

# 目录

前言	1.1
iOS逆向概述	1.2
iOS逆向领域架构图	1.2.1
iOS逆向内容概述	1.2.2
iOS典型逆向开发流程	1.2.2.1
iOS逆向重点和难点	1.2.3
子教程	1.3
心得	1.4
附录	1.5
参考资料	1.5.1

# iOS逆向开发

- 最新版本: v0.6
- 更新时间: 20221020

## 简介

概述如何进行iOS的逆向开发，包括核心流程，即先准备iPhone越狱，再到从app中砸壳出ipa，再到静态分析ipa中二进制，以及动态调试ipa，最后才是相关的iOS逆向开发内容，包括tweak越狱插件开发等内容。

## 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\\_reverse\\_dev: iOS逆向开发](#)

### 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 book.crifan.org](#)
- [iOS逆向开发 crifan.github.io](#)

### 离线下载阅读

- [iOS逆向开发 PDF](#)
- [iOS逆向开发 ePUB](#)
- [iOS逆向开发 MOBI](#)

## 版权说明

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

## 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 100+ 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](#)



## 概述

---

TODO:

把之前已发布的：

GitHub - crifan/prevent\_iphone\_hacked\_ios\_security: 防止iPhone被黑：iOS安全

[https://github.com/crifan/prevent\\_iphone\\_hacked\\_ios\\_security](https://github.com/crifan/prevent_iphone_hacked_ios_security)

整合进来

---

## iOS逆向概述

为了新手对于iOS逆向开发有个更全面直观的了解，下面先对iOS逆向开发做个总体的概述

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：2022-06-11 10:50:15

# iOS逆向领域架构图

此处整理，iOS逆向所涉及到等多个领域的知识，以及每个领域之间的关系，即：

iOS逆向领域架构图

- 在线浏览（支持缩放）
  - [iOS逆向开发内容架构图 | ProcessOn免费在线作图](#)
- 离线查看

◦

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：2022-10-20 15:09:53

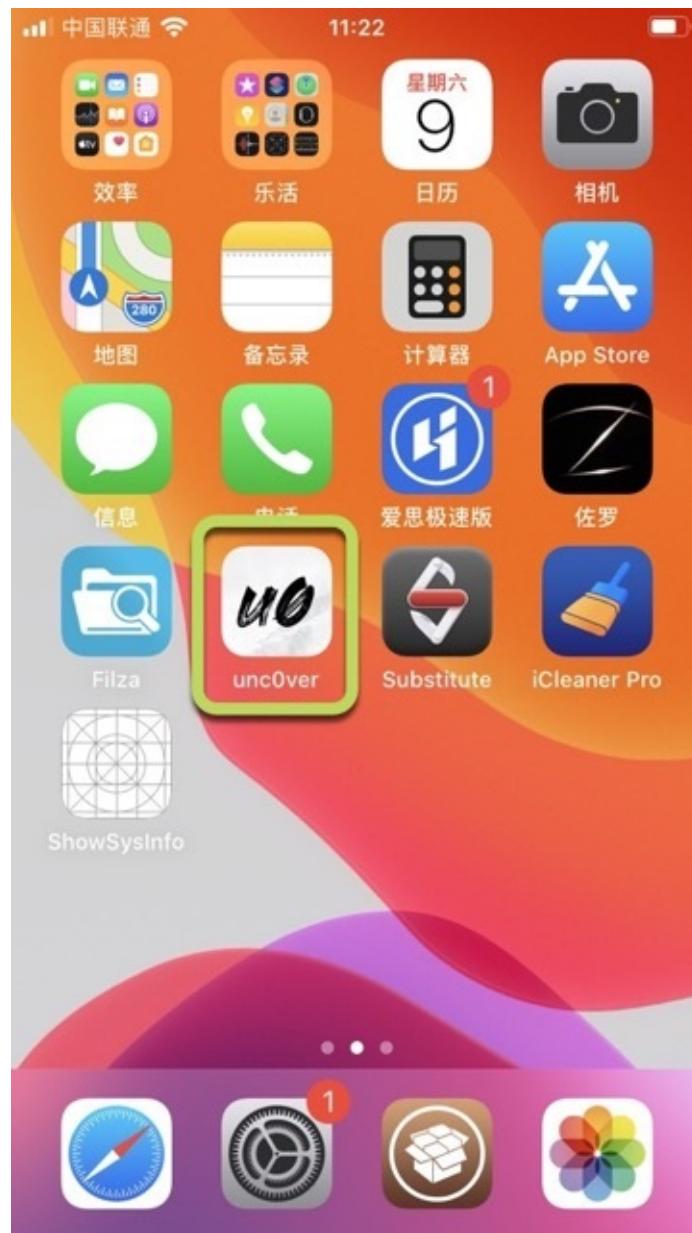
## iOS逆向内容概述

- 先准备越狱iPhone

- 越狱工具

- unc0ver

- 自己下载unc0ver到iPhone中



- 运行unc0ver去越狱

- checkra1n

- Mac中运行checkra1n，根据操作提示去越狱



- 也可以借助于 爱思助手 的 一键越狱 去越狱
  - 底层也是用unc0ver、checkra1n等工具



- 包括准备好常用越狱插件开发工具，比如
  - Filza : 文件管理
  - OpenSSH : ssh连接操作iPhone
  - AFC2 : 允许通过USB操作iPhone
  - iCleaner Pro : 临时禁止或启用插件
  - AppSync Unified : 免签名安装app
  - 等等
- 再从AppStore安装的正版app中砸壳出ipa

◦

- 常用工具

- frida-ios-dump

```
→ ~ iproxy 2222 22
Creating listening port 2222 for device port 22
waiting for connection
New connection for 2222->22, fd = 5
waiting for connection
Requesting connection to USB device handle 35 (serial: a
dab53e3250e8be1ee0db75bccdc2063df608b46), port 22
New connection for 2222->22, fd = 5
waiting for connection
Requesting connection to USB device handle 35 (serial: a
dab53e3250e8be1ee0db75bccdc2063df608b46), port 22
New connection for 2222->22, fd = 5
waiting for connection
Requesting connection to USB device handle 35 (serial: a
dab53e3250e8be1ee0db75bccdc2063df608b46), port 22
[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-1355-46D1-8580-273B2CE98802/
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-1355-46D1-8580-273B2CE98802/
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-1355-46D1-8580-273B2CE98802/
YouTube.app/Frameworks/Module_Framework.framework/Module_Framework
Module_Framework.fid: 100% [██████████] 114M/114M [00:03:00:00, 36.6MB/s]
0.00B [00:00, 7B/s]
```

```
→ 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/53mn2kn920dfq4ww2gdqfpvc0000gn/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-1355-46D1-8580-273B2CE98802/
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-1355-46D1-8580-273B2CE98802/
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-1355-46D1-8580-273B2CE98802/
YouTube.app/Frameworks/Module_Framework.framework/Module_Framework
Module_Framework.fid: 100% [██████████] 114M/114M [00:03:00:00, 36.6MB/s]
0.00B [00:00, 7B/s]
```

- Clutch

- dumpdecrypted

- bfinject

- 接着才是逆向开发

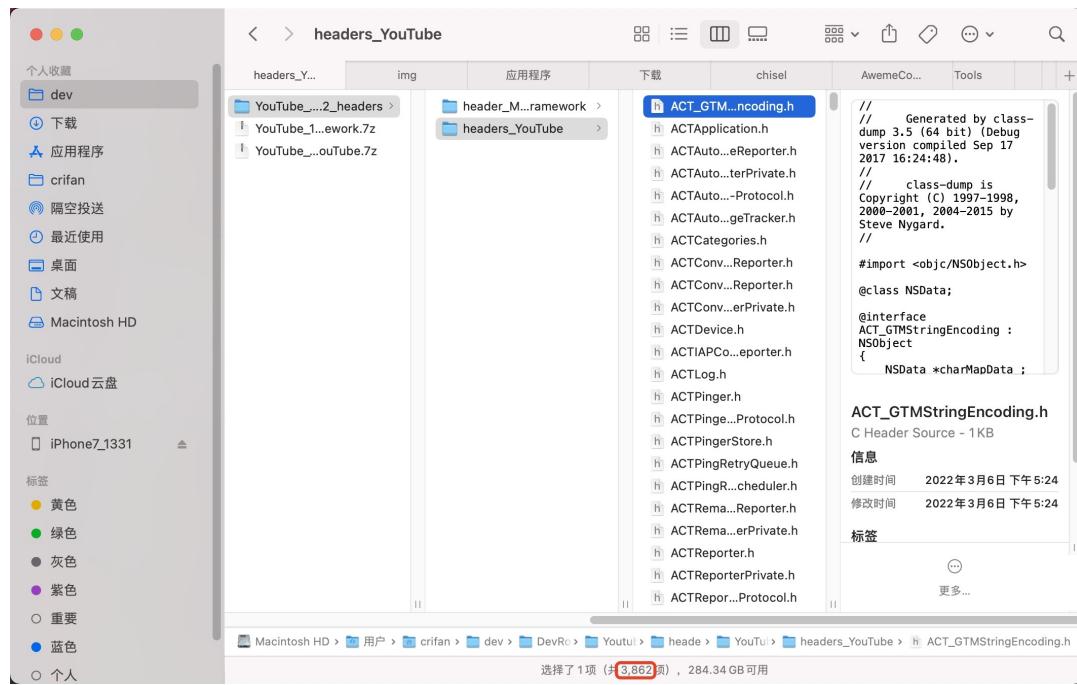
- 主要分两类

- 静态分析
- 动态调试

- 下面分别详细解释：

- 静态分析

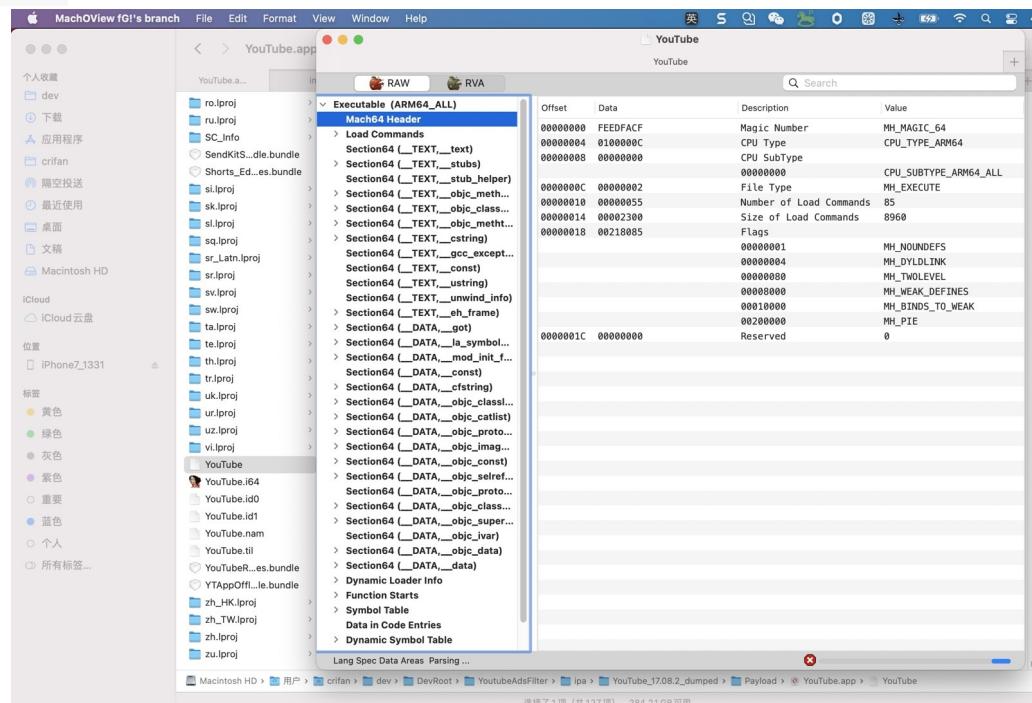
- 从脱壳ipa解压得到app包
  - 用class-dump导出头文件



### ■ 以搞懂包含哪些类

### ■ 查看二进制信息

#### ■ MachOView

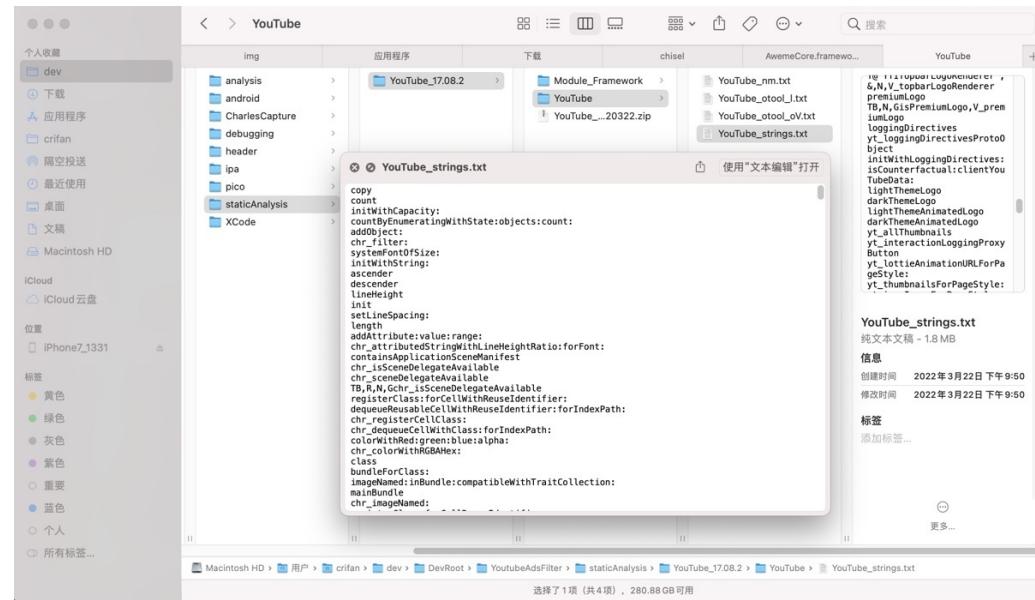


#### ■ jtool2

#### ■ rabin2

### ■ 导出字符串等资源

#### ■ strings



## ■ nm

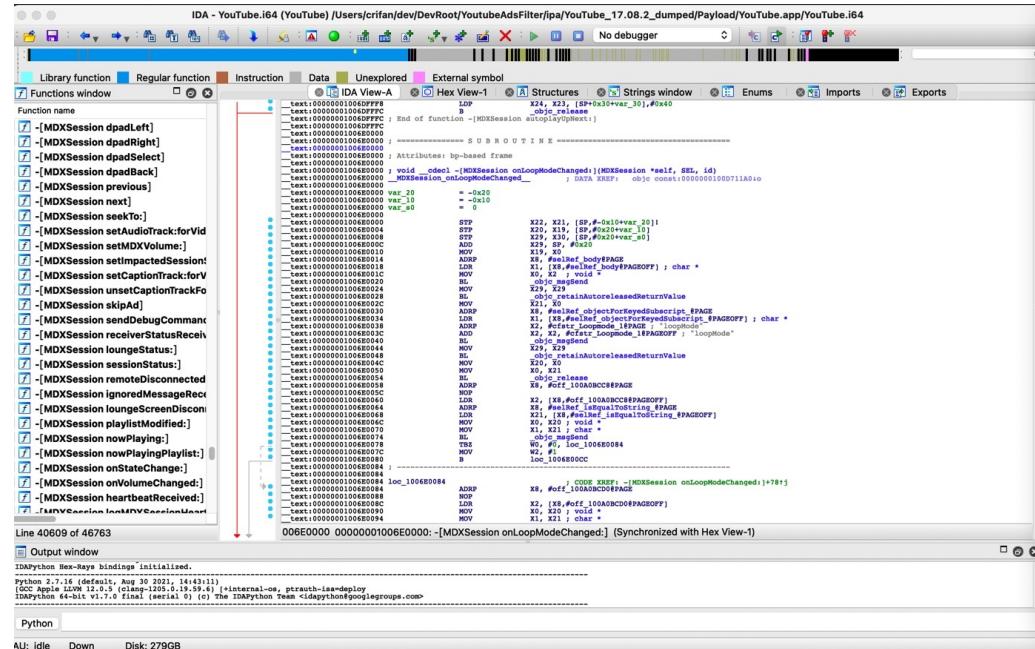
- otool

- jtool2

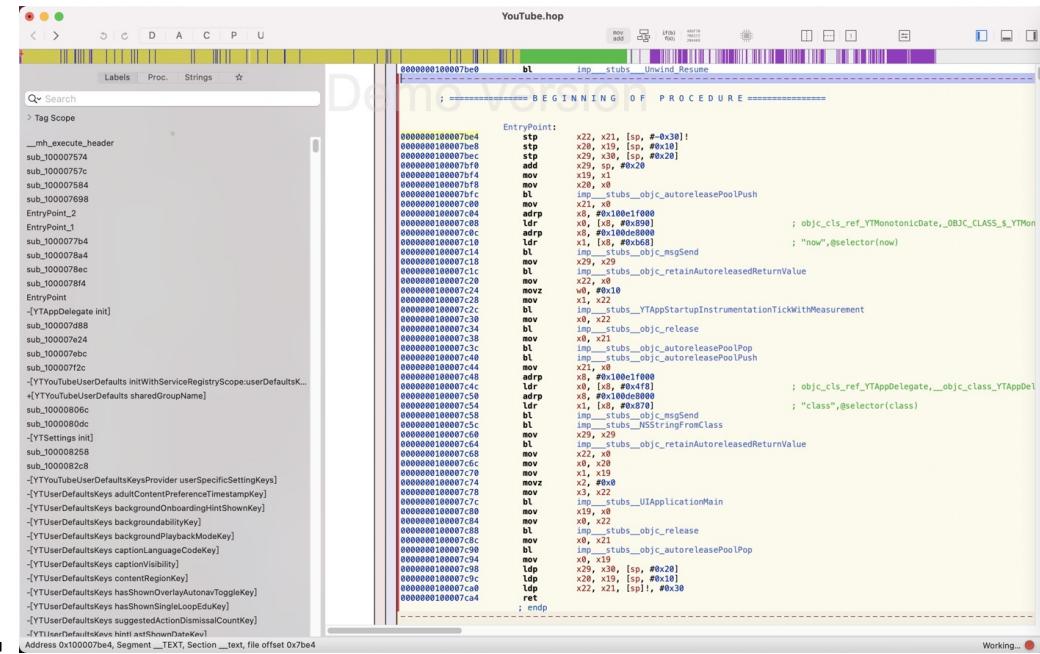
- rabin2

## ■ 分析代码逻辑

- IDA: Functions、Strings、Imports、F5伪代码等好用的功能模块



## ■ Hopper

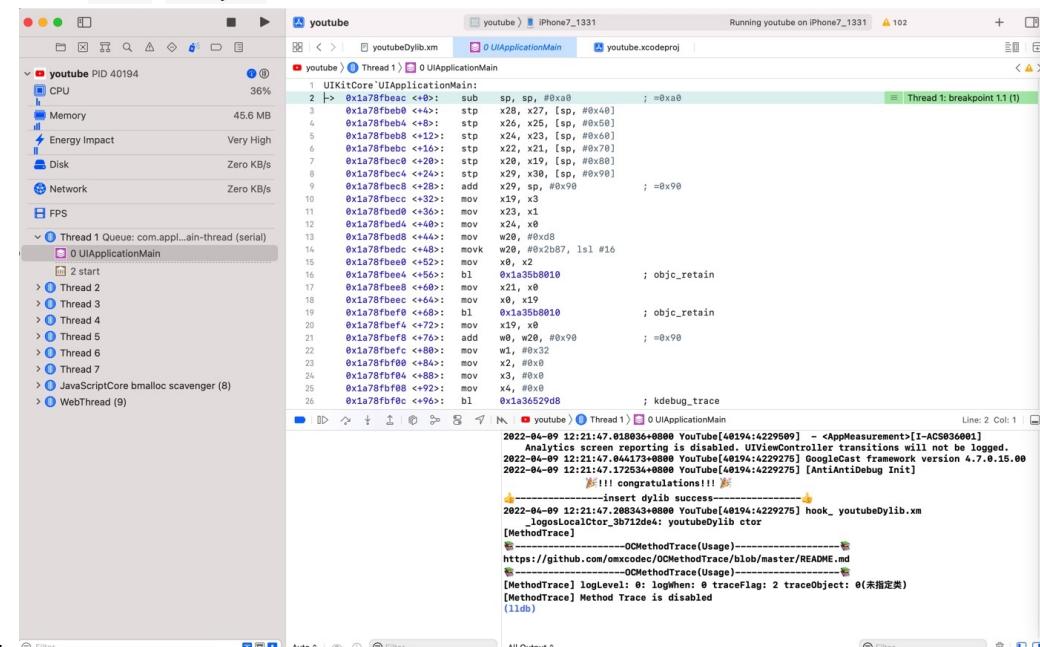


## ● 动态调试

### ○ 用各种调试方式和工具去调试app的逻辑

#### ■ 常用调试方式

##### ■ 图形界面：XCode + MonkeyDev



##### ■ XCode内部也有 lldb

##### ■ 命令行：debugserver + lldb

```

X root@10.0.0.58 (ssh)
1 +0.000000 sec [27ac/0303]: RNBRRunLoopAttaching Attaching to pid 10146...
2 +0.003597 sec [27ac/0303]: MachProcess::MachProcess()
3 +0.000022 sec [27ac/0303]: (DebugHub) attaching to pid 10146...
4 +0.000014 sec [27ac/0303]: MachProcess::SetState(Unloaded) ignoring redundant state
change...
5 +0.000023 sec [27ac/0303]: MachProcess::SetState(Attaching) updating state (previous
state was Unloaded), event_mask = 0x00000001
6 +0.000023 sec [27ac/0303]: MachTask::StartExceptionThread()
7 +0.000035 sec [27ac/0303]: ::task_for_pid (target_ptport = 0x0105, pid = 10146, &ta
k) => err = 0x00000000 (success) err = 0x00000000
8 +0.000029 sec [27ac/0303]: ::task_info (target_task = 0x1407, flavor = TASK_BASIC_I
NFO, task_info_out >= 0x1d855f8, task_info_outCnt >= 10 ) err = 0x00000000
9 +0.000021 sec [27ac/0303]: task_basic_info (i suspend_count = 0, virtual_size = 0x1
51a2c000, resident_size = 0x1708000, user_time = 0x64964, system_time = 4.64964 )
10 +0.000028 sec [27ac/0303]: MachException::PortInfo::Sove (task = 0x1407)
11 +0.000025 sec [27ac/0303]: ::task_get_exception_ports (task = 0x1407, mask = 0x1bf
e, maskOut >= 3, ports, behaviors, flavors) err = 0x00000000
12 +0.000026 sec [27ac/0303]: ::task_set_exception_ports (task = 0x1407, exception_ma
sk = 0x000001fe, new_port = 0x2903, behavior = 0x00000001, new_flavor = 0x00000005 ) e
rr = 0x00000000
13 +0.000023 sec [27ac/1703]: MachTask::ExceptionThread (arg = 0x10400668) starting
thread...
iPhone7P:1341:~/forDebug root# pwd
/var/root/forDebug
iPhone7P:1341:~/forDebug dbgserver -x auto 0.0.0.0:20221 /private/var/contai
ners/bundle/Application/9AB25481-0A03-435C-A02E-68F96235358B/Aweme.app/Aweme
debugserver -e #PROGRAM LLDB PROJECT:lldb-900.3.104
for arm64.
Listening to port 20221 for a connection from 0.0.0.0...
Got a connection, launched process /private/var/containers/Bundle/Application/9AB25481
-0A03-435C-A02E-68F96235358B/Aweme.app (pid = 10174).
Exiting.
iPhone7P:1341:~/forDebug root# []

```

## ■ 常用逆向工具

### ■ Frida : hook对应函数，调试输入参数和返回值

```

* ~ frida -U -f com.ss.iphone.ugc.Aweme
  _ _ _ | Frida 15.1.14 - A world-class dynamic instrumentation toolkit
| C_ | |
> _ | Commands:
/_\_| help      -> Displays the help system
... . object?    -> Display information about 'object'
... . exit/quit   -> Exit
... . More info at https://frida.re/docs/home/
Spawneed com.ss.iphone.ugc.Aweme . Use %resume to let the main thread start executing!
stdout> objc[20744]: Class PodsDummy_EffectPlatformSDK is implemented in both /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework/byteaudio (0x104f7f90) and /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.framework/VoLcEngineRTC (0x11827cb10). One of the two will b
e used. Which one is undefined.
stdout> objc[20744]: Class IESAlgorithmModelCleaner is implemented in both /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework/byteaudio (0x104f7f80) and /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.framework/VoLcEngineRTC (0x11827cb8). One of the two will be u
sed. Which one is undefined.
stdout> objc[20744]: Class IESAlgorithmModelConfig is implemented in both /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework/byteaudio (0x104f7800) and /private/va
r/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.framework/VoLc
stdout> objc[20744]: Class EngineRTC (0x11827cb8). One of the two will be used. Which one is undefined.
stdout> objc[20744]: Class IESAlgorithmModelDownloadQueue is implemented in both /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-4-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework/byteaudio (0x104f7805) and /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.framework/VoLcEngineRTC (0x11827cd8). One of the two will be used. Which one is undefined.
stdout> ob
stdout> jc[20744]: Class IESAlgorithmModelDownloadTask is implemented in both /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework/byteaudio (0x104f7808) and /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.framework/VoL
stdout> ation[20744]: Class IESAlgorithmModelManager is implemented in both /private/var/conta
[ iPhone:com.ss.iphone.ugc.Aweme ]> stdout> iners/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/byteaudio.framework
/byteaudio (0x104f7808) and /private/var/containers/Bundle/Application/0C35E4DD-5B6B-40DE-BB14-1F1EE9266003/Aweme.app/Frameworks/VoLcEngineRTC.fr

```

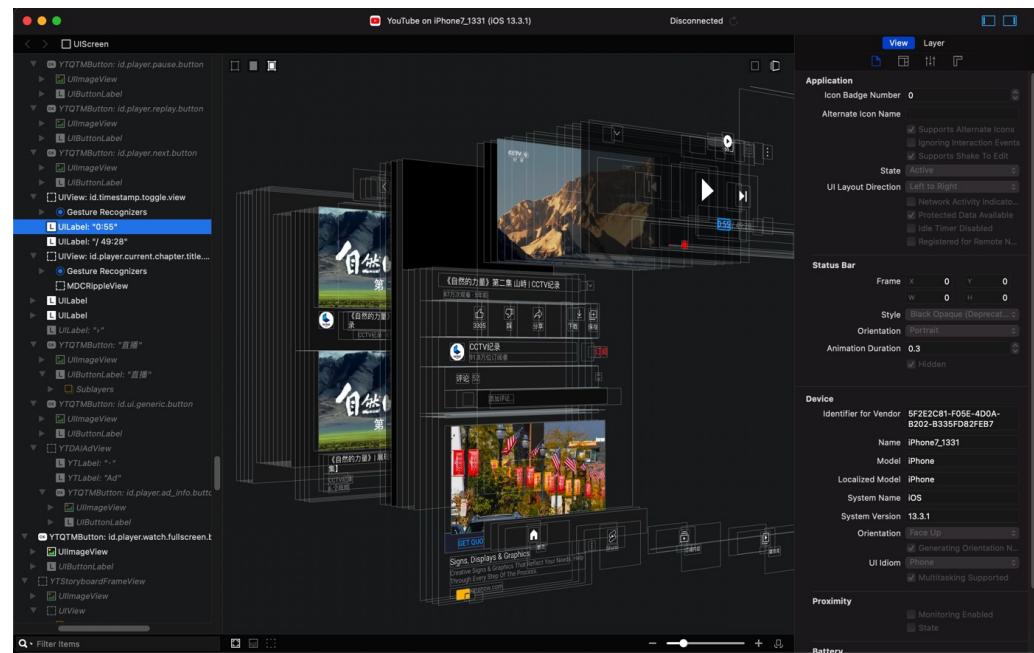
### ■ Cycript : 查看页面元素的类和属性、实时修改各种值

```

cycript -r168.192.1.12:6666
..s/2.6/usr/lib (-zsh)
cy# APPID
com.google.ios.youtube
cy# pviewsOpvcsoyc
cy# pvcso
<YTAppViewController 0x10e02f400>, state: appeared, view: <YTAppView 0x10c3d2000>
  <YIScrollIndicatorNavigationController 0x10e06000>, state: disappeared, view: <UILayoutContainerView 0x10e3cf180> not in the window
  | <YTHeaderContentComboViewController 0x11312ce00>, state: disappeared, view: <YTHeaderContentComboView 0x10e3cf180> not in the window
  | <YTBrowserViewController 0x11d48000>, state: disappeared, view: <YTBrowserView 0x10e05000> not in the window
  | <YTRowseResponseViewController 0x10e901200>, state: disappeared, view: <YTVariableHeightHeaderView 0x10e6bd020> not in the window
  | <YTTabViewController 0x113156170>, state: disappeared, view: <YTTabView 0x113156170> not in the window
  | <YTPageViewController 0x1131784c0>, state: disappeared, view: <YTPageView 0x10f251a00> not in the window
  | <YTPageViewController 0x1131784c0>, state: disappeared, view: <YTPageView 0x10f251a00> not in the window
  | <YTSectionListListViewController 0x1131582d0>, state: disappeared, view: <YTVariableHeightHeaderView 0x10e3c6240> not in the window
  | <YTSectionListListViewController 0x1131582d0>, state: disappeared, view: <YTVariableHeightHeaderView 0x1131582d0> not in the window
  | <YTVariableHeightHeaderViewController 0x11315d000>, state: disappeared, view: <YTVariableHeightHeaderView 0x11315d000> not in the window
  | <YTAppCollectionViewController 0x10f2b6000>, state: disappeared, view: <YTAsyncCollectionView 0x10f307000> not in the window
  | <YTHeaderViewController 0x10f2a0100>, state: disappeared, view: <YTHeaderView 0x11312e040> not in the window
  | <YTHeaderContentComboViewController 0x121426030>, state: disappeared, view: <YTHeaderContentComboView 0x121426030>
  | <YTSearchResultsViewController 0x10f570000>, state: disappeared, view: <YTHeaderView 0x121426030>
  | <YTSearchResponseViewController 0x1214242d0>, state: disappeared, view: <YTHeaderView 0x12141c7d0>
  | <YTSectionListListViewController 0x111fc6500>, state: disappeared, view: <YTView 0x1214360d0>
  | <YTVariableHeightHeaderViewController 0x11f1c60b20>, state: disappeared, view: <YTVariableHeightHeaderView 0x1131e5950>
  | <YTAppCollectionViewController 0x10e9b6f000>, state: disappeared, view: <YTAsyncCollectionView 0x10ecc3e00>
  | <YTHeaderViewController 0x10ec0e400>, state: disappeared, view: <YTHeaderView 0x121407b80>
<YTPlotBarViewController 0x10e0e7800>, state: appeared, view: <YTPlotBarView 0x10e0ed70>
<YTWatchLayerViewController 0x10f23d000>, state: appeared, view: <YTWatchLayerView 0x10e3d6970>
<YTWatchMiniBarViewController 0x10e951000>, state: appeared, view: <YTWatchMiniBarView 0x10e3d7390>
<YTWatchViewController 0x10e0e3d000>, state: appeared, view: <YTWatchSingleItemView 0x12557f70>
<YTEngagementPanelMultipleStacksContainerViewController 0x125541c00>, state: appeared, view: <YTEngagementPanelMultipleStacksContainerView 0x12559a850>
<YTPlayerViewController 0x10f635200>, state: appeared, view: <YTPlayerView 0x12554c00>
  | <YTMainAppVideoPlayerOverlayViewController 0x10f349800>, state: appeared, view: <YTMainAppVideoPlayerOverlayView 0x12554fe00>
  | <YTOverflowMenuViewController 0x125573460>, state: appeared, view: <YTOverflowMenuView 0x11317dd20>
  | <YTVCaptionOverlayViewController 0x127573300>, state: appeared, view: <YTVCaptionOverlayView 0x113139d90>
  | <YTCreatorEndscreenViewController 0x12719e10>, state: disappeared, view: (View not loaded)
  | <YTInfoCardDrawerViewController 0x1271d040>, state: disappeared, view: (View not loaded)
  | <YTInfoCardTeaserViewController 0x12758c00>, state: appeared, view: <YTInfoCardTeaserContainerView 0x1271d1310>
  | <YTWatchNextViewController 0x12573830>, state: appeared, view: <YTHeaderView 0x12573830>
  | <YTWatchNextResponseViewController 0x125739270>, state: appeared, view: <YTWatchNextView 0x125739080>
  | <YTWatchNextViewController 0x10fbca800>, state: appeared, view: <YTAsyncCollectionView 0x10ef07000>
  | <YTInfoCardDrawerViewController 0x113148f50>, state: appeared, view: <YTInfoCardDrawerView 0x11d3c0060>
<MDXViewController 0x11311d990>, state: appeared, view: <MDXView 0x113117780>

```

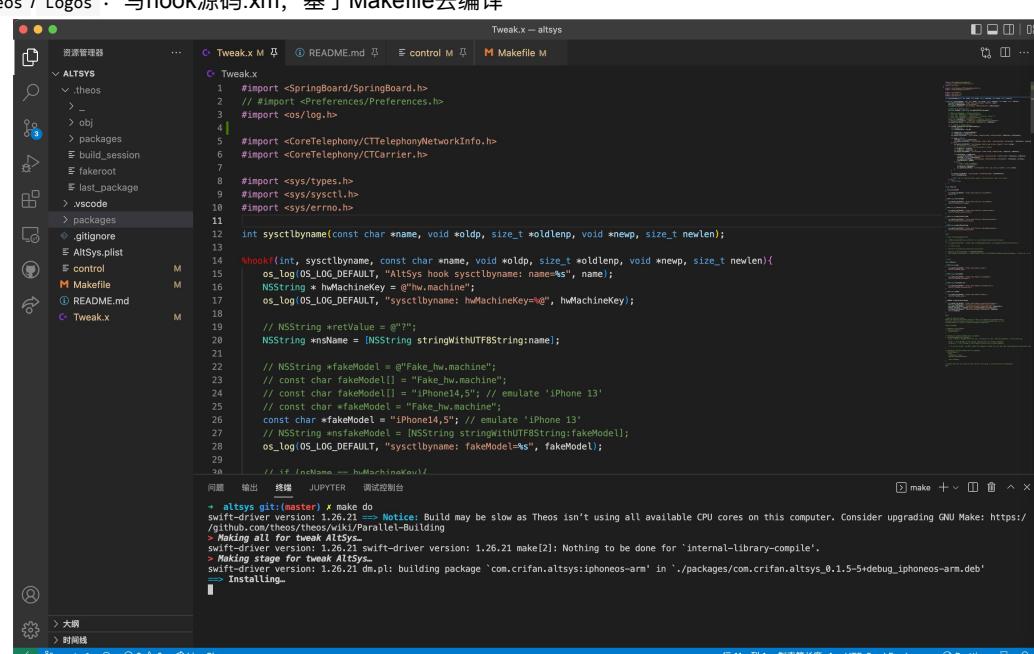
### ■ Reveal : 查看UI页面详细属性



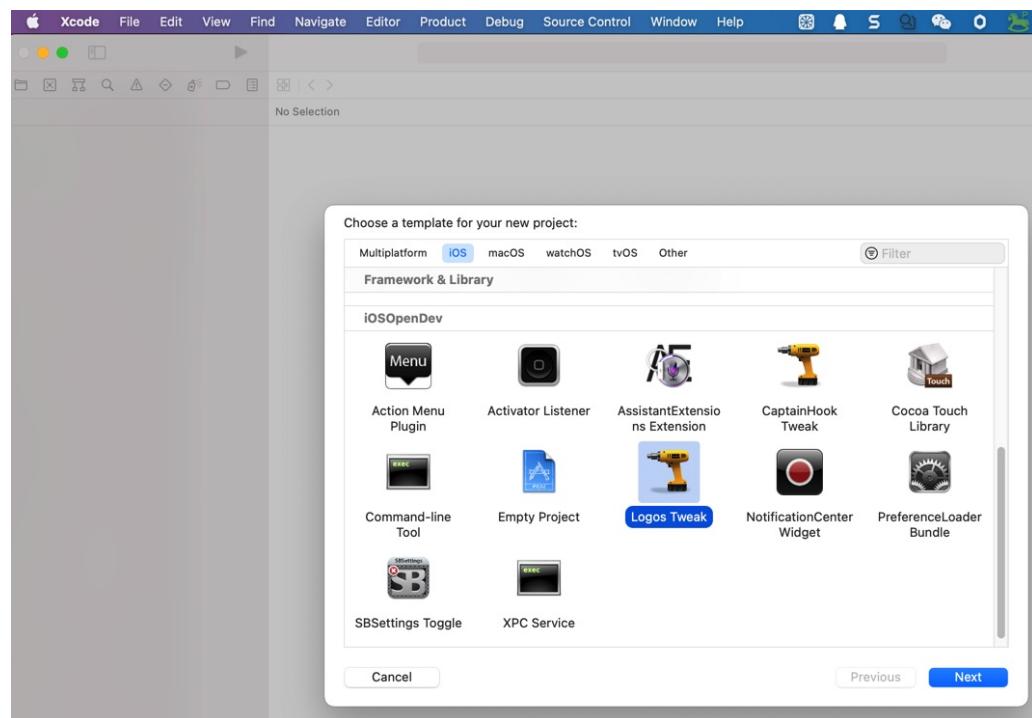
- 目的：搞懂我们所关心的app内部相关逻辑
    - 用于后续去写hook代码去修改成我们要的逻辑和值
  - 最后才是：Tweak插件开发
    - 常见插件开发框架
      - 基于 CydiaSubstrate
        - Theos / Logos
        - fishhook

### ○ 具体开发方式

- ## ■ 命令行

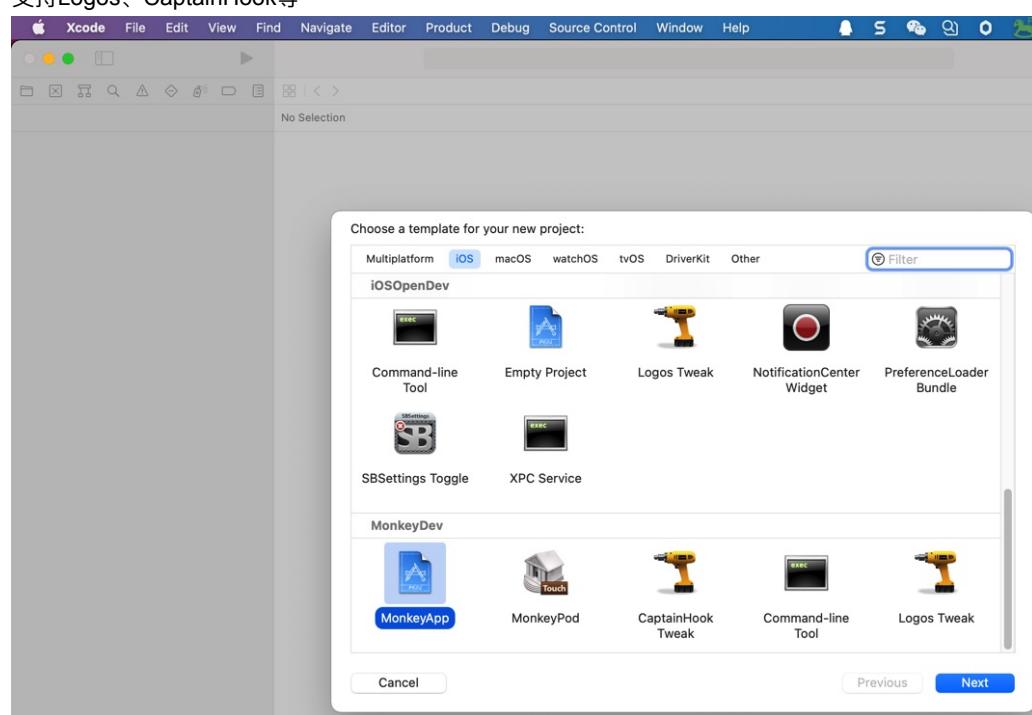


- 图形界面
    - iOSOpenDev：把Logos等开发集成进了XCode
    - 支持Logos、CaptainHook等



- MonkeyDev : iOSOpenDev的改进版

- 支持Logos、CaptainHook等



```

youtube
youtubeDylib.xm
311     }
312 }
313     return iwu_umet;
314 }
315 }
316 }end
317 /*
318 NSHTTPURLResponse
319 */
320 -----
321 //hook NSHTTPURLResponse
322 -(NSHTTPURLResponse*)initWithURL:(NSURL *)url
323 {
324     self = [super init];
325     if (self) {
326         statusCode=(NSInteger)statusCode;
327         HTTPVersion:(NSString *)HTTPVersion;
328         headerFields:(NSDictionary<NSString *, NSString *> *)headerFields;
329         NSHTTPURLResponse* iwu_ushh = [orig
330 //    if (isYoutubeAdsVideo(url))
331 //        iosLogInfo("is ads video: url=%@,statusCode=%d,HTTPVersion=%@ -> iwu_ushh=%@", url, statusCode,
332 //                   HTTPVersion, headerFields, iwu_ushh);
333 //    if (isYoutubeAdsVideo_ctierA(url))
334 //        iosLogInfo("is ctierA ads video: url=%@,statusCode=%d,HTTPVersion=%@ -> iwu_ushh=%@", url,
335 //                   statusCode, HTTPVersion, headerFields, iwu_ushh);
336 //}
337     return iwu_ushh;
338 }
339
340 }end

```

Q Find v reload  
x0 = 0x00000000282d4cdc8  
(lldb) po 0x00000000282d4cdc8  
10784916984  
(lldb) reg r w0  
w0 = 0x00000000c8  
(lldb) p/w 0x00000000c8  
(int) \$10 = 200

- 去开发插件=写hook代码

- 前提：通过静态分析的头文件和动态调试，已知的app内部的类的属性和函数
- 核心逻辑：去hook对应类的函数和属性
  - 实现对应的效果，比如：
    - 调试：输出函数的输入参数和输出结果
    - 修改逻辑：屏蔽原先逻辑，重写自己想要的逻辑

- 开发出的插件常用于

- 逆向破解特定app
  - 绕过ssl证书校验，实现Charles抓包https可看到明文数据
  - 修改app原有逻辑，实现特定的功能
    - 支付宝：修改显示的余额
    - 微信：抢红包
    - 抖音：点赞关注
- 反反调试
- 反越狱检测

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：2022-09-19 08:51:58

# iOS典型逆向开发流程

典型的iOS逆向开发的流程是：

- 先：iPhone越狱
  - 越狱工具
    - unc0ver
    - checkra1n
- 再：逆向破解某iOS的app
  - 对于要研究的某个iOS的app来说，从工作内容角度，主要分：
    - 从AppStore中搜索和安装正版app
    - 去砸壳得到ipa
      - frida-ios-dump
    - 再去研究
      - 静态分析
        - 字符串分析
          - strings
          - nm
          - otool
        - 查看详情
          - MachOView
          - jtool2
          - rabin2
        - 导出头文件
          - class-dump
        - 分析代码逻辑
          - IDA
            - 汇编代码
            - 伪代码
      - 动态调试
        - 命令行方式
          - debugserver + lldb
        - GUI图形界面方式
          - XCode + MonkeyDev
            - lldb
      - 写hook插件
        - 再去调试和验证
          - 如此反复
    - 最后结果：得到一个插件tweak，实现了你要的目的
      - 比如反越狱检测、改机、hook某个app的某些特定功能（比如微信抢红包）等等

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：2022-06-16 09:39:58

## iOS逆向的重点和难点

对于iOS逆向开发涉及到众多方面的内容，其中属于重点和难点的是：

- 先搞懂你想要干啥
  - 比如 微信抢红包
- 重点和难点
  - 用静态分析工具和动态调试，共同配合，找出要hook的类
    - 如何用好各种静态分析工具（`otool`、`MachOView`、`IDA Pro`、`Hopper`等），如何进行动态调试（`Xcode + MonkeyDev`、`debugserver + lldb`、`frida`、`Cycript`、`Reveal`等），从而找到要分析的类和代码运行逻辑等等，才是重点和难点
    - 再去写hook代码，开发出插件，实现对应的功能

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：2022-10-20 15:15:03

## 子教程

上述所有和iOS逆向开发相关的内容，分别整理到多个独立的子教程以及相关内容：

- 系列

- 【整理Book】iOS逆向开发：逆向开发流程
- 【整理Book】iOS逆向开发：iPhone越狱
  - 包括iPhone管理：插件的安装和卸载、文件管理等
- 【整理Book】iOS逆向开发：越狱插件开发
- 【整理Book】iOS逆向开发：越狱检测和反越狱检测
- 【整理Book】iOS逆向开发：抖音逆向

- 相关

- 【整理Book】iOS逆向心得：Apple苹果相关开发资料 20220527
- 【整理Book】iOS开发心得
- 【整理Book】ARM汇编基础知识 20220513
- 【整理Book】asm汇编开发心得
- 【整理Book】XCode中lldb 调试心得 开发心得
- 【整理Book】XCode经验总结：开发心得之调试心得
- 【整理】XCode经验总结：开发心得之调试心得
- 【整理Book】IDA Pro使用心得 20220602
- 【整理Book】iOS逆向心得：Block的相关基础知识

# iOS逆向心得

## 不同app的逆向破解出的难度不同

TODO:

- 【整理】iOS逆向心得：不同app的逆向破解出的难度不同

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：2022-04-09 11:19:01

## 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

- [crifan \(Crifan Li\)](#)

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: 2022-04-09 11:20:20