

# 目录

前言	1.1
砸壳ipa概览	1.2
常见砸壳工具	1.3
frida-ios-dump	1.3.1
dumpdecrypted	1.3.2
Clutch	1.3.3
bfinject	1.3.4
砸壳实例	1.4
frida-ios-dump	1.4.1
YouTube的ipa	1.4.1.1
抖音的ipa	1.4.1.2
TikTok的ipa	1.4.1.3
砸壳后	1.5
安装ipa	1.5.1
砸壳常见问题	1.6
附录	1.7
参考资料	1.7.1

# iOS逆向开发：砸壳ipa

- 最新版本: v0.7
- 更新时间: 20221116

## 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ipa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

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。

## 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](#)

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： 2022-11-16 14:16:46

# 砸壳ipa概览

## 什么是壳?

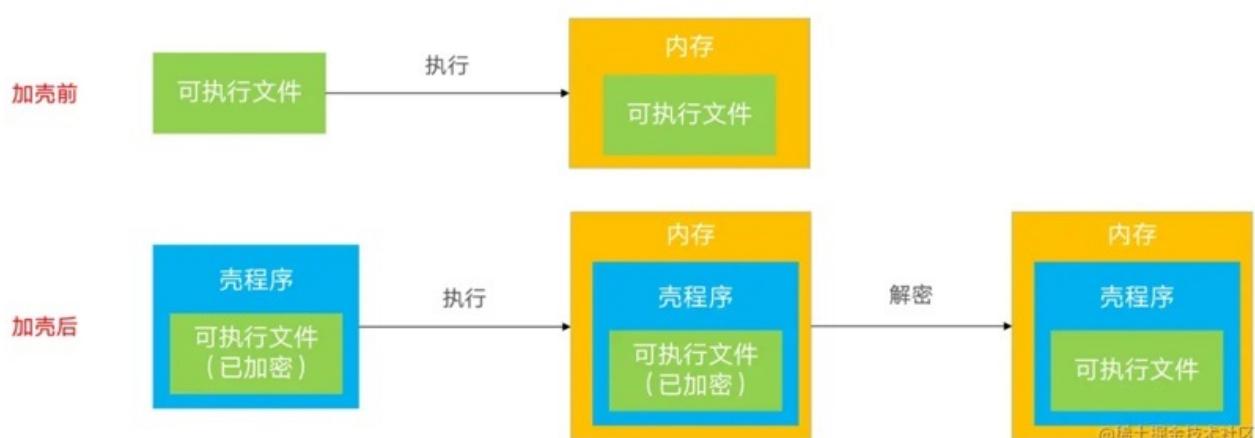
壳，在安全和逆向领域，泛指：用技术手段，给原程序额外加上一层保护程序

## 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

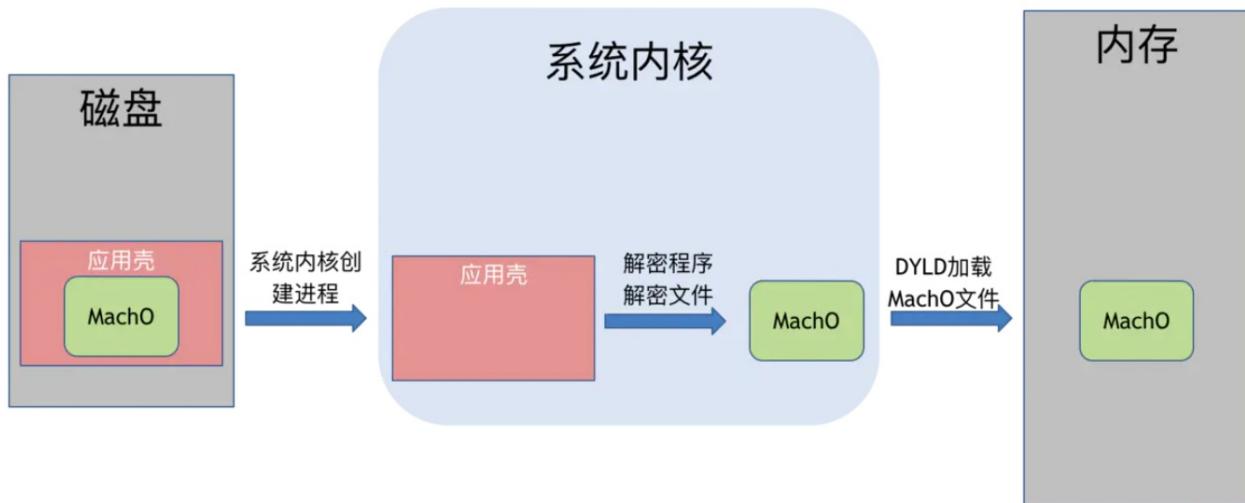
- 后续的典型的逆向过程是
  - 用 IDA / Hooper 等去 反编译
  - 用 class-dump 等去 导出头文件
    - 说明
      - class-dump 直接去导出，未砸壳的，App Store 上的二进制的话
        - 只能导出 CDStructures.h 这个空的头文件，无法得到想要的各种类的头文件
  - 对砸壳后的ipa，去用 MonkeyDev 动态调试
  - 等等

## 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进行解密
    - 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](#)

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：2022-10-21 17:49:35

# 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`

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：2022-11-08 11:35:03

## 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>

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: 2022-11-08 11:34:34

# 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>

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: 2022-10-21 17:12:54

# 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

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: 2022-10-21 11:52:29

## 砸壳实例

此处介绍，具体如何用砸壳工具去砸壳出ipa文件。

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：2022-10-21 17:23:08

## frida-ios-dump

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

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：2022-10-21 17:23:37

# YouTube的ipa

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

TODO:

【已解决】Mac中用frida-ios-dump去砸壳出YouTube的ipa文件

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: 2022-11-16 14:15:56

## 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。

TODO:

- 【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

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: 2022-11-16 14:16:14

# TikTok的ipa

TODO:

- 【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

给TikTok砸壳出ipa：

## 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
- 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

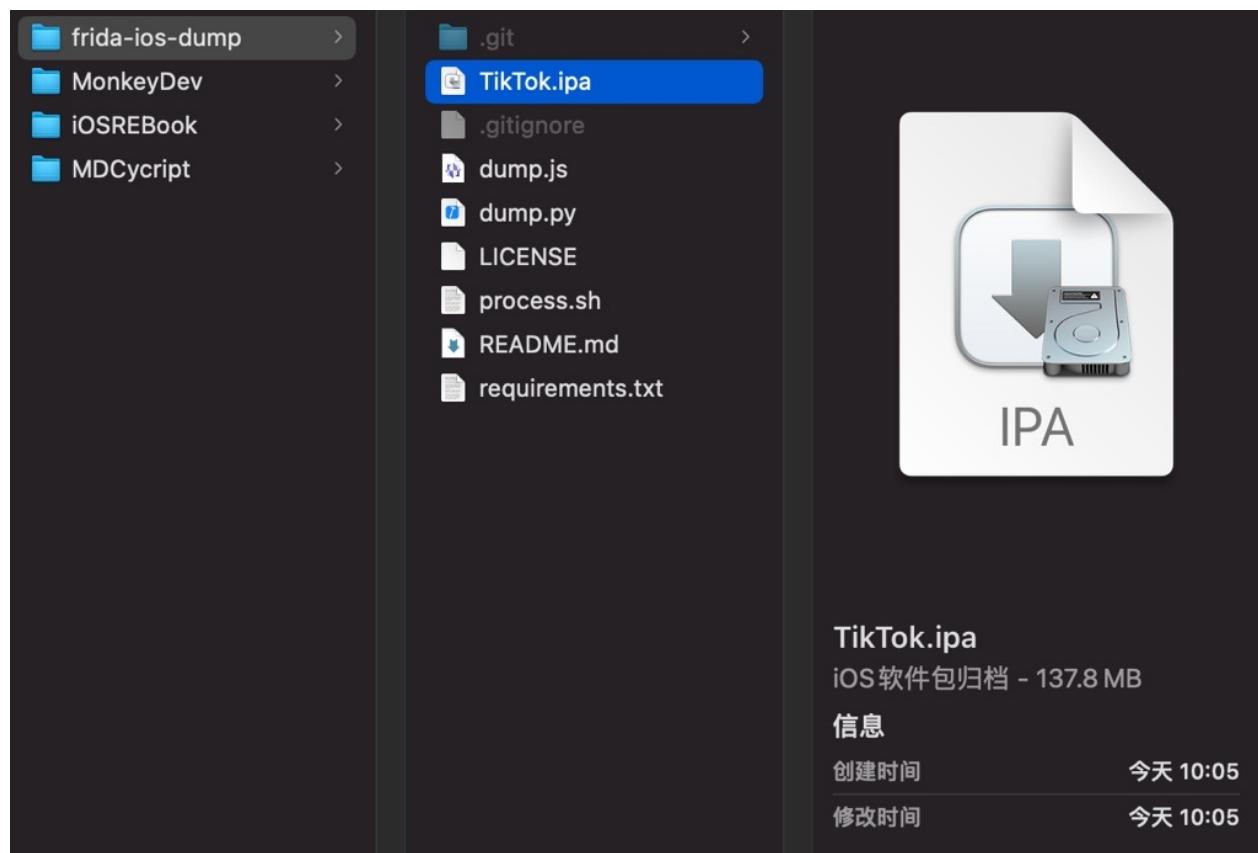
(6) 真正开始砸壳

```
cd /Users/crifan/dev/dev_src/ios_reverse/AloneMonkey/frida-ios-dump
./dump.py com.zhiliaapp.musically
```

```
crifan@licrifandeMacBook-Pro ~ ~/dev/dev_src/ios_reverse/AloneMonkey/frida-ios-dump
./dump.py com.zhiliaapp.musically
Dumping TikTok to /var/folders/yy/46k2nmx7nv344lm9zb6q66c0000gn/T
[frida-ios-dump]: Load libvcl.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]: Load MusicallyCore.framework has been loaded.
[frida-ios-dump]: Load SCSDKCoreKit.framework success.
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/TikTok.app/Frameworks/RTCFmpeg.framework/RTCFmpeg
RTCFmpeg.fid: 100% [ 617k/617k [00:00<00:00, 6.91MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/TikTok.app/Frameworks/bytaudio.framework/bytaudio
bytaudio.fid: 100% [ 1.60M/1.60M [00:00<00:00, 11.4MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF772C/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-E5BEACEF772C/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-E5BEACEF772C/TikTok.app/Frameworks/libvcl.framework/libvcl
libvcl.fid: 100% [ 192k/192k [00:00<00:00, 2.94MB/s]
start dump /private/var/containers/Bundle/Application/51BA1962-1A1E-40D8-AB83-E5BEACEF772C/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-E5BEACEF772C/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



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: 2022-11-16 14:14:49

## 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

```
otool -l QDReaderAppStore | grep crypt
```

TODO:

把自己的相关查看是否已解密的帖子整理过来

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：2022-10-21 17:35:04

## 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

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：2022-10-21 17:28:52

## 砸壳常见问题

此处整理，砸壳出ipa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

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：2022-10-21 17:27:01

## 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: 2022-03-17 20:39:28

## 参考资料

- 【已解决】iPhone中tiktok国际版砸壳出ipa文件
- 
- [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](#))
- 
- iOS逆向开发：iPhone越狱
- 

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：2022-11-16 14:02:46