# 目录

# iOS逆向开发：砸壳ipa

- 最新版本：`v0.9.1`
- 更新时间：`20231012`

## 简介

介绍iOS逆向中的砸壳脱壳出ipa方面的内容。主要包括什么是壳，为何要砸壳，常见砸壳工具，比如frida-ios-dump、dumpdecrypted、clutch等；以及举例介绍如何用frida-ios-dump砸壳YouTube、抖音等app得到ipa文件；以及砸壳出iap后的事情，包括ipa的安装；以及整理常见的问题及解决办法。

## 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/ios_re_crack_shell_ipa: iOS逆向开发：砸壳ipa

#### 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

- iOS逆向开发：砸壳ipa book.crifan.org
- iOS逆向开发：砸壳ipa crifan.github.io

### 离线下载阅读

- iOS逆向开发：砸壳ipa PDF
- iOS逆向开发：砸壳ipa ePub
- iOS逆向开发：砸壳ipa Mobi

## 版权和用途说明



## 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。

# 其他

## 作者的其他电子书

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

## 关于作者

关于作者更多介绍，详见：

关于CrifanLi李茂 – 在路上

# 砸壳ipa概览

## 什么是壳?

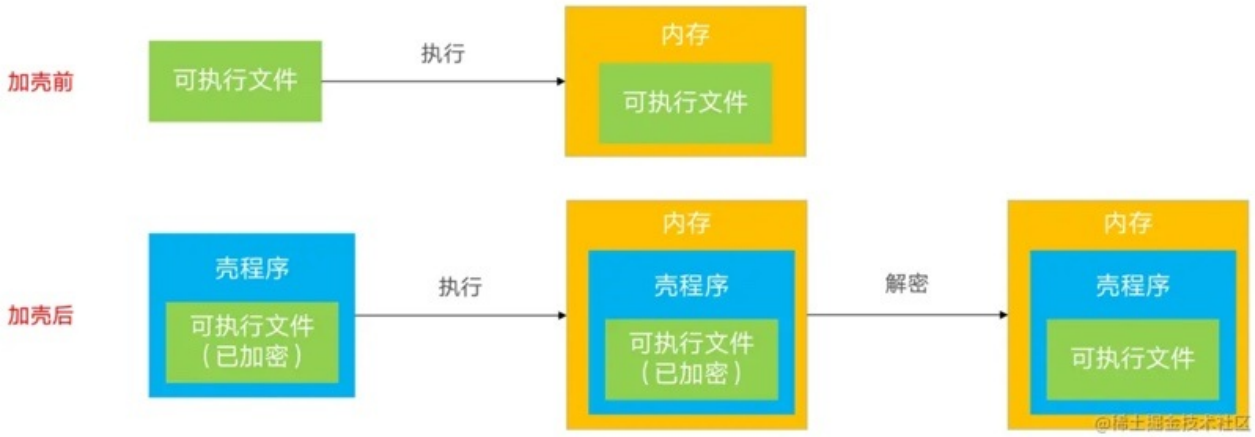
`壳`，在安全和逆向领域，泛指：用技术手段，给原程序额外加上一层保护程序

## 什么是iOS的app的壳?

iOS中的app，发布渠道一般都是 `App Store` 。

从 `App Store` 下载的APP全都是经过苹果加密过的 `ipa` 包。

而Apple会为了安全，给app加密(使用Apple ID相关的对称加密算法)，这个过程俗称为： `加壳` ，就像给app外部上加了一层壳




而加密后的 `ipa` 包，是无法继续后续的逆向过程的

- 后续的典型的逆向过程是
  - 用 `IDA / Hopper` 等去 `反编译`
  - 用 `class-dump` 等去 `导出头文件`
    - 说明
      - `class-dump` 直接去导出，未砸壳的， `App Store` 上的二进制的话
        - 只能导出 `CDStructures.h` 这个空的头文件，无法得到想要的各种类的头文件
  - 对砸壳后的ipa，去用 `MonkeyDev` 动态调试
  - 等等

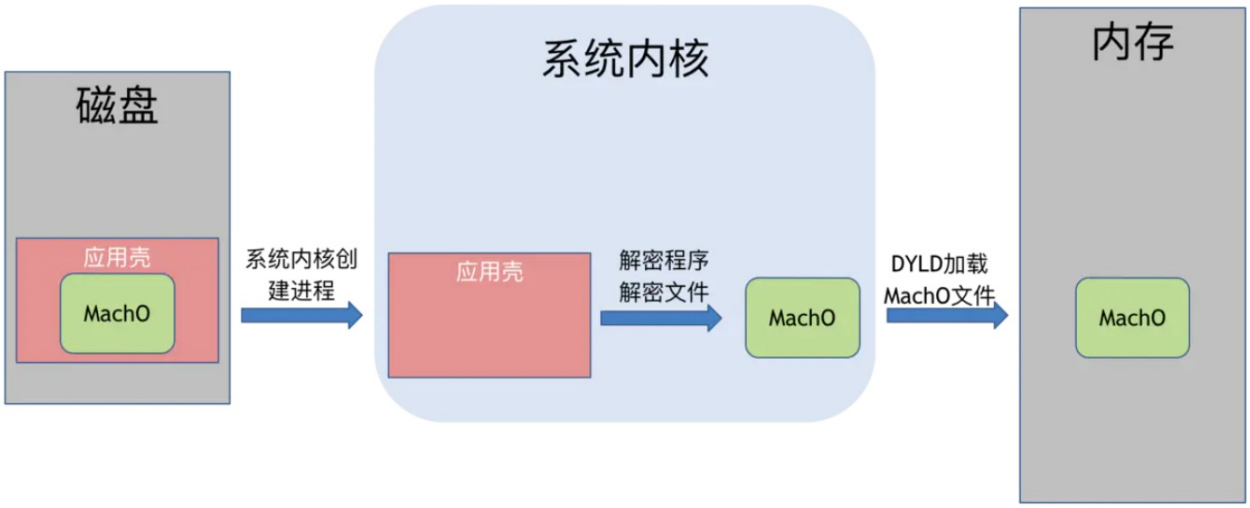
## 什么是iOS的砸壳 + 如何砸壳?

想要破解分析iOS的app之前，需要 `把这层壳砸破 = 砸壳 = 脱壳` 。

- 砸壳有两种机制
  - 静态砸壳：使用已知的解密方法对软件进行解密叫静态砸壳，静态砸壳难度大，需要知道其软件的加密算法才能对其解密
    - 现在没有这种工具
  - 动态砸壳

    - 现在绝大多数工具都是用此方式

如何（动态）砸壳呢？就要先了解app运行机制：app程序运行起来都会直接在内存解密出原始代码




可以在越狱的设备里面通过内存 `dump` 方式提取解密后的程序，这种解密过程，也就是给app去壳的过程，又称为 `砸壳` = `破壳`

- 额外说明
    - 解密之后还需要手动恢复 `Mach-O` 头信息才能运行
    - 由于高版本非完美越狱里面，都没有删掉签名验证
        - 所以直接运行都会出现 `killed 9`
            - 需要手动签名之后才能使用

## 砸壳的前提

- 确保iOS设备（iPhone等）已越狱
    - 详见：
        - iOS逆向开发：iPhone越狱

# 砸壳工具

常用的iOS的app的砸壳的工具：

- `frida-ios-dump`：最新，最好用，最常用
- 其他更早的工具
  - `dumpdecrypted`
    - 一般配合 `Cycript` 使用?
  - `clutch`
  - `bfinject`

# frida-ios-dump

- frida-ios-dump
  - 一句话描述：Pull a decrypted IPA from a jailbroken device
  - Github
    - AloneMonkey/frida-ios-dump: pull decrypted ipa from jailbreak device
      - https://github.com/AloneMonkey/frida-ios-dump

## 下载

- 下载命令

```
git clone https://github.com/AloneMonkey/frida-ios-dump.git
```

- 下载后的文件的说明
  - `dump.py` ： 最核心的文件，用来砸壳的Python脚本
  - `requirements.txt` ： Python依赖包的列表
    - 后续安装依赖的库，需要用到

## 初始化环境

- 先安装Frida
  - 概述
    - （Win/Mac等）电脑端

```
pip install frida
```

    - （iOS/Android等）移动端
      - iPhone
        - `Cydia` -> 添加源 https://build.frida.re -> 安装插件： `Frida`
  - 详解
    - 安装Frida · 逆向调试利器：Frida
- 再去安装依赖的其他的库
  - 直接用官网的依赖文件 `requirements.txt` 去安装

```
sudo pip install -r requirements.txt
```

  - 或已知需要哪些库，手动安装

```
pip install paramiko scp tqdm
```

- USB端口转发
  - 目的：方便本地直接访问对应端口，即可映射为，实际的iOS设备
  - 步骤
    - 概述

```
iproxy 2222 22
```

    - 详解
        - frida-ios-dump砸壳TikTok的ipa的实例

# 使用=砸壳

- 概述
    - 查看app包名或app名称
        - 方式1：`frida-ps`

          ```
          frida-ps -Uai
          ```

        - 方式2：`ideviceinstaller`

          ```
          ideviceinstaller -l -o list_user
          ```

    - 开始砸壳
        - 命令

          ```
          ./dump.py iOSAppPackageOriOSAppName
          ```

        - 举例

          ```
          ./dump.py com.zhiliaoapp.musically

          ./dump.py com.ss.iphone.ugc.Aweme

          ./dump.py com.google.ios.youtube
          ./dump.py YouTube
          ```

- 详解
    - 详见后续章节：frida-ios-dump实例

# dumpdecrypted

- dumpdecrypted
  - 一句话描述：iOS的砸壳工具
    - Dumps decrypted iPhone Applications to a file
  - 资料
    - GitHub
      - stefanesser/dumpdecrypted: Dumps decrypted mach-o files from encrypted iPhone applications from memory to disk. This tool is necessary for security researchers to be able to look under the hood of encryption.
        - https://github.com/stefanesser/dumpdecrypted

# Clutch

- `Clutch`
  - 是什么
    - Fast iOS executable dumper
    - a high-speed iOS decryption tool
  - 功能：脱壳=砸壳
    - 针对（越狱的）iOS设备，（解密）导出头文件
  - 支持平台
    - 所有iOS设备：iPhone/iPod Touch/iPad
  - 资料
    - GitHub
      - KJCracks/Clutch: Fast iOS executable dumper
        - https://github.com/KJCracks/Clutch
      - Wiki
        - Home · KJCracks/Clutch Wiki
          - https://github.com/KJCracks/Clutch/wiki
        - Tutorial · KJCracks/Clutch Wiki
          - https://github.com/KJCracks/Clutch/wiki/Tutorial
        - FAQ · KJCracks/Clutch Wiki
          - https://github.com/KJCracks/Clutch/wiki/FAQ

## help语法

```
Clutch [OPTIONS]
-b --binary-dump     Only dump binary files from specified bundleID
-d --dump            Dump specified bundleID into .ipa file
-i --print-installed Print installed application
--clean              Clean /var/tmp/clutch directory
--version            Display version and exit
-? --help            Display this help and exit
```

# bfinject

# 砸壳实例

此处介绍，具体如何用砸壳工具去砸壳出ipa文件。

# frida-ios-dump实例

此处介绍，如何用 `frida-ios-dump` 去砸壳出iOS的app的ipa文件。

# TikTok的ipa

给TikTok砸壳出ipa：

- 概述

```
./dump.py com.zhiliaoapp.musically
```

- 详解：

## 前提

- 前提：越狱iPhone中已安装 `TikTok`
  - 注：通过境外比如美区 `AppleID` 登录后的 `AppStore` 中才能搜索和下载 `TikTok`

## 砸壳ipa的步骤

（1）先确认app包名

```
ideviceinstaller -l -o list_user
```

输出能看到：

- `com.zhiliaoapp.musically, "268010", "TikTok"`

得到 `TikTok` 包名是：`com.zhiliaoapp.musically`

（2）确保Mac中当前Python中已安装frida（以及相关的库）

如果没有装，要去安装：

```
pip install frida paramiko scp tqdm
```

（3）另外新建一个终端，开启端口映射

新建一个终端窗口（或Tab），去运行端口映射

```
iproxy 2222 22
```

（4）确保frida版本一致：Mac中和iPhone中frida版本是一样的

说明：

- 如何查看frida版本
  - iPhone

```
frida-server --version
```

    - Mac

```
pip show frida
```

- 如果frida版本不一致
    - 后续会报错：`Failed to enumerate applications unable to communicate with remote frida-server`
        - `Failed to enumerate applications: unable to communicate with remote frida-server; please ensure that major versions match and that the remote Frida has the feature you are trying to use`
    - 需要去确保一致
        - 举例：
            - 此处frida版本：
                - Mac：`16.0.2`
                - iPhone：`15.1.27`
            - 如何解决
                - 去iPhone中`Cydia`中升级frida到最新版`16.0.2`

（5）确保被砸壳的app已退出，没在运行

可选？ iPhone中被砸壳的app，已退出，不要已启动真正运行

（6）真正开始砸壳

- 概述

```
./dump.py com.zhiliaoapp.musically
```

- 详解

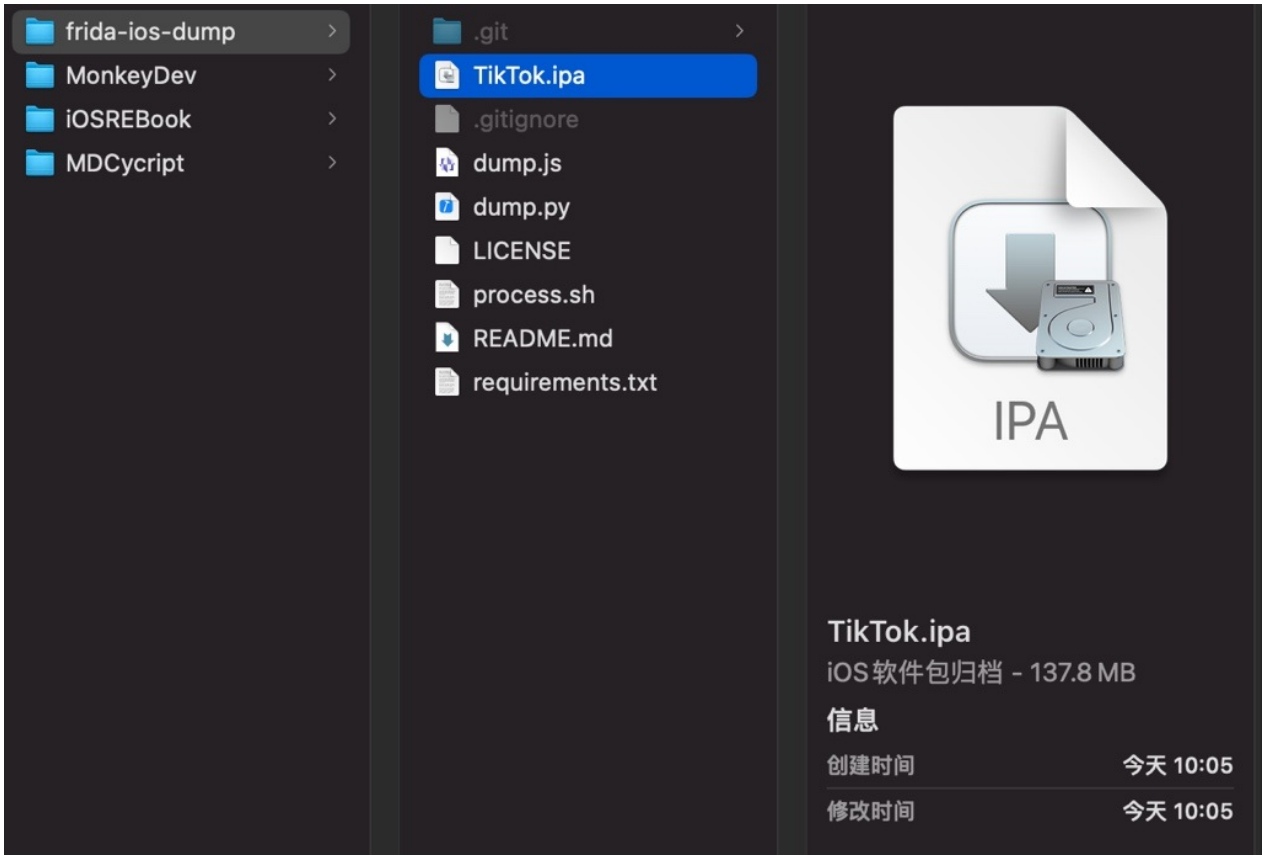
```
crifan@licrifandeMacBook-Pro『 ~/dev/dev_src/ios_reverse/AloneMonkey/frida-ios-dump『 ＼
master『 ./dump.py com.zhiliaoapp.musically
Start the target app com.zhiliaoapp.musically
Dumping TikTok to /var/folders/yy/46k2nmtx7nv344lm9zb6q66c0000gn/T
[frida-ios-dump]: Load libvcn.framework success.
[frida-ios-dump]: Load crypto.framework success.
[frida-ios-dump]: Load VolcEngineRTC.framework success.
[frida-ios-dump]: Load byteaudio.framework success.
[frida-ios-dump]: Load TikTokMSearchFramework.framework success.
[frida-ios-dump]: Load MuseDiscoverFramework.framework success.
[frida-ios-dump]: Load AAWELaunchTracker.framework success.
[frida-ios-dump]: Load RTCFFmpeg.framework success.
[frida-ios-dump]: Load TTFFmpeg.framework success.
[frida-ios-dump]: Load AAWEBootChecker.framework success.
[frida-ios-dump]: Load ffmpeg_dashdec.framework success.
[frida-ios-dump]: Load BDLRepairer.framework success.
[frida-ios-dump]: Load SCSDKCreativeKit.framework success.
[frida-ios-dump]: Load boringssl.framework success.
[frida-ios-dump]: MusicallyCore.framework has been loaded.
[frida-ios-dump]: Load SCSDKCoreKit.framework success.
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/TikTok
```

```
TikTok.fid: 100% [                                                    ]
[                                                    ]  74.5k/74.5k [00:00 00:00, 577kB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/MusicallyCore.framework/MusicallyCore
MusicallyCore.fid: 100% [                                                    ]
[                                                    ]  185M/185M [00:07 00:00, 24.9MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/BDLRepairer.framework/BDLRepairer
BDLRepairer.fid: 100% [                                                    ]
[                                                    ]  67.9k/67.9k [00:00 00:00, 542kB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/AAWEBootChecker.framework/AAWEBootChecker
AAWEBootChecker.fid: 100% [                                                    ]
[                                                    ]  72.6k/72.6k [00:00 00:00, 1.22MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/AAWELaunchTracker.framework/AAWELaunchTracker
AAWELaunchTracker.fid: 100% [                                                    ]
[                                                    ]  71.1k/71.1k [00:00 00:00, 1.46MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/RTCFFmpeg.framework/RTCFFmpeg
RTCFFmpeg.fid: 100% [                                                    ]
[                                                    ]  617k/617k [00:00 00:00, 6.91MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/SCSDKCoreKit.framework/SCSDKCoreKit
SCSDKCoreKit.fid: 100% [                                                    ]
[                                                    ]  351k/351k [00:00 00:00, 6.16MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/SCSDKCreativeKit.framework/SCSDKCreativeKit
SCSDKCreativeKit.fid: 100% [                                                    ]
[                                                    ]  129k/129k [00:00 00:00, 2.44MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/TTFFmpeg.framework/TTFFmpeg
TTFFmpeg.fid: 100% [                                                    ]
[                                                    ]  3.09M/3.09M [00:00 00:00, 23.4MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/VolcEngineRTC.framework/VolcEngineRTC
VolcEngineRTC.fid: 100% [                                                    ]
[                                                    ]  8.14M/8.14M [00:00 00:00, 28.6MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/boringssl.framework/boringssl
boringssl.fid: 100% [                                                    ]
[                                                    ]  632k/632k [00:00 00:00, 10.3MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/byteaudio.framework/byteaudio
byteaudio.fid: 100% [                                                    ]
[                                                    ]  1.60M/1.60M [00:00 00:00, 11.4MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/crypto.framework/crypto
crypto.fid: 100% [                                                    ]
[                                                    ]  1.44M/1.44M [00:00 00:00, 15.6MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/ffmpeg_dashdec.framework/ffmpeg_dashdec
ffmpeg_dashdec.fid: 100% [                                                    ]
[                                                    ]  101k/101k [00:00 00:00, 1.90MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/libvcn.framework/libvcn
```

```
libvcn.fid: 100% ████████████████████████████████████████████████████████
████████████████████████████████████  192k/192k [00:00 00:00, 2.94MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/TikTokMSearchFramework.framework/TikTokMSearchFramework
TikTokMSearchFramework.fid: 100% ████████████████████████████████████████████████████
████████████████████████████████████  3.68M/3.68M [00:00 00:00, 18.1MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF7
72C/TikTok.app/Frameworks/MuseDiscoverFramework.framework/MuseDiscoverFramework
MuseDiscoverFramework.fid: 100% ████████████████████████████████████████████████████
████████████████████████████████████  351k/351k [00:00 00:00, 4.65MB/s]
icon_home_dislike_new.json: 281MB [00:23, 12.7MB/s]
0.00B [00:00, ?B/s]Generating "TikTok.ipa"
```


成功的话，有相关日志输出：`Generating "TikTok.ipa"`

即可在当前目录找到砸壳后的ipa文件：`TikTok.ipa`



# 抖音的ipa

此处介绍，具体如何用 `frida-ios-dump` 去砸壳 `抖音` 的ipa文件：

- 概述

```
./dump.py com.ss.iphone.ugc.Aweme
```

## 详解

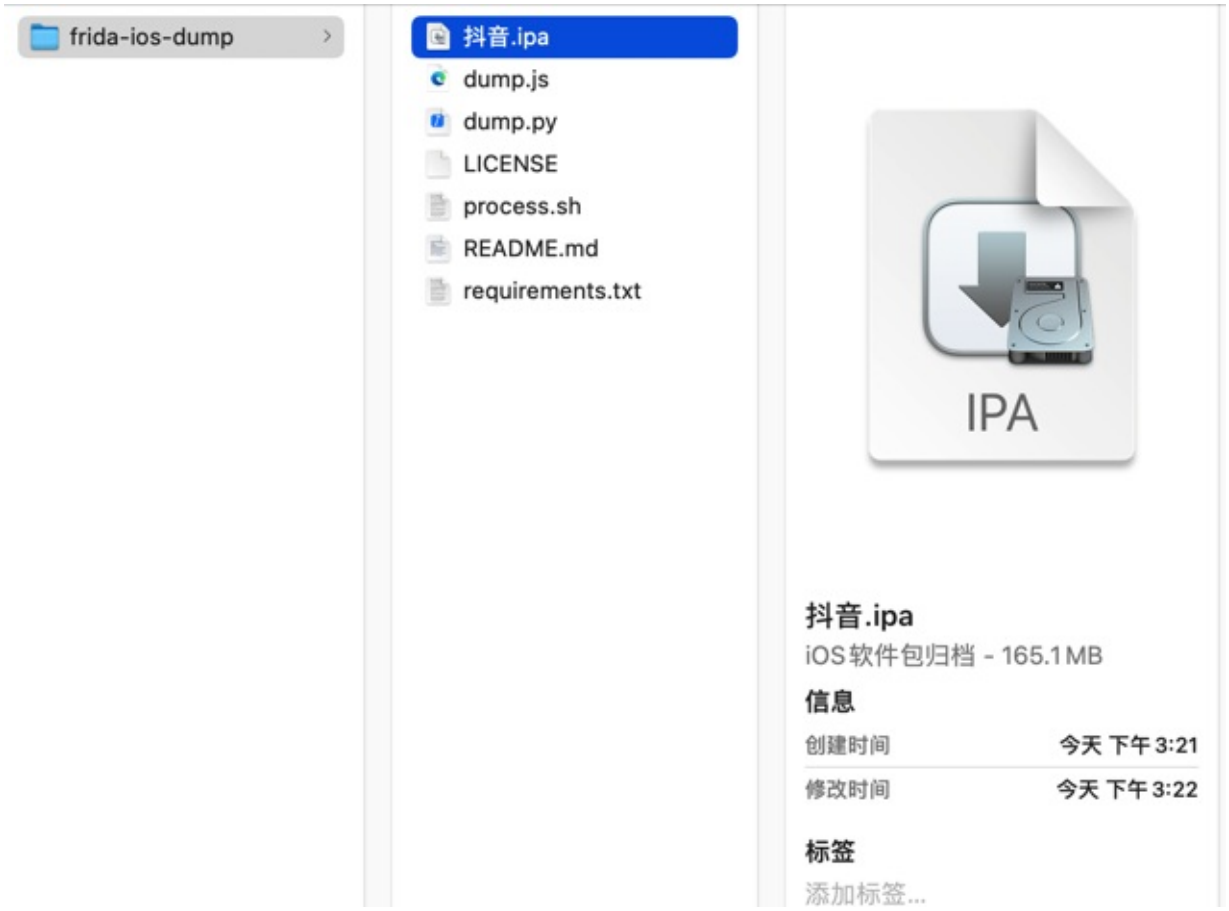
砸壳的具体过程和详细输出：

```
➜  frida-ios-dump git:(master) ./dump.py com.ss.iphone.ugc.Aweme
Start the target app com.ss.iphone.ugc.Aweme
Dumping 抖音 to /var/folders/2f/53mn2kn920dfq4ww2gdqfpxc0000gn/T
[frida-ios-dump]: Load VolcEngineRTC.framework success.
[frida-ios-dump]: Load byteaudio.framework success.
[frida-ios-dump]: Load AwemeCore.framework success.
[frida-ios-dump]: Load BDLRepairer.framework success.
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB
8F6/Aweme.app/Aweme
Aweme.fid: 100% |████████████████████████████████████████████████████████████████
████████████████████████████████████████| 71.7k/71.7k [00:00 00:00, 724kB
/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB
8F6/Aweme.app/Frameworks/AwemeCore.framework/AwemeCore
AwemeCore.fid: 100% |████████████████████████████████████████████████████████████████
████████████████████████████████████████| 230M/230M [00:07 00:00, 33.8MB
/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB
8F6/Aweme.app/Frameworks/BDLRepairer.framework/BDLRepairer
BDLRepairer.fid: 100% |████████████████████████████████████████████████████████████████
████████████████████████████████████████| 68.2k/68.2k [00:00 00:00, 1.76MB
/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB
8F6/Aweme.app/Frameworks/VolcEngineRTC.framework/VolcEngineRTC
VolcEngineRTC.fid: 100% |████████████████████████████████████████████████████████████████
████████████████████████████████████████| 10.6M/10.6M [00:00 00:00, 25.0MB
/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB
8F6/Aweme.app/Frameworks/byteaudio.framework/byteaudio
byteaudio.fid: 100% |████████████████████████████████████████████████████████████████
████████████████████████████████████████| 2.14M/2.14M [00:00 00:00, 23.2MB
/s]
Assets.car: 286MB [00:16, 18.6MB/s]
0.00B [00:00, ?B/s]Generating "抖音.ipa"
```

```
total 80
-rw-r--r--  1 crifan  staff   1.0K  1  4 15:16 LICENSE
-rw-r--r--  1 crifan  staff   3.2K  1  4 15:16 README.md
-rw-r--r--  1 crifan  staff    11K  1  4 15:16 dump.js
-rwxr-xr-x  1 crifan  staff    11K  1  4 15:16 dump.py
-rwxr-xr-x  1 crifan  staff   2.0K  1  4 15:16 process.sh
-rw-r--r--  1 crifan  staff   157B  1  4 15:16 requirements.txt
➜  frida-ios-dump git:(master) ./dump.py com.ss.iphone.ugc.Aweme
Start the target app com.ss.iphone.ugc.Aweme
Dumping 抖音 to /var/folders/2f/53mn2kn920dfq4ww2gdqfpxc0000gn/T
[frida-ios-dump]: Load VolcEngineRTC.framework success.
[frida-ios-dump]: Load byteaudio.framework success.
[frida-ios-dump]: Load AwemeCore.framework success.
[frida-ios-dump]: Load BDLRepairer.framework success.
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB8F6/Aweme.app/Aweme
Aweme.fid: 100%|                                                                                | 71.7k/71.7k [00:00<00:00, 724kB/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB8F6/Aweme.app/Frameworks/AwemeCore.framework/AwemeCore
AwemeCore.fid: 100%|                                                                            | 230M/230M [00:07<00:00, 33.8MB/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB8F6/Aweme.app/Frameworks/BDLRepairer.framework/BDLRepairer
BDLRepairer.fid: 100%|                                                                          | 68.2k/68.2k [00:00<00:00, 1.7GMB/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB8F6/Aweme.app/Frameworks/VolcEngineRTC.framework/VolcEngineRTC
VolcEngineRTC.fid: 100%|                                                                        | 10.6M/10.6M [00:00<00:00, 25.0MB/s]
start dump /private/var/containers/Bundle/Application/B625C6BC-D6B2-429F-B621-10A5EF7EB8F6/Aweme.app/Frameworks/byteaudio.framework/byteaudio
byteaudio.fid: 100%|                                                                            | 2.14M/2.14M [00:00<00:00, 23.2MB/s]
Assets.car: 286MB [00:16, 18.6MB/s]
0.00B [00:00, ?B/s]Generating "抖音.ipa"

➜  frida-ios-dump git:(master)
```

砸壳出的ipa文件：`抖音.ipa`



# YouTube的ipa

此处介绍，具体如何用 `frida-ios-dump` 去砸壳 `YouTube` 的ipa文件：

- 概述

```
./dump.py com.google.ios.youtube
```

## 详解

先查看出Youtube的包名：

```
➜  frida-ios-dump git:(master) ideviceinstaller -l -o list_user
CFBundleIdentifier, CFBundleVersion, CFBundleDisplayName
rn.notes.best, "11122019", "爱思极速版"
com.suiyi.foodshop1, "4911", "食行生鲜"
com.cisco.anyconnect, "4.6.03052", "AnyConnect"
com.baidu.BaiduMobile, "10.5.5.10", "百度"
com.ishuyin.iShuYin, "1.22", "爱书音"
com.evernote.iPhone.Evernote, "358974", "印象笔记"
com.alipay.iphoneclient, "10.1.2.091512", "支付宝"
ctrip.com, "8.3.0", "携程旅行"
com.Qting.QTTour, "8.0.1.4", "蜻蜓FM"
com.360buy.jdmobile, "7.3.6", "京东"
com.taobao.tmall, "10948419", "手机天猫"
com.netease.cloudmusic, "876", "网易云音乐"
com.tencent.mqq, "7.2.9.404", "QQ"
com.crifan.ShowSysInfo, "1", "ShowSysInfo"
com.tencent.xin, "8.0.16.35", "微信"
com.google.ios.youtube, "17.08.2", "YouTube"
developer.apple.wwdc-Release, "801.5.2", "Developer"
com.ss.iphone.ugc.Aweme, "179011", "抖音"
com.3WRHBBSBW4.com.rileytestut.AltStore, "1", "AltStore"
```

其中有我们要找的 `YouTube` 的详细信息：

- com.google.ios.youtube, "17.08.2", "YouTube"
  - 包名: `com.google.ios.youtube`
  - 版本: `17.08.2`
  - 名称: `YouTube`

砸壳的具体过程和输出日志：

```
➜  frida-ios-dump git:(master) pwd
/Users/crifan/dev/DevSrc/iOS/AloneMonkey/frida-ios-dump

➜  frida-ios-dump git:(master) ./dump.py com.google.ios.youtube
Start the target app com.google.ios.youtube
Dumping YouTube to /var/folders/2f/53mn2kn920dfq4ww2gdqfpxc0000gn/T
[frida-ios-dump]: Load widevine_cdm_secured_ios.framework success.
```

```
[frida-ios-dump]: Module_Framework.framework has been loaded.
start dump /private/var/containers/Bundle/Application/ECB295AB-1355-46D1-8580-273B2CE98
802/YouTube.app/YouTube
YouTube.fid: 100% ████████████████  16.3M/16.3M [00:00 00:00, 17.7MB/s]
start dump /private/var/containers/Bundle/Application/ECB295AB-1355-46D1-8580-273B2CE98
802/YouTube.app/Frameworks/widevine_cdm_secured_ios.framework/widevine_cdm_secured_ios
widevine_cdm_secured_ios.fid: 100% ▉  3.44M/3.44M [00:00 00:00, 24.2MB/s]
start dump /private/var/containers/Bundle/Application/ECB295AB-1355-46D1-8580-273B2CE98
802/YouTube.app/Frameworks/Module_Framework.framework/Module_Framework
Module_Framework.fid: 100% ██████████  114M/114M [00:03 00:00, 36.6MB/s]
Localizable.strings: 190MB [00:44, 4.45MB/s]
0.00B [00:00, ?B/s]
Generating "YouTube.ipa"
```

# 砸壳后

砸壳后，得到ipa文件后，还有些事情要做：

- 确认app已解密 = 确认砸壳成功
- 安装ipa

## 确认app已解密

可以用 `otool` 查看字段crypt的值：

- `cryptid=1`：已加密
- `cryptid=0`：没加密=已解密

### 举例

- 官网原始版本，安装到iPhone中后的，抖音的二进制文件 `Aweme`：已加密

```
➜  Aweme.app otool -l Aweme | grep crypt
    cryptoff 28672
    cryptsize 4096
      cryptid 0
```

- 砸壳后的抖音的ipa中的二进制文件 `Aweme`：已解密

```
➜  Aweme.app pwd
xxx/Aweme抖音/iPhone7-137black/Aweme.app
➜  Aweme.app otool -l Aweme | grep crypt
   cryptoff 28672
  cryptsize 4096
    cryptid 1
```

# 砸壳后安装ipa

砸壳得到ipa文件之后，如果后续需要动态调试，包括部分的静态分析，往往需要：

确保ipa可以正常安装

此时往往也会遇到很多问题：

- 安装ipa
  - 【已解决】换用Filza安装砸壳抖音ipa
  - 【已解决】越狱iPhone中删除之前通过ipa安装的抖音app
  - 【已解决】脱壳抖音ipa用爱思助手安装后启动失败闪退
  - 【已解决】把砸壳后抖音ipa安装到iPhone中
  - 【已解决】越狱iPhone中如何实现respring重启桌面SpringBoard
  - 【记录】对比研究抖音ipa不同方式安装后embedded.mobileprovision签名证书中appId的区别
  - 【已解决】确认抖音ipa的app内部是否有重签名证书文件embedded.mobileprovision
  - 【已解决】iPhone中Filza安装17.8.0抖音ipa报错：Failed to verify code signature The application does not have a valid signature
  - 【已解决】iPhone中Filza安装17.8.0抖音ipa报错：Application is missing the application identifier entitlement

# 砸壳常见问题

此处整理，砸壳出iap的常见问题和解决办法。

- DeviceNotSupportedByThinning
  - 【已解决】砸壳后抖音ipa安装失败：DeviceNotSupportedByThinning
- 启动崩溃
  - 【已解决】脱壳后抖音app启动就崩溃闪退
- 00:00无进度
  - 【已解决】frida-ios-dump砸壳抖音卡死无进度：0.00B 00:00

# 附录

下面列出相关参考资料。

# 参考资料

- 【已解决】iPhone中tiktok国际版砸壳出ipa文件
- 【已解决】iOS版抖音的砸壳脱壳
- 【已解决】Mac中用frida-ios-dump给iOS版抖音脱壳出ipa
- 【已解决】用frida-ios-dump给17.8.0版本抖音砸壳ipa
- 【已解决】Mac中用frida-ios-dump去砸壳出YouTube的ipa文件
- 【已解决】iPhone中查看已安装app的包名
- 【已解决】用frida-ios-dump砸壳报错：Failed to enumerate applications unable to communicate with remote frida-server
- 【已解决】iPhone中Cydia中安装升级最新版的16.0.2的Frida
- 【已解决】frida-ios-dump砸壳报错：ModuleNotFoundError No module named frida
- 【已解决】iOS的二进制用otool看不到crypt以及MachOView看不到LC_ENCRYPTION_INFO_64字段
-
- iOS逆向开发：iPhone越狱
- 安装Frida · 逆向调试利器：Frida
-
- frida-ps | Frida • A world-class dynamic instrumentation toolkit
- [iOS]判断ipa是否脱壳_风浅月明的博客-CSDN博客_ipa脱壳
- [iOS逆向]18、砸壳 - 简书 (jianshu.com)
- 十、iOS逆向之《越狱砸壳/ipa脱壳》 - 简书 (jianshu.com)
- iOS逆向：App脱壳/ipa破解-华盟网 (77169.net)
- iOS逆向攻防实战 - 掘金 (juejin.cn)
-