thunder
 - 【记录】尝试逆向破解安卓app: 迅雷
 - 【未解决】给破解和脱壳后的安卓迅雷重新打包出可用apk
 - 【已解决】给腾讯乐固加固的安卓app脱壳后重新打包apk的逻辑和思路
-

给一个安卓apk重新打包apk的典型流程是：

- apk解包 : 用apktool解包, 得到文件和资源
- 改动 : 改你要的内容或资源
- 重新打包apk : 用apktool重新打包出apk
- 重签名 : 重新签名apk
- 对齐 : 用zipalign对齐
- 确认成功 : 用aapt确保能正常检测出apk信息, 重新安装apk可以正常安装不报错

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:

2022-10-29 11:26:50

apk解包

TODO:

- 【已解决】Mac中升级apktool到最新版
- 【记录】用apktool反编译破解迅雷安卓app

```
apktool d apk_file.apk
```

- d = decode = 解码
 - = 破解 = 反编译 = 解包

举例：

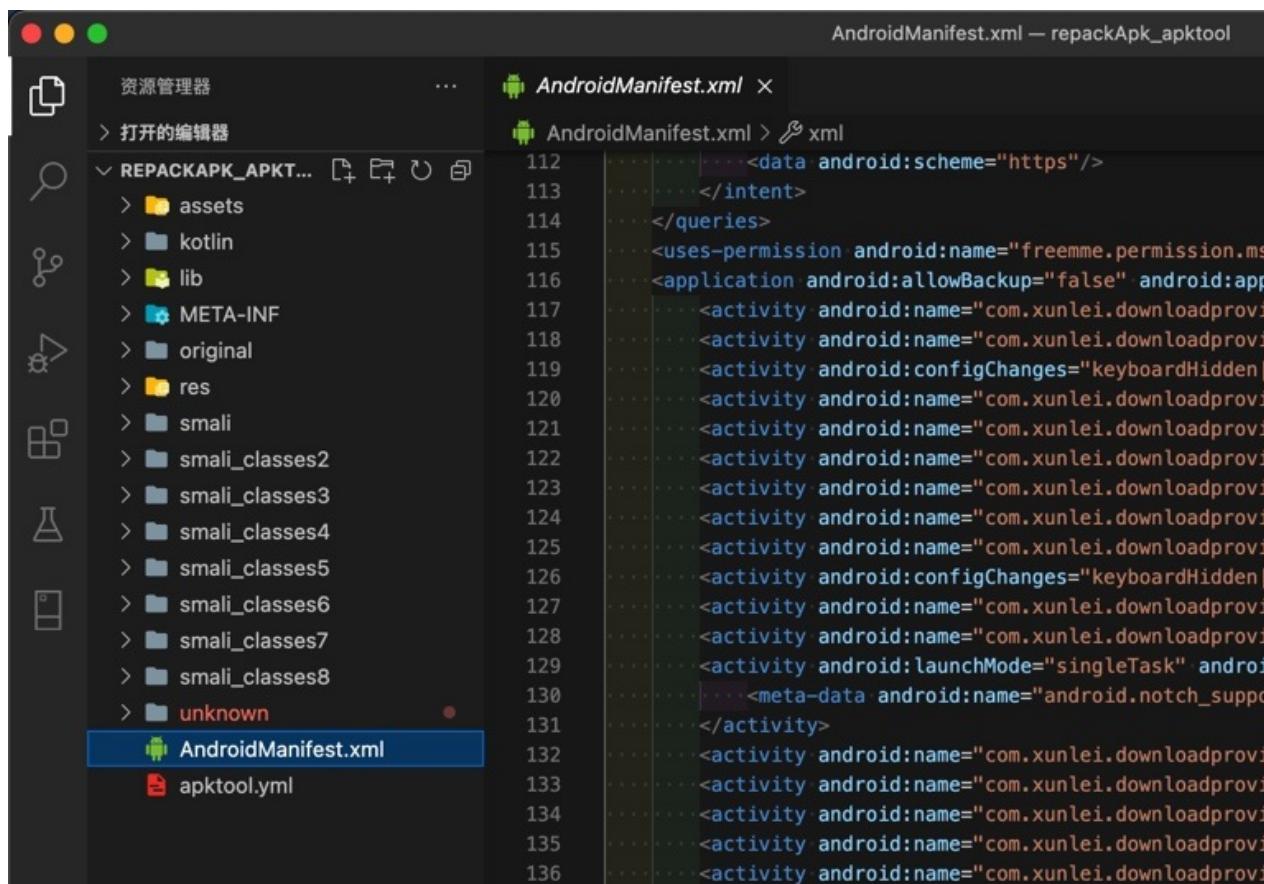
apktool解包迅雷的apk

```
../../../../reverse_engineering/apktool/apktool d ../../Thunder/OfficialSite_MobileThunder2.apk -f --only-main-classes -o
```

- 输出：反编译后的目录，包含各种文件

详细log：

```
../../../../reverse_engineering/apktool/apktool d ../../Thunder/OfficialSite_MobileThunder2.apk -f --only-main-classes -o
I: Using Apktool 2.5.0 on OfficialSite_MobileThunder2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /Users/crifan/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes4.dex...
I: Baksmaling classes5.dex...
I: Baksmaling classes6.dex...
I: Baksmaling classes7.dex...
I: Baksmaling classes8.dex...
I: Copying raw assets/39285EFA.dex file...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
```



apktool反编译小花生的apk

```

→ apk ll
total 51280
-rw-r--r--@ 1 crifan staff 25M 3 14 09:00 xiaohuashengv3.6.9_downcc.com.apk
→ apk apktool d xiaohuashengv3.6.9_downcc.com.apk
I: Using Apktool 2.4.0 on xiaohuashengv3.6.9_downcc.com.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (/Users/crifan/Library/apktool/framework), using /var/folders/46
/2hjxz38n22n3ypp_5f6_p__00000gn/T/ instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize
--frame-path if the default storage directory is unavailable
I: Loading resource table from file: /var/folders/46/2hjxz38n22n3ypp_5f6_p__00000gn/T/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
→ apk ll
total 51280
drwxr-xr-x 12 crifan staff 384B 3 14 13:39 xiaohuashengv3.6.9_downcc.com
-rw-r--r--@ 1 crifan staff 25M 3 14 09:00 xiaohuashengv3.6.9_downcc.com.apk
→ apk cd xiaohuashengv3.6.9_downcc.com
→ xiaohuashengv3.6.9_downcc.com ll
total 160

```

```
-rw-r--r-- 1 crifan staff 63K 3 14 13:39 AndroidManifest.xml
-rw-r--r-- 1 crifan staff 14K 3 14 13:39 apktool.yml
drwxr-xr-x 10 crifan staff 320B 3 14 13:39 assets
drwxr-xr-x 8 crifan staff 256B 3 14 13:39 kotlin
drwxr-xr-x 9 crifan staff 288B 3 14 13:39 lib
drwxr-xr-x 3 crifan staff 96B 3 14 13:39 libs
drwxr-xr-x 4 crifan staff 128B 3 14 13:39 original
drwxr-xr-x 143 crifan staff 4.5K 3 14 13:39 res
drwxr-xr-x 3 crifan staff 96B 3 14 13:39 smali
drwxr-xr-x 10 crifan staff 320B 3 14 13:39 unknown
```

得到项目的目录文件：

The screenshot shows a code editor interface with the title "apktool.yml — xiaohuashengv3.6.9_downcc.com". The left sidebar is labeled "EXPLORER" and shows the project structure:

- OPEN EDITORS: apktool.yml
- XIAOHUASHENGV3.6.9...:
 - assets
 - kotlin
 - lib
 - libs
 - original
 - res
 - smali
 - unknown
 - AndroidManifest.xml
- apktool.yml

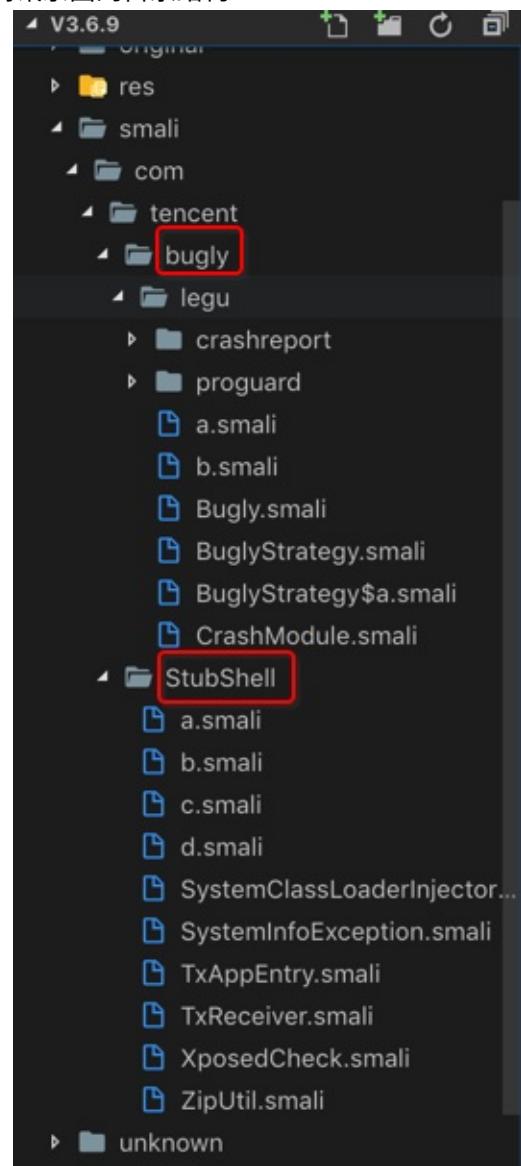
The right pane displays the contents of the apktool.yml file:

```
1  !!brut.androlib.meta.MetaInfo
2  apkFileName: xiaohuashengv3.6.9_downcc.com.apk
3  compressionType: false
4  doNotCompress:
5  - png
6  - res/drawable-hdpi-v4/abc_ab_share_pack_mtrl_alpha.9.png
7  - res/drawable-hdpi-v4/abc_btn_switch_to_on_mtrl_00001.9.png
8  - res/drawable-hdpi-v4/abc_btn_switch_to_on_mtrl_00012.9.png
9  - res/drawable-hdpi-v4/abc_cab_background_top_mtrl_alpha.9.png
10 - res/drawable-hdpi-v4/abc_list_divider_mtrl_alpha.9.png
11 - res/drawable-hdpi-v4/abc_list_focused_holo.9.png
12 - res/drawable-hdpi-v4/abc_list_longpressed_holo.9.png
13 - res/drawable-hdpi-v4/abc_list_pressed_holo_dark.9.png
14 - res/drawable-hdpi-v4/abc_list_pressed_holo_light.9.png
15 - res/drawable-hdpi-v4/abc_list_selector_disabled_holo_dark.9.png
16 - res/drawable-hdpi-v4/abc_list_selector_disabled_holo_light.9.png
17 - res/drawable-hdpi-v4/abc_menu_hardkey_panel_mtrl_mult.9.png
18 - res/drawable-hdpi-v4/abc_popup_background_mtrl_mult.9.png
19 - res/drawable-hdpi-v4/abc_scrubber_primary_mtrl_alpha.9.png
20 - res/drawable-hdpi-v4/abc_scrubber_track_mtrl_alpha.9.png
21 - res/drawable-hdpi-v4/abc_spinner_mtrl_am_alpha.9.png
22 - res/drawable-hdpi-v4/abc_switch_track_mtrl_alpha.9.png
23 - res/drawable-hdpi-v4/abc_tab_indicator_mtrl_alpha.9.png
24 - res/drawable-hdpi-v4/abc_textfield_activated_mtrl_alpha.9.png
25 - res/drawable-hdpi-v4/abc_textfield_default_mtrl_alpha.9.png
26 - res/drawable-hdpi-v4/abc_textfield_search_activated_mtrl_alpha.9.png
27 - res/drawable-hdpi-v4/abc_textfield_search_default_mtrl_alpha.9.png
28 - jpg
29 - res/drawable-hdpi-v4/bg_pop_pressed.9.png
```

其中有：

- 最基本的： AndroidMenifest.xml

- 但得不到我们要的 dex 文件
- 可得到：和app业务逻辑相关代码的 smali 文件
 - 想要得到最终 java 源码的话
 - 需要再去找 smali转java 的工具才可以
 - 此处即使不去转换得到java源码
 - 也可以从 smali 文件的目录结构和文件名，大概能看出app内部的类/文件名了
 - 而用 apktool 转换apk得到smali源码，是有前提的：apk没有加固
 - 加固了的apk反编译后只能看到被加固后的目录结构，看不到app业务逻辑代码和结构
 - 举例：某个被腾讯乐固legu加了密的apk，反编译后看不到原始类名和目录结构，只能看到腾讯乐固的目录结构



apktool反编译微信6.6.7的apk

后来去从[这里](#)下载到 安卓版微信v6.6.7，然后去用 apktool 去反编译：

```
→ apktool ./apktool d /Users/crifan/dev/dev_tool/android/apk/weixin/weixin_android_v6.6.7_minAndroid4.2.x.apk
I: Using Apktool 2.4.0 on weixin_android_v6.6.7_minAndroid4.2.x.apk
```

```

I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (/Users/crifan/Library/apktool/framework), using /var/folders/46
/2hjxz38n22n3ypp_5f6_p__00000gn/T/ instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize
--frame-path if the default storage directory is unavailable
I: Loading resource table from file: /var/folders/46/2hjxz38n22n3ypp_5f6_p__00000gn/T/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values /* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes4.dex...
I: Baksmaling classes5.dex...
I: Baksmaling classes6.dex...
I: Baksmaling classes7.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
→ apktool ll
total 65480
-rwxr-xr-x@ 1 crifan staff 2.3K 3 14 11:26 apktool
-rw-r--r--@ 1 crifan staff 16M 3 14 11:29 apktool.jar
-rw-r--r--@ 1 crifan staff 16M 3 14 11:29 apktool_2.4.0.jar
drwxr-xr-x 15 crifan staff 480B 4 30 17:35 weixin_android_v6.6.7_minAndroid4.2.x

```

然后去看看输出的信息：

```

→ apktool cd weixin_android_v6.6.7_minAndroid4.2.x
→ weixin_android_v6.6.7_minAndroid4.2.x ll
total 440
-rw-r--r-- 1 crifan staff 202K 4 30 17:35 AndroidManifest.xml
-rw-r--r-- 1 crifan staff 8.5K 4 30 17:35 apktool.yml
drwxr-xr-x 78 crifan staff 2.4K 4 30 17:35 assets
drwxr-xr-x 3 crifan staff 96B 4 30 17:35 lib
drwxr-xr-x 4 crifan staff 128B 4 30 17:35 original
drwxr-xr-x 118 crifan staff 3.7K 4 30 17:35 res
drwxr-xr-x 8 crifan staff 256B 4 30 17:35 smali
drwxr-xr-x 8 crifan staff 256B 4 30 17:35 smali_classes2
drwxr-xr-x 10 crifan staff 320B 4 30 17:35 smali_classes3
drwxr-xr-x 12 crifan staff 384B 4 30 17:35 smali_classes4
drwxr-xr-x 10 crifan staff 320B 4 30 17:35 smali_classes5
drwxr-xr-x 12 crifan staff 384B 4 30 17:35 smali_classes6
drwxr-xr-x 8 crifan staff 256B 4 30 17:35 smali_classes7
→ weixin_android_v6.6.7_minAndroid4.2.x cat AndroidManifest.xml
→ weixin_android_v6.6.7_minAndroid4.2.x cat apktool.yml
| brut.androlib.meta.MetaInfo
apkFileName: weixin_android_v6.6.7_minAndroid4.2.x.apk
compressionType: false
doNotCompress:
- arsc
- png
- sec

```

```
- conf
- dat
- txt
- data
- assets/infowindow_bg.9.png
- m4a
- wav
- assets/xwalk-command-line
- jpg
- mp3
- assets/wxa_library/local/IGNORE
- r/a/w6.9.png
- apk
- r/a7/v3.9.png
...
- r/q/aa2.9.png
- r/q/emoji_grid_item_bottom.9.png
- r/q/emoji_grid_item_left.9.png
- r/q/emoji_grid_item_middle.9.png
- gif
...
- r/r/ark.9.png
- r/r/emoji_app_msg_mask.9.png
- r/r/emoji_bottombar_bg.9.png
- r/r/emoji_box.9.png
- r/r/emoji_grid_item_fg_normal.9.png
- r/r/emoji_grid_item_fg_pressed.9.png
- r/r/tenpay_keybg.9.png
- r/r/tenpay_keyitem_bottom.9.png
- r/r/v3.9.png
...
- r/z/w8.9.png
isFrameworkApk: false
packageInfo:
  forcedPackageId: '127'
  renameManifestPackage: null
sdkInfo:
  minSdkVersion: '17'
  targetSdkVersion: '23'
sharedLibrary: false
sparseResources: false
unknownFiles: {}
usesFramework:
  ids:
    - 1
    tag: null
version: 2.4.0
versionInfo:
  versionCode: '1321'
  versionName: 6.6.7
```

```

apktool.yml — weixin_android_v6.6.7_minAndroid4.2.x

apktool.yml x
sdkVersion

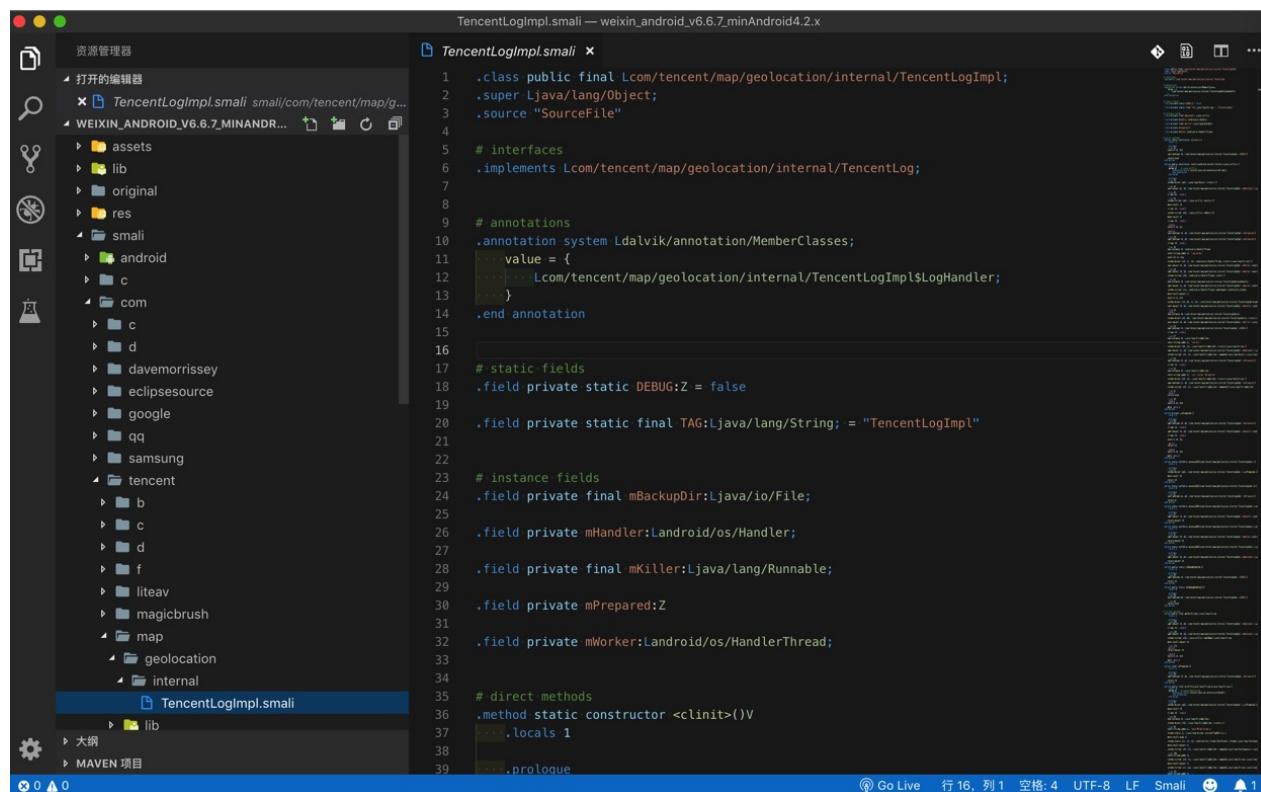
508 - r/r/zf.9.png
509 - r/s/w6.9.png
510 - r/y/w6.9.png
511 - r/z/acl.9.png
512 - r/z/as4.9.png
513 - r/z/as5.9.png
514 - r/z/asa.9.png
515 - r/z/asb.9.png
516 - r/z/emoji_box.9.png
517 - r/z/v_.9.png
518 - r/z/va.9.png
519 - r/z/w6.9.png
520 - r/z/w7.9.png
521 - r/z/w8.9.png
522 isFrameworkApk: false
523 packageInfo:
524 forcedPackageId: '127'
525 renameManifestPackage: null
526 sdkInfo:
527 minSdkVersion: '17'
528 targetSdkVersion: '23'
529 sharedLibrary: false
530 sparseResources: false
531 unknownFiles: {}
532 usesFramework:
533 ids:
534 - 1
535 tag: null
536 version: 2.4.0
537 versionInfo:
538 versionCode: '1321'
539 versionName: 6.6.7
540

```

可以看出一些版本方面的信息：

- minSdkVersion: '17' : 最低安卓版本 17
 - 17对应：安卓 4.2 , 4.2.2
- targetSdkVersion: '23' : 目标安卓版本 23
 - 23对应：安卓 6.0
- versionName: 6.6.7 : 微信版本号 6.6.7

以及有很多的smali代码：



```

TencentLogImpl.smali — weixin_android_v6.6.7_minAndroid4.2.x

1 .class public final Lcom/tencent/map/geolocation/internal/TencentLogImpl;
2 .super Ljava/lang/Object;
3 .source "SourceFile"
4
5 # interfaces
6 .implements Lcom/tencent/map/geolocation/internal/TencentLog;
7
8
9 # annotations
10 .annotation system Ldalvik/annotation/MemberClasses;
11 ... value = {
12     ...
13 }
14 .end annotation
15
16
17 # static-fields
18 .field private static DEBUG:Z = false
19
20 .field private static final TAG:Ljava/lang/String; = "TencentLogImpl"
21
22
23 # instance-fields
24 .field private final mBackupDir:Ljava/io/File;
25
26 .field private mHandler:Landroid/os/Handler;
27
28 .field private final mKiller:Ljava/lang/Runnable;
29
30 .field private mPrepared:Z
31
32 .field private mWorker:Landroid/os/HandlerThread;
33
34
35 # direct methods
36 .method static constructor <clinit>()V
37     .locals 1
38
39     .prologue

```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2022-10-29 11:33:17

改动

TODO:

【未解决】安卓逆向：用工具修改AndroidManifest.xml中的android:debugable为true让YouTube的apk可调试

举例：

迅雷的apk：改动版本号

此处最小的改动：修改版本号

`apktool.yml`

的

`versionName: 7.24.0.7525`

改为：

`versionName: 7.24.0.8000`

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:52:03

重新打包apk

TODO:

- 【记录】用apktool给脱壳后加了smali代码的各种文件重新打包为apk
- 【未解决】如何把破解和脱壳后的安卓apk重新打包出可用apk
-

```
apktool b inputFolder  
apktool b inputFolder -o outputFilename.apk
```

- `b = build` = 编译
 - = 重新打包
- `o = output` = 输出文件名
 - 默认是: dist/name.apk

举例:

重新打包迅雷的apk

```
apktool b -o thunderRepack_unsigned_unAlign.apk repackApk_jadx
```

或:

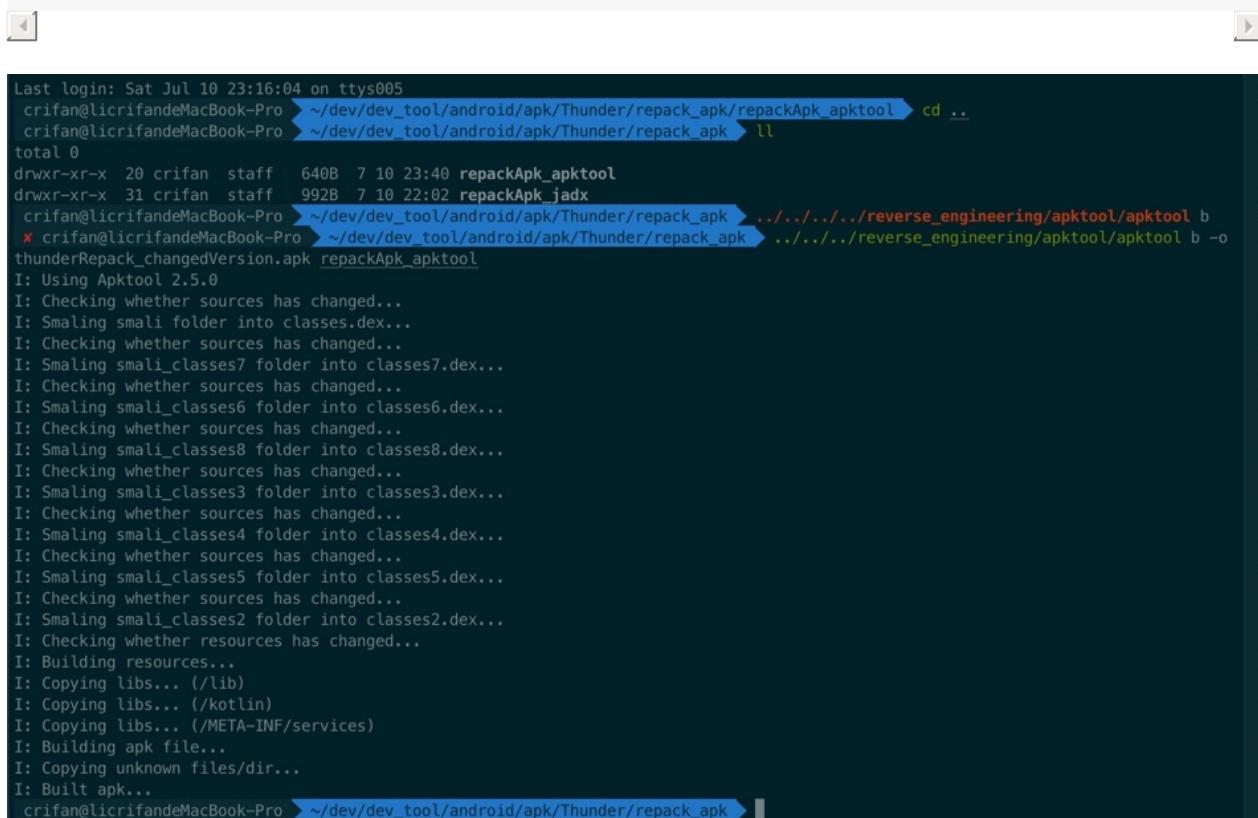
```
../../../../reverse_engineering/apktool/apktool b -o thunderRepack_changedVersion.apk repackApk_apktool
```

- 输出: 新的apk文件

详细log:

```
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/apk/Thunder/repack_apk 11  
total 0  
drwxr-xr-x 20 crifan staff 640B 7 10 23:40 repackApk_apktool  
drwxr-xr-x 31 crifan staff 992B 7 10 22:02 repackApk_jadx  
x crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/apk/Thunder/repack_apk ../../reverse_engineering/apktool/apktool b -o thunderRepack_changedVersion.apk repackApk_apktool  
I: Using Apktool 2.5.0  
I: Checking whether sources has changed...  
I: Smaling smali folder into classes.dex...  
I: Checking whether sources has changed...  
I: Smaling smali_classes7 folder into classes7.dex...  
I: Checking whether sources has changed...  
I: Smaling smali_classes6 folder into classes6.dex...  
I: Checking whether sources has changed...  
I: Smaling smali_classes8 folder into classes8.dex...
```

```
I: Checking whether sources has changed...
I: Smaling smali_classes3 folder into classes3.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes4 folder into classes4.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes5 folder into classes5.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```



```
Last login: Sat Jul 10 23:16:04 on ttys005
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_tool/android/apk/Thunder/repack_apk/repackApk_apktool cd ..
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/apk/Thunder/repack_apk ll
total 0
drwxr-xr-x 20 crifan staff 640B 7 10 23:40 repackApk_apktool
drwxr-xr-x 31 crifan staff 992B 7 10 22:02 repackApk_jadx
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/apk/Thunder/repack_apk ../../../../../../reverse_engineering/apktool/apktool b
x crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/repack_apk ../../../../../../reverse_engineering/apktool/apktool b -o
thunderRepack_changedVersion.apk repackApk_apktool
I: Using Apktool 2.5.0
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes7 folder into classes7.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes6 folder into classes6.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes8 folder into classes8.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes3 folder into classes3.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes4 folder into classes4.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes5 folder into classes5.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Copying libs... (/lib)
I: Copying libs... (/kotlin)
I: Copying libs... (/META-INF/services)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
crifan@licrifandeMacBook-Pro ~/dev/dev_tool/android/apk/Thunder/repack_apk █
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:

2022-10-29 11:52:27

重签名

TODO:

- 【已解决】给apktool重新打包的YouTube的apk重签名
- 【记录】用jarsigner给重新打包的安卓迅雷apk签名
- 【已解决】用keytool去给apk生成keystore证书

典型步骤是：

- 先用 keytool 生成证书文件
- 再用 jarsigner 去重新签名

举例：

迅雷的apk

用keytool去生成keystore证书

```
keytool -genkeypair -v -keystore thunderRepack.keystore -keyalg RSA -keysize 2048 -validity 10000 -alias thunderRepack
```

- 参数解释：
 - -genkeypair : 生成密码对（公钥和私钥）
 - -v : 表示verbose, 输出详细信息
 - -keystore thunderRepack.keystore : 生成的keystore文件名
 - -keyalg RSA : 秘钥算法用RSA
 - -keysize 2048 : key的大小采用2048
 - 此处也是RSA算法的默认大小值
 - -validity 10000
 - -alias thunderRepack : 设置别名，确保别名值是唯一，不重复的
 - 默认值是： mykey
 - 注：后续apk签名会用到这个alias值
- 输出：
 - keystore文件 : thunderRepack.keystore
- 其他说明
 - 期间有个秘钥的密码，要记住，以备后用
 - 此处设置的密码是：thunderRepack
 - 设置一堆信息后，最后需要确认
 - 最后确认信息时，（由于此处是中文提示信息），要输入：是
 - 如果像我输入yes，搞错了，就要重复再确认一遍。。。

用jarsigner签名

```
jarsigner -verbose -digestalg SHA1 -sigalg SHA1withRSA -keystore thunderRepack.keystore -signedjar thunderRepack_changedVersion_jarSigned.apk thunderRepack_changedVersion.apk thunderRepack
```

- 参数解释

- `-verbose` : 输出详情
- `-digestalg SHA1` : 摘要算法用SHA1
- `-sigalg SHA1withRSA` : 签名算法用SHA1withRSA
- `-keystore thunderRepack.keystore` : keystore文件, 用的是前面keytool生成的
- `-signedjar thunderRepack_changedVersion_jarSigned.apk` : output输出的, 签名后的, apk文件名
- `thunderRepack_changedVersion.apk` : 要签名的apk文件
- `thunderRepack` : 前面 (keytool生成的) keystore (指定的) 的alias别名

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:

2022-10-29 11:54:30

对齐

TODO:

- 【已解决】用zipalign给重新打包后的apk去对齐
-

举例：

迅雷apk

给迅雷的apk对齐

```
zipalign -p 4 thunderRepack_changedVersion_jarSigned.apk thunderRepack_changedVersion_jarSigned_aligned.apk
```

- 参数解释
 - `-p` : memory page alignment for stored shared object files
 - 如果要强制覆盖输出的文件，加 `-f`
 - `zipalign -f -p 4 xxx.apk xxx_aligned.apk`
 - 如果要输出详情，加 `-v`
 - `zipalign -v -f -p 4 xxx.apk xxx_aligned.apk`

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：

2022-10-29 12:03:09

确认成功

TODO:

- 【记录】对比确保apktool重新打包的YouTube的apk可以正常安装和使用
 - 【已解决】给腾讯乐固加固的安卓app脱壳后重新打包apk的逻辑和思路
-

- 验证重新打包apk成功
 - 用jad反编译
 - 看看是否能成功反编译，看到源码
 - 用aapt查看apk信息

```
aapt dump badging yourRepackedSigned.apk
```

- 确保能输出正常的信息，包括包名，版本等

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:49:24

相关工具

此处整理安卓逆向的重新打包apk期间常涉及到的工具。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-29 11:05:30

apktool

- apktool
 - 是什么：一个逆向安卓apk的工具
 - 主要功能
 - 解包
 - 输入： apk 文件
 - 输出：各种安卓 资源文件
 - 重新打包
 - 典型用途
 - 构建 自动化 工作 -> 取代重复的手动工作
 - 比如自动编译apk等工作
 - 给原有apk 添加额外东西
 - 比如反编译出apk后
 - 再本地化localizing
 - 添加其他功能
 - 分析 原有apk的功能和逻辑
- 如何安装
 - [Apktool - How to Install](#)
- github主页
 - [iBotPeaches/Apktool: A tool for reverse engineering Android apk files](#)
- 官网
 - [Apktool - A tool for reverse engineering 3rd party, closed, binary Android apps](#)
 - 官方提示
 - 请不要用apktool用于盗窃破解apk

apktool语法

```

apktool --help
Unrecognized option: --help
Apktool v2.5.0 - a tool for reengineering Android apk files
with smali v2.4.0 and baksmali v2.4.0
Copyright 2010 Ryszard Wiśniewski <brut.alll@gmail.com>
Copyright 2010 Connor Tumbleson <connor.tumbleson@gmail.com>

usage: apktool
--advance,--advanced  prints advance information.
--version,--version  prints the version then exits
usage: apktool if|install-framework [options] <framework.apk>
-p,--frame-path <dir>  Stores framework files into <dir> .
-t,--tag <tag>  Tag frameworks using <tag> .
usage: apktool d[ecode] [options] <file_apk>
-f,--force  Force delete destination directory.
-o,--output <dir>  The name of folder that gets written. Default is apk.out
-p,--frame-path <dir>  Uses framework files located in <dir> .
-r,--no-res  Do not decode resources.

```

```
-s,--no-src           Do not decode sources.  
-t,--frame-tag <tag>  Uses framework files tagged by <tag> .  
usage: apktool b[uild] [options] app_path  
-f,--force-all       Skip changes detection and build all files.  
-o,--output <dir>    The name of apk that gets written. Default is dist/name.apk  
-p,--frame-path <dir>  Uses framework files located in <dir> .
```

For additional info, see: <https://ibotpeaches.github.io/Apktool/>

For smali/baksmali info, see: <https://github.com/JesusFreke/smali>

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:

2022-10-29 11:30:30

签名相关

此处整理重新打包apk期间涉及到的，和签名相关的工具。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:05:56

keytool

- keytool: 生成keystore文件

```
keytool -genkey -alias demo.keystore -keyalg RSA -validity 40000 -keystore demo.keystore
```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-29 11:38:13

jarsigner

- jarsigner: 签名, JDK提供的针对jar包签名的通用工具
 - 位置
 - Win: `JDK/bin/jarsigner.exe`
 - 用法

```
jarsigner -verbose -keystore demo.keystore demo.apk demo.keystore
jarsigner -verify [待验证的apk]
```

apksigner

- `apksigner` : Google官方提供的针对Android apk签名及验证的专用工具
 - 位置:
 - Win: `Android SDK/build-tools/SDK版本/apksigner.bat`

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-29 11:40:09

优化

此处整理Android重新打包apk期间涉及到的，优化方面的工具。

比如：

- 对齐
 - `zipalign`

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:47:10

zipalign

- 对齐
 - 常用工具: `zipalign`
 - 为什么要对齐 = 对齐的目的: 优化性能
 - 对齐之前: CPU访问apk内部的未压缩的资源文件, 需要额外复制到RAM中, 再去访问
 - 增加了内存使用量
 - 对齐后: 直接用 `mmap` 去访问apk内部的资源文件
 - 无需消耗额外RAM, 降低了内存使用量, 提高了性能和效率

对齐的时机

- 如果您使用的是 `apksigner`, 只能在为APK文件签名之前执行 `zipalign`
 - 如果您在使用 `apksigner` 为APK签名之后对APK做出了进一步更改, 签名便会失效
- 如果您使用的是 `jarsigner`, 只能在为APK文件签名之后执行 `zipalign`

语法:

- 对齐

```
zipalign -p -f -v 4 infile.apk outfile.apk
```

- 验证是否对齐

```
zipalign -c -v 4 existing.apk
```

zipalign语法

```
zipalign -h
Zip alignment utility
Copyright (C) 2009 The Android Open Source Project

Usage: zipalign [-f] [-p] [-v] [-z] <align> infile.zip outfile.zip
        zipalign -c [-p] [-v] <align> infile.zip

<align>: alignment in bytes, e.g. '4' provides 32-bit alignment
-f: overwrite existing outfile.zip
-p: memory page alignment for stored shared object files
-v: verbose output
-z: recompress using Zopfli
```


常见问题

TODO:

- 【部分解决】apktool反编译安卓迅雷apk报错：Baksmaling assets dex Not a valid dex magic value
 - 【已解决】apktool重新打包报错：brut.androlib.AndrolibException brut.common.BrutException could not exec
 - 【已解决】搞懂重新打包apk流程中的证书签名和对齐的逻辑和命令
 - 【已解决】给重新打包后的apk签名用jarsigner还是apksigner
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：

2022-10-29 11:08:03

心得

关于安卓重新打包apk的心得：

- 少数apk：能顺利重新打包
 - 说明：
 - 本身防破解做的比较弱
 - 举例
 - 迅雷？
- 多数apk：无法顺利打包，会遇到很多问题，无法继续
 - 举例
 - YouTube？
 - 说明：
 - 本身防破解做的比较好，比较强
 - 即使开始能重新打包，但是后续安装apk也会报各种错误，即最终重新打包是失败的

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 11:35:26

附录

下面列出相关参考资料。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 09:52:26

参考资料

- 【已解决】apktool反编译安卓迅雷后用最小改动后再重新打包出新apk
-
- [Android反编译apk修改版本号重新打包签名详细教程（超详细）_马彦虎的博客-CSDN博客_android apk 反编译](#)
- [安卓apk反编译、修改、重新打包、签名全过程_dreamer2020的专栏-CSDN博客_apk编译](#)
- [Android逆向-反编译apk并重新打包 - 简书\(jianshu.com\)](#)
- [Android apk安全测评、应用加固、字节对齐、二次签名（有这一篇就够了）_osc_j34n26zn - MdEditor](#)
- [Android APK对齐总结 - 简书](#)
- [【Android】Apk签名及zipalign对齐_toaksg的博客-CSDN博客_zipalign对齐](#)
- [zipalign | Android 开发者 | Android Developers](#)
- [1.11 反编译APK获取代码&资源 | 菜鸟教程](#)
- [APK应用程序的解包、修改、编辑、打包及应用- IT开发者百科 - Powered by IT619.NET!](#)
- [android apk解包和打包_JaedongXue的博客-CSDN博客_apk解包](#)
- [简单记录Apk的解包打包 - 知乎](#)
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-29 12:04:49