

目录

| | |
|----------------|-------|
| 前言 | 1.1 |
| iOSOpenDev概览 | 1.2 |
| 安装iOSOpenDev | 1.3 |
| 常见问题 | 1.3.1 |
| 确认安装成功 | 1.3.2 |
| 普通的插件开发流程 | 1.4 |
| 新建iOSOpenDev项目 | 1.4.1 |
| 初始化项目配置 | 1.4.2 |
| 写hook插件代码 | 1.4.3 |
| 调试插件代码 | 1.4.4 |
| 带界面的插件开发流程 | 1.5 |
| 心得 | 1.6 |
| 附录 | 1.7 |
| 参考资料 | 1.7.1 |

iOS逆向开发：iOSOpenDev开发插件

- 最新版本： v0.8
- 更新时间： 20221108

简介

介绍iOS逆向中如何用iOSOpenDev开发越狱插件tweak。先是*对iOSOpenDev概览*；然后介绍如何安装*iOSOpenDev*，涉及到常见问题和安装后确认安装成功；然后是普通的插件的开发流程，包括新建*iOSOpenDev*的Xcode项目、初始化项目的配置、写hook插件tweak代码、编译代码调试代码等；以及带UI界面的插件开发的流程；以及一些心得。

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/ios_re_iosopendev_tweak: iOS逆向开发：iOSOpenDev开发插件](#)

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

离线下载阅读

- [iOS逆向开发：iOSOpenDev开发插件 PDF](#)
- [iOS逆向开发：iOSOpenDev开发插件 ePUB](#)
- [iOS逆向开发：iOSOpenDev开发插件 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：
2022-11-08 17:11:51

iOSOpenDev概览

TODO:

- 【整理】iOS越狱插件开发工具：iOSOpenDev
-

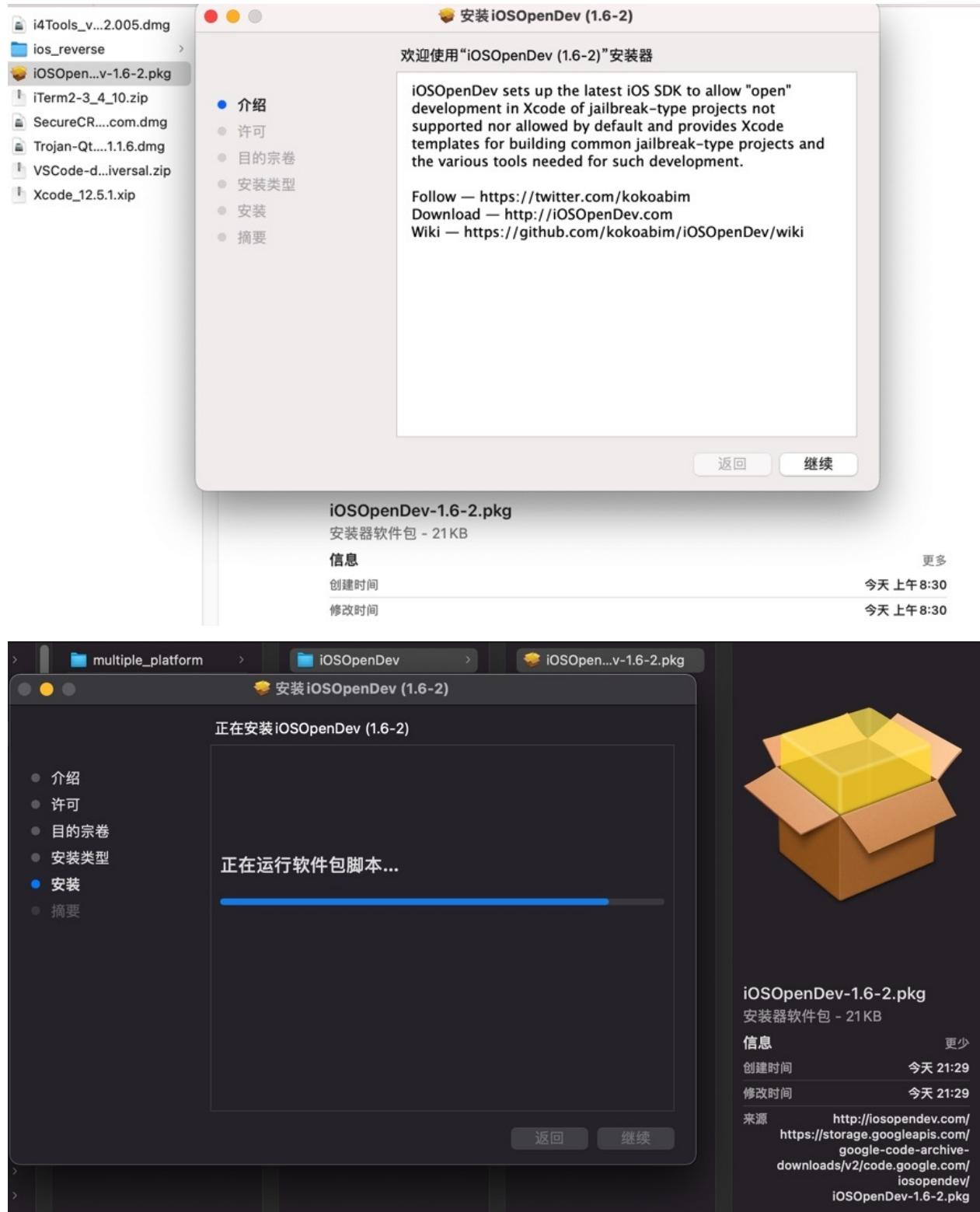
在iOS逆向期间，往往涉及到去[开发越狱插件tweak](#)，其中常见的工具=框架之一就是：`iOSOpenDev`。

- `iOSOpenDev`
 - 官网
 - <http://iosopendev.com/>
 - Github
 - <https://github.com/kokoabim/iOSOpenDev>
 - Wiki
 - <https://github.com/kokoabim/iOSOpenDev/wiki>

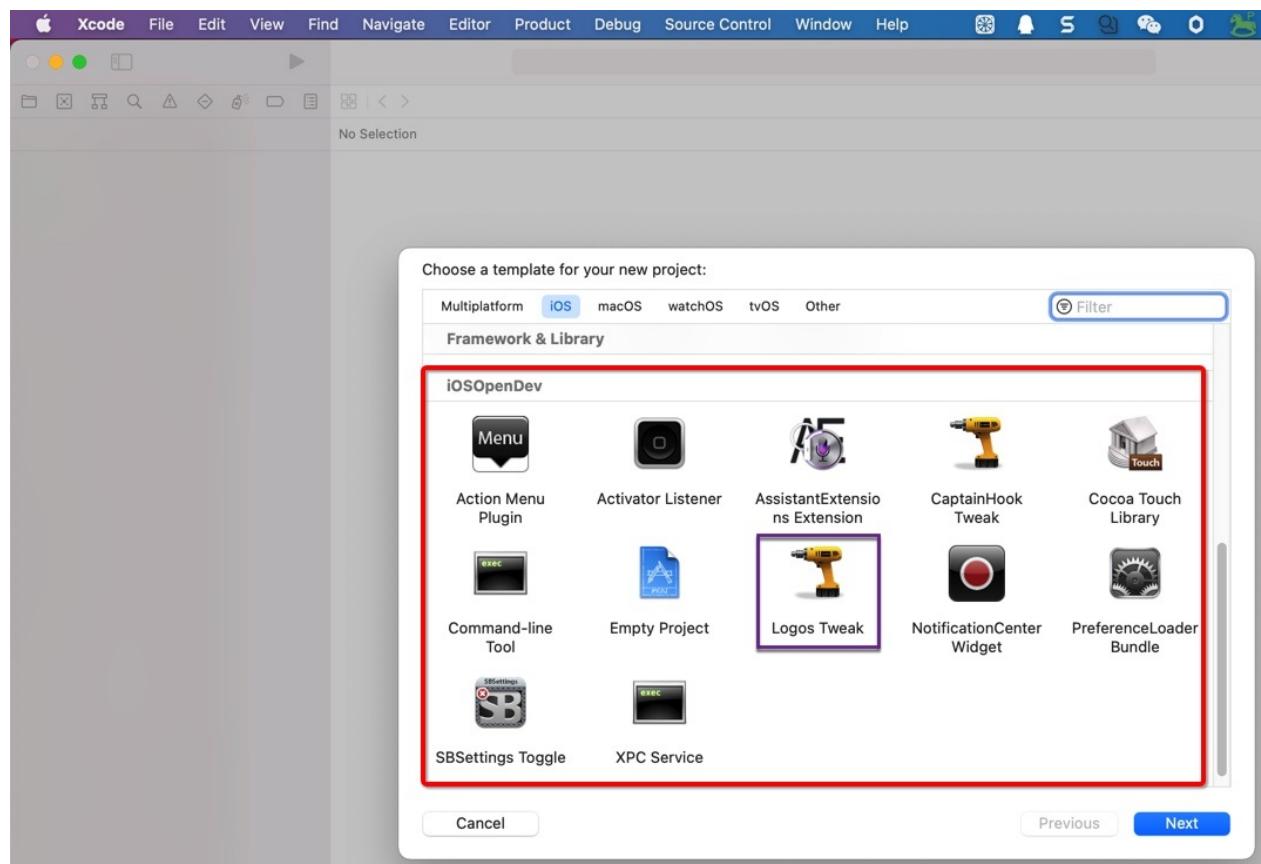
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-11-08 14:29:06

安装iOSOpenDev

从官网[iOSOpenDev—Download](#)下载到: [iOSOpenDev-1.6-2.pkg](#), 双击去安装:



成功安装后, 去 Xcode 中新建 iOS 项目, 即可看到 iOSOpenDev 的选项:



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 16:59:45

常见问题

安装器遇到了一个错误，导致安装失败

安装到最后，报错： 安装失败 安装器遇到了一个错误，导致安装失败



解决办法：

其实此时 iosopenDev 的主体文件已安装到了默认的位置 /opt 中，接着去用工具初始化即可解决问题：

```
cd /opt/iOSOpenDevSetup/bin  
sudo ./iod-setup base  
sudo ./iod-setup sdk -sdk iphoneos
```

PrivateFramework directory not found XCode iPhoneOS15.0.sdk

iod-setup sdk -sdk iphoneos 时报错：

```
→ bin sudo ./iod-setup sdk -sdk iphoneos  
Setting up iPhoneOS 15.0 SDK...  
Modifying SDK settings...  
Symlinking to private frameworks header files...  
PrivateFramework directory not found: /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS15.0.sdk/System/Library/PrivateFrameworks
```

原因：

此处是比较新的 XCode 13 和对应的 iOS 15

->而最新版XCode和iOS早已将私有库PrivateFrameworks移走了

->即 iPhoneOSxx.xx.sdk/System/Library/ 下面没有 PrivateFrameworks 了

解决办法：

- 自己之后是否用到私有库PrivateFrameworks
 - 否
 - 直接新建一个空目录即可

```
cd /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS15.0.sdk/System/Library  
sudo mkdir PrivateFrameworks
```

- 是
 - 除了新建目录外，还要把相关iPhoneOS版本的私有库的内容放过去
 - 先要找到相关iPhoneOS的PrivateFrameworks
 - 举例
 - iPhoneOS 9.2 的 sdk，可以从这里下载到：
 - zhangkn/knPrivateFrameworks:
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform
/Developer/SDKs/iPhoneOS9.2.sdk/System/Library/PrivateFrameworks
(github.com)

File not found XCode Specifications iPhoneOSPackageTypes.xcspec

iod-setup sdk -sdk iphoneos 报错：

```
→ bin sudo ./iod-setup sdk -sdk iphoneos  
Password:  
Setting up iPhoneOS 15.0 SDK...  
Modifying SDK settings...  
Symlinking to private frameworks header files...  
Adding specifications to platform...  
File not found: /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/Specifications/iPhoneOSPackageTypes.xcspec
```

原因：

找不到specifications

解决办法：

下载别人给的：

- 4个iPhoneOS的spec文件

- 4个iPhoneSimulator的spec文件

分别放到对应位置，即可。

下载来源：

- 来源1：
 - [iosopendev专用Specifications.zip](#)
- 来源2：
 - [越狱开发:用iosOpenDev配置越狱开发环境 编写第一个hello world_我的杯洗具的博客-CSDN博客](#)

下载后，可以看到Specifications中有8个spec。

分别新建Specifications目录：

```
sudo mkdir /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/Specifications  
sudo mkdir /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/Library/Xcode/Specifications
```

再去

- 移动文件
 - 把
 - 4个 iPhoneos 的文件
 - iPhoneOSPackageTypes.xcspec
 - iPhoneOSPackageTypes.xcspec.iOSOpenDev
 - iPhoneOSProductTypes.xcspec
 - iPhoneOSProductTypes.xcspec.iOSOpenDev
 - 放到：
 - /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/Specifications
 - 把：
 - 4个 iPhoneSimulator 的文件
 - iPhone Simulator PackageTypes.xcspec
 - iPhone Simulator PackageTypes.xcspec.iOSOpenDev
 - iPhone Simulator ProductTypes.xcspec
 - iPhone Simulator ProductTypes.xcspec.iOSOpenDev
 - 放到：
 - /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/Library/Xcode/Specifications

放好后是：

```
→ Xcode 11 /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/Specifications  
total 48  
-rwxr-xr-x@ 1 crifan wheel 3.2K 12 24 2015 iPhoneOSPackageTypes.xcspec  
-rwxr-xr-x@ 1 crifan wheel 5.4K 12 24 2015 iPhoneOSPackageTypes.xcspec.iOSOpenDev  
-rwxr-xr-x@ 1 crifan wheel 4.0K 12 24 2015 iPhoneOSProductTypes.xcspec  
-rwxr-xr-x@ 1 crifan wheel 6.4K 12 24 2015 iPhoneOSProductTypes.xcspec.iOSOpenDev
```

```
→ Xcode 11 /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/Library/Xcode/Specifications
total 48
-rwxr-xr-x@ 1 crifan wheel  3.4K 12 24  2015 iPhone Simulator PackageTypes.xcspec
-rwxr-xr-x@ 1 crifan wheel  6.9K 12 24  2015 iPhone Simulator PackageTypes.xcspec.iOSOpenDev
-rwxr-xr-x@ 1 crifan wheel  3.4K 12 24  2015 iPhone Simulator ProductTypes.xcspec
-rwxr-xr-x@ 1 crifan wheel  6.1K 12 24  2015 iPhone Simulator ProductTypes.xcspec.iOSOpenDev
```

另外，新建usr的bin目录：

```
sudo mkdir /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/usr/bin
```

即可。

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 17:02:11

确认安装成功

环境变量

再去确认，是否把iOSOpenDev的相关环境变量，加到启动脚本（此处是 zsh，所以是 ./zshrc）中了：

```
→ ~ cat ~/.zshrc | grep iOSOpenDev
export iOSOpenDevPath=/opt/iOSOpenDev
export iOSOpenDevDevice=
export PATH /opt/iOSOpenDev/bin:$PATH
```

如果没有：

```
x crifan@licrifandeMacBook-Pro /opt/iOSOpenDevSetup/bin    cat ~/.zshrc | grep iOSOpenDev
```

则自己手动去加上：

```
crifan@licrifandeMacBook-Pro /opt/iOSOpenDevSetup/bin    vi ~/.zshrc
crifan@licrifandeMacBook-Pro /opt/iOSOpenDevSetup/bin    cat ~/.zshrc | grep iOSOpenDev
export iOSOpenDevPath=/opt/iOSOpenDev
export iOSOpenDevDevice=
export PATH /opt/iOSOpenDev/bin:$PATH
crifan@licrifandeMacBook-Pro /opt/iOSOpenDevSetup/bin    source ~/.zshrc
```

Xcode中的iOSOpenDev的模板

确认是否有多出的template模板：

```
→ ~ ll ~/Library/Developer/Xcode/
total 0
drwxr-xr-x  8 crifan  staff   256B 10 14 11:13 DerivedData
srwrxr-xr-x  1 crifan  staff     0B 10 27 08:54 GPUToolsAgent.sock
drwxr-xr-x  3 crifan  staff   96B 10 27 08:49 Templates
drwxr-xr-x  6 crifan  staff  192B 10 26 22:37 UserData
drwxr-xr-x  5 crifan  staff  160B  9 30 22:11 iOS Device Logs
drwxr-xr-x  4 crifan  staff  128B 10 13 13:54 iOS DeviceSupport
→ ~ ll ~/Library/Developer/Xcode/Templates
total 0
lrwxr-xr-x  1 root   staff    25B 10 27 08:49 iOSOpenDev -> /opt/iOSOpenDev/templates
```

此处是有的：

- 多出了软链接：
 - ~/Library/Developer/Xcode/Templates/iOSOpenDev
 - 指向的是：
 - /opt/iOSOpenDev/templates

以及接着去看看，具体有哪些模板：

```
→ ~ ll /opt/iOSOpenDev/templates
total 48
drwxr-xr-x  5 root  wheel  160B 10 27 08:32 Action Menu Plugin.xctemplate
drwxr-xr-x  6 root  wheel  192B 10 27 08:32 Activator Listener.xctemplate
drwxr-xr-x 12 root  wheel  384B 10 27 08:32 AssistantExtensions Extension.xctemplate
drwxr-xr-x  4 root  wheel  128B 10 27 08:32 Base.xctemplate
drwxr-xr-x  6 root  wheel  192B 10 27 08:32 CaptainHook Tweak.xctemplate
drwxr-xr-x  6 root  wheel  192B 10 27 08:32 Cocoa Touch Library.xctemplate
drwxr-xr-x  5 root  wheel  160B 10 27 08:32 Command-line Tool.xctemplate
drwxr-xr-x  4 root  wheel  128B 10 27 08:32 Debian Package.xctemplate
drwxr-xr-x  4 root  wheel  128B 10 27 08:32 Empty Project.xctemplate
-rw-r--r--  1 root  wheel   18K 10 27 08:49 LICENSE
drwxr-xr-x  6 root  wheel  192B 10 27 08:32 Logos Tweak.xctemplate
drwxr-xr-x  5 root  wheel  160B 10 27 08:32 ManPage.xctemplate
drwxr-xr-x 11 root  wheel  352B 10 27 08:32 NotificationCenter Widget.xctemplate
drwxr-xr-x 12 root  wheel  384B 10 27 08:32 PreferenceLoader Bundle.xctemplate
drwxr-xr-x  7 root  wheel  224B 10 27 08:32 PreferenceLoader.xctemplate
-rw-r--r--  1 root  wheel  352B 10 27 08:49 README.md
drwxr-xr-x  5 root  wheel  160B 10 27 08:32 SBSettings Toggle.xctemplate
drwxr-xr-x  4 root  wheel  128B 10 27 08:32 Unit Tests.xctemplate
drwxr-xr-x  7 root  wheel  224B 10 27 08:32 XPC Service.xctemplate
```

很明显，部分模板，应该就是对应着界面中看到的各个模板：

比如：

- Logos Tweak.xctemplate -> Logos Tweak
- Command-line Tool.xctemplate -> Command-line Tool
- PreferenceLoader Bundle.xctemplate -> PreferenceLoader Bundle

iOSOpenDev中的内容

顺带再去看看，当前iOSOpenDev目录中的内容：

```
→ /opt ll
total 0
drwxr-xr-x  9 root  wheel  288B 10 27 08:32 iOSOpenDev
drwxr-xr-x  3 root  wheel   96B 10 27 08:32 iOSOpenDevSetup
drwxr-xr-x  3 root  wheel   96B 10 27 08:32 iOSOpenDevUninstall
→ /opt cd iOSOpenDev
→ iOSOpenDev pwd
/opt/iOSOpenDev
→ iOSOpenDev ll
total 48
-rw-r--r--  1 root  wheel   18K 10 27 08:49 LICENSE
-rw-r--r--  1 root  wheel  352B 10 27 08:49 README.md
drwxr-xr-x  6 root  wheel  192B 10 27 08:32 bin
drwxr-xr-x  2 root  wheel   64B 10 27 08:32 frameworks
drwxr-xr-x  8 root  wheel  256B 10 27 08:32 include
drwxr-xr-x  5 root  wheel  160B 10 27 08:32 lib
drwxr-xr-x 21 root  wheel  672B 10 27 08:32 templates
```

```

→ iOSOpenDev 11 bin
total 3000
-rw xr-xr-x 1 root wheel 428K 10 27 08:49 class-dump
-rw xr-xr-x 1 root wheel 628K 10 27 08:49 class-dump-z
-rw xr-xr-x 1 root wheel 59K 10 27 08:49 iosod
-rw xr-xr-x 1 root wheel 383K 10 27 08:49 ldid
→ iOSOpenDev 11 frameworks
→ iOSOpenDev 11 include
total 48
drwxr-xr-x 3 root wheel 96B 10 27 08:32 ActionMenu
drwxr-xr-x 3 root wheel 96B 10 27 08:32 AssistantExtensions
drwxr-xr-x 3 root wheel 96B 10 27 08:32 CaptainHook
drwxr-xr-x 10 root wheel 320B 10 27 08:32 libactivator
drwxr-xr-x 3 root wheel 96B 10 27 08:32 logos
-rw-r--r-- 1 root wheel 21K 10 27 08:49 substrate.h
→ iOSOpenDev 11 lib
total 1216
-rw xr-xr-x 1 root wheel 77K 10 27 08:49 libactionmenu.dylib
-rw xr-xr-x 1 root wheel 422K 10 27 08:49 libactivator.dylib
-rw xr-xr-x 1 root wheel 101K 10 27 08:49 libsubstrate.dylib
→ iOSOpenDev 11 templates
total 48
drwxr-xr-x 5 root wheel 160B 10 27 08:32 Action Menu Plugin.xctemplate
drwxr-xr-x 6 root wheel 192B 10 27 08:32 Activator Listener.xctemplate
drwxr-xr-x 12 root wheel 384B 10 27 08:32 AssistantExtensions Extension.xctemplate
drwxr-xr-x 4 root wheel 128B 10 27 08:32 Base.xctemplate
drwxr-xr-x 6 root wheel 192B 10 27 08:32 CaptainHook Tweak.xctemplate
drwxr-xr-x 6 root wheel 192B 10 27 08:32 Cocoa Touch Library.xctemplate
drwxr-xr-x 5 root wheel 160B 10 27 08:32 Command-line Tool.xctemplate
drwxr-xr-x 4 root wheel 128B 10 27 08:32 Debian Package.xctemplate
drwxr-xr-x 4 root wheel 128B 10 27 08:32 Empty Project.xctemplate
-rw-r--r-- 1 root wheel 18K 10 27 08:49 LICENSE
drwxr-xr-x 6 root wheel 192B 10 27 08:32 Logos Tweak.xctemplate
drwxr-xr-x 5 root wheel 160B 10 27 08:32 ManPage.xctemplate
drwxr-xr-x 11 root wheel 352B 10 27 08:32 NotificationCenter Widget.xctemplate
drwxr-xr-x 12 root wheel 384B 10 27 08:32 PreferenceLoader Bundle.xctemplate
drwxr-xr-x 7 root wheel 224B 10 27 08:32 PreferenceLoader.xctemplate
-rw-r--r-- 1 root wheel 352B 10 27 08:49 README.md
drwxr-xr-x 5 root wheel 160B 10 27 08:32 SBSettings Toggle.xctemplate
drwxr-xr-x 4 root wheel 128B 10 27 08:32 Unit Tests.xctemplate
drwxr-xr-x 7 root wheel 224B 10 27 08:32 XPC Service.xctemplate

```

普通的插件开发流程

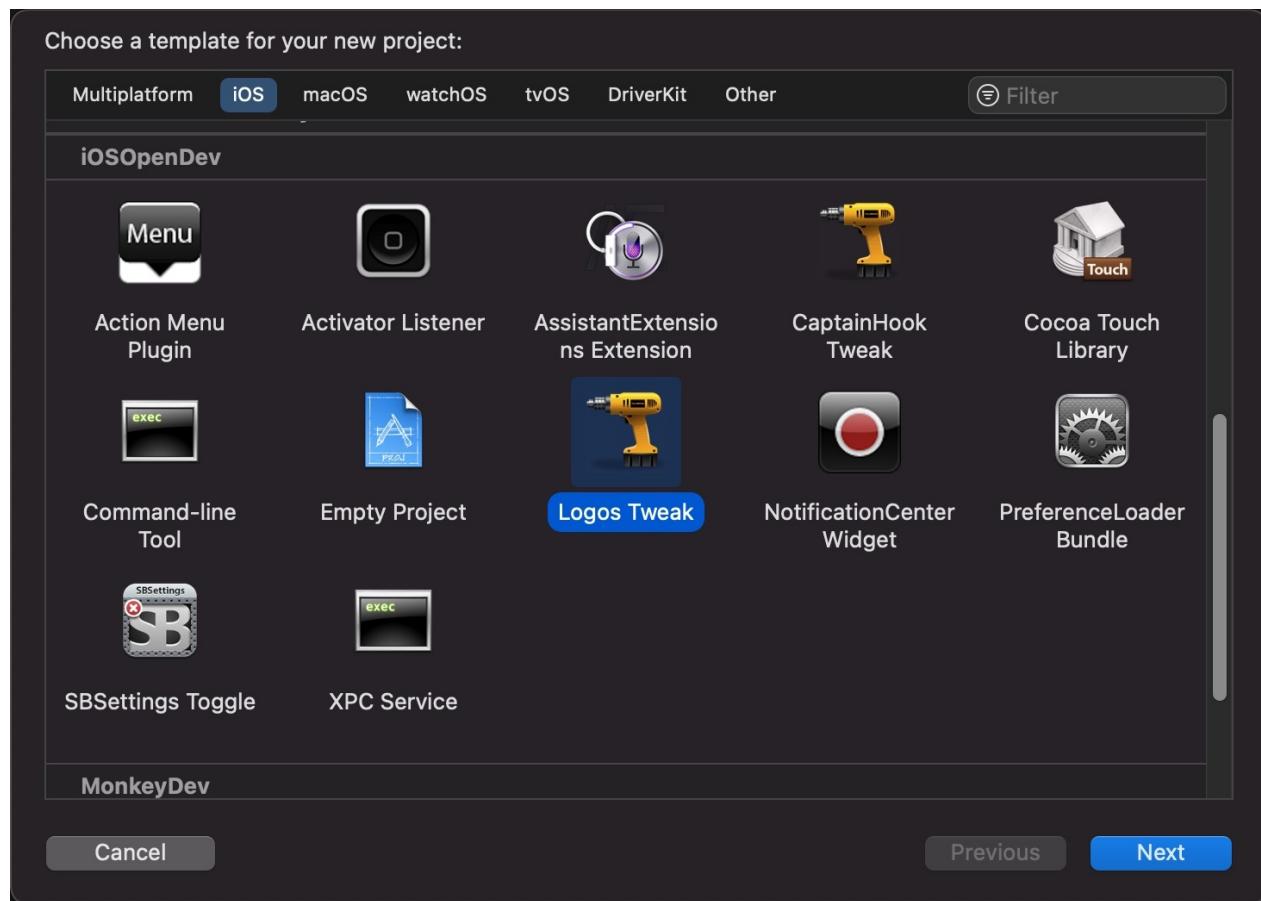
TODO:

- 【已解决】Mac中用iOSOpenDev开发iOS的theos的Logos的tweak插件
 - 【已解决】给iOSOpenDev的Logos的tweak的XCode项目去做基本配置
-

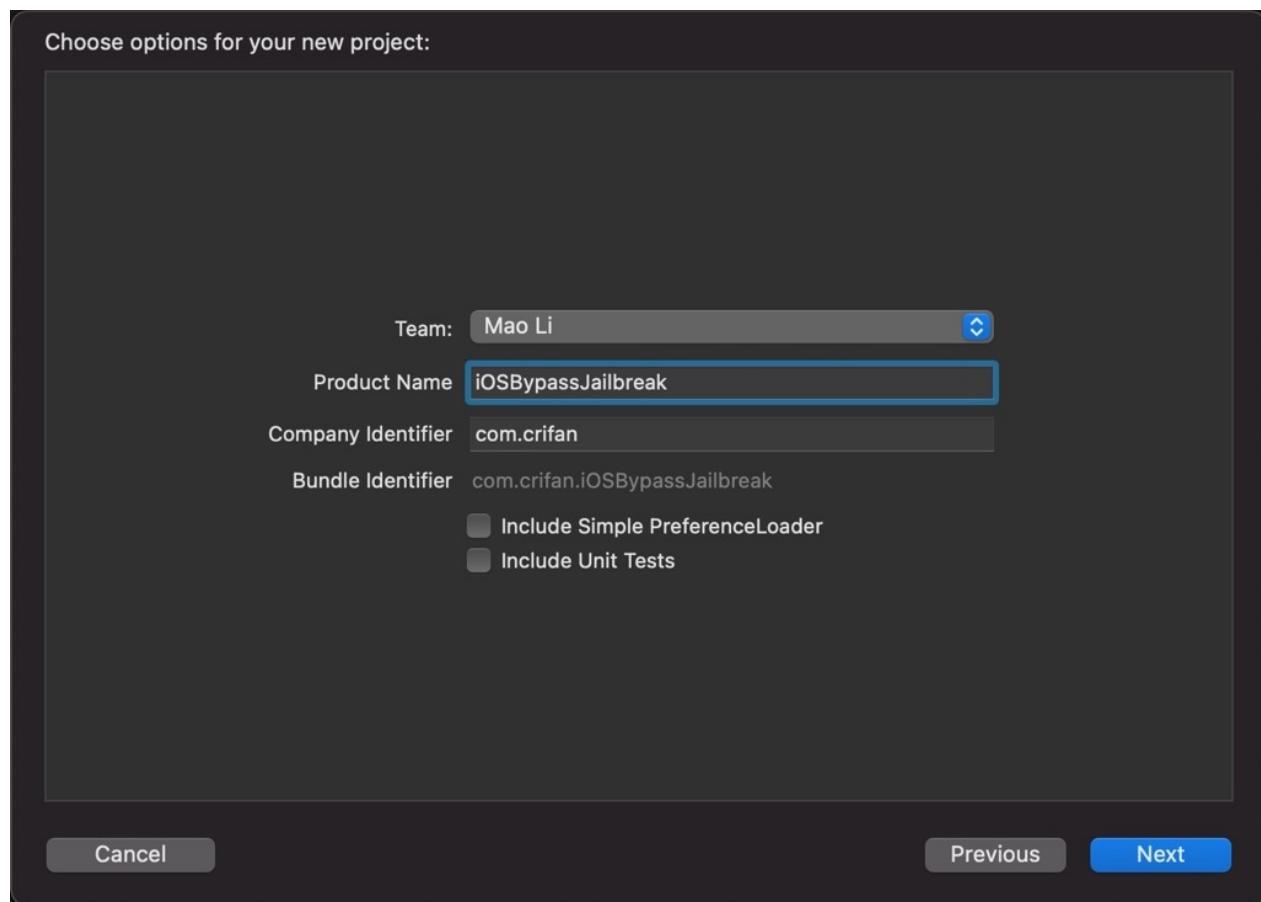
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-08 17:03:23

新建iOSOpenDev的Xcode项目

Xcode 中新建 iOS 项目，选择： iOSOpenDev -> Logos Tweak :



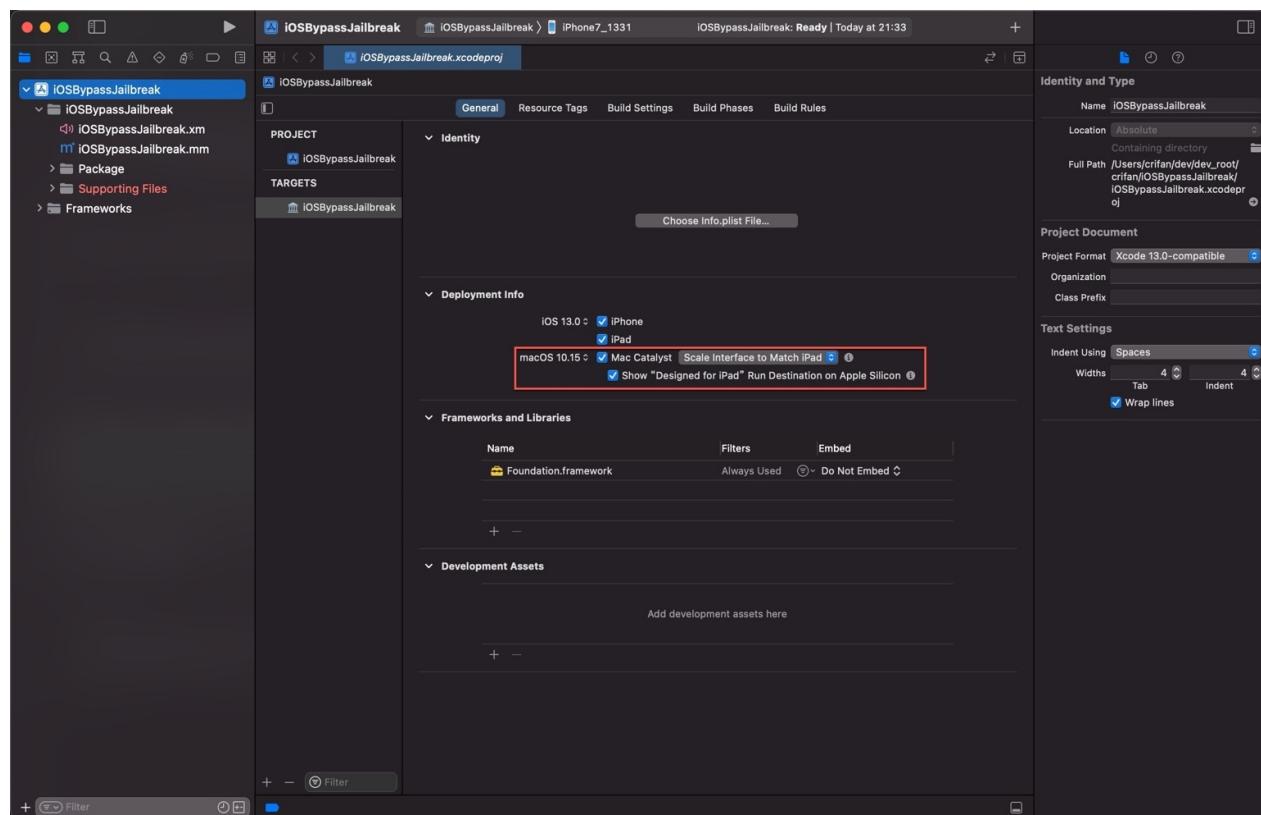
然后填写项目信息：



比如*iOSBypassJailbreak*的：

- Product Name : iOSBypassJailbreak
- Company Bundle : com.crifan
- Bundle Identifier : 自动生成出 com.crifan.iOSBypassJailbreak

点击 Next 继续，即可新建出，看起来和普通 Xcode 没多大区别的项目：

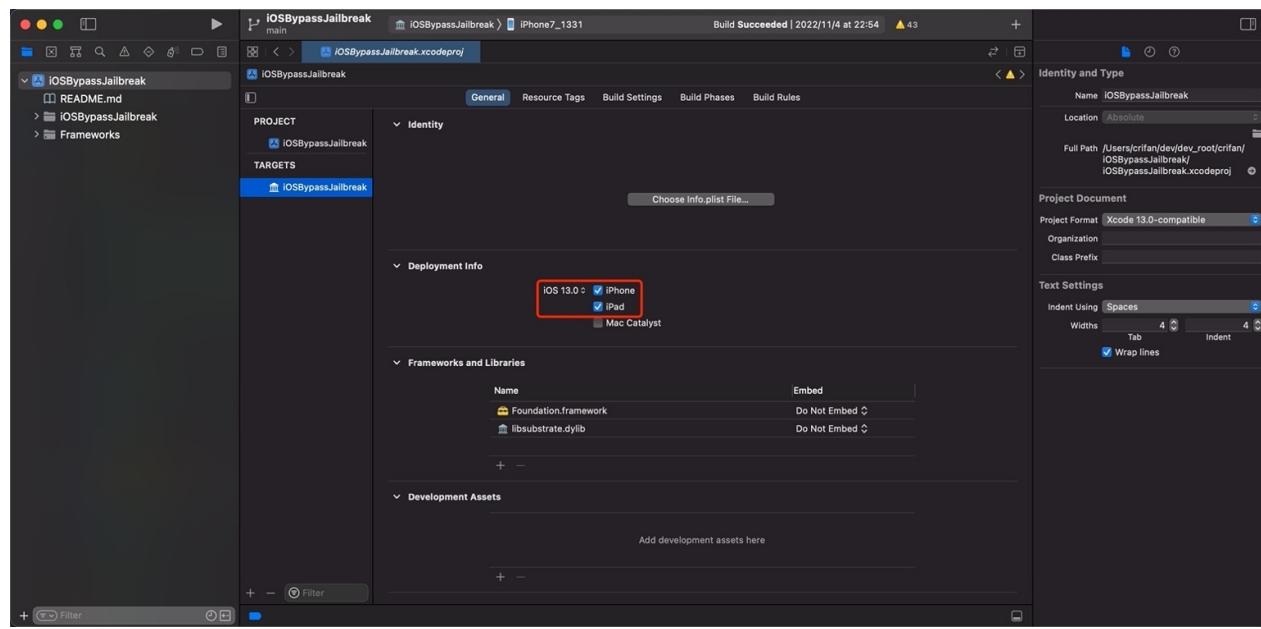


crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-11-08 16:05:21

初始化配置iOSOpenDev的Xcode项目

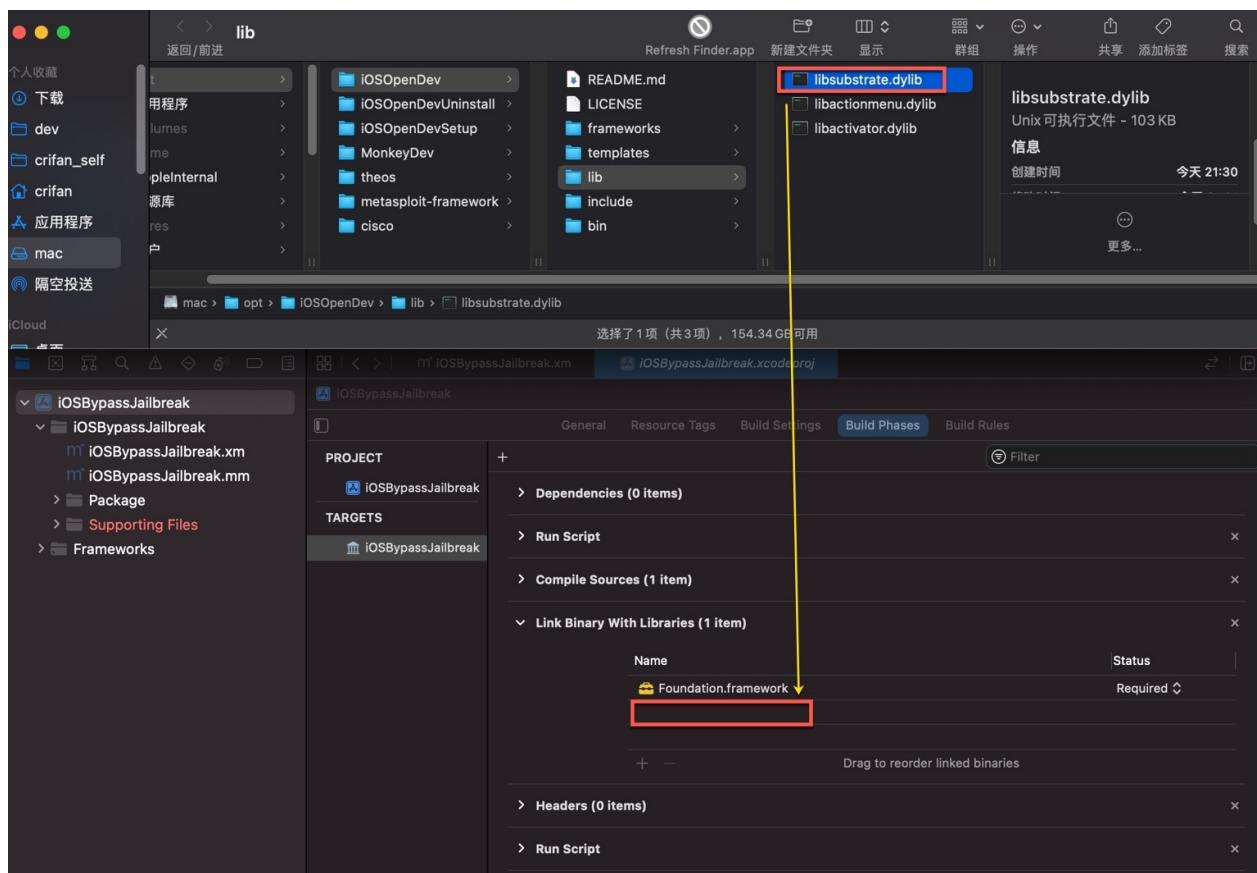
去掉 Deployment Info 中的 Mac

此处，先去做第一个配置方面的改动： Deployment Info 中去掉 Mac，因为我们开发的是 iOS 的插件，不需要发布到 Mac

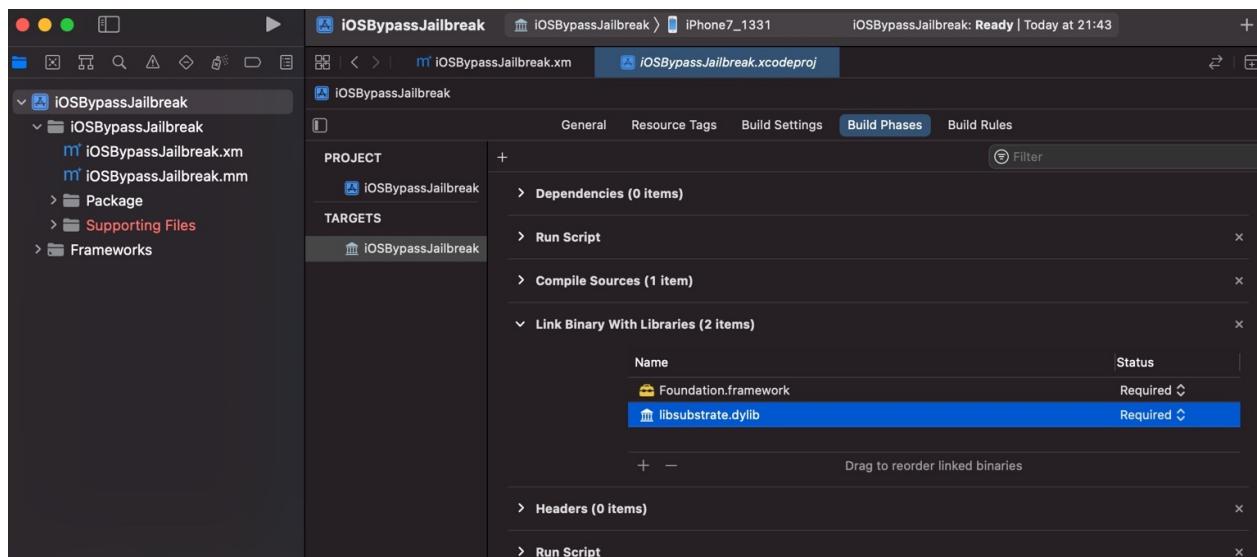


导入依赖库 libsubstrate.dylib

把 libsubstrate.dylib (一般在 /opt/iOSOpenDev/lib/libsubstrate.dylib)：



导入到项目中的： Targets -> YourProjectName -> Build Phases -> Link Binary With Libraries



设置被hook的app的包名

去把要hook的，被拦截的app的包名，加到被hook的包名的列表中：

```
YourProjectName -> YourProjectName -> Package -> Libarary -> MobileSubstrate -> DynamicLibraries -> CurrentProjectBundleIdentifier.plist
```

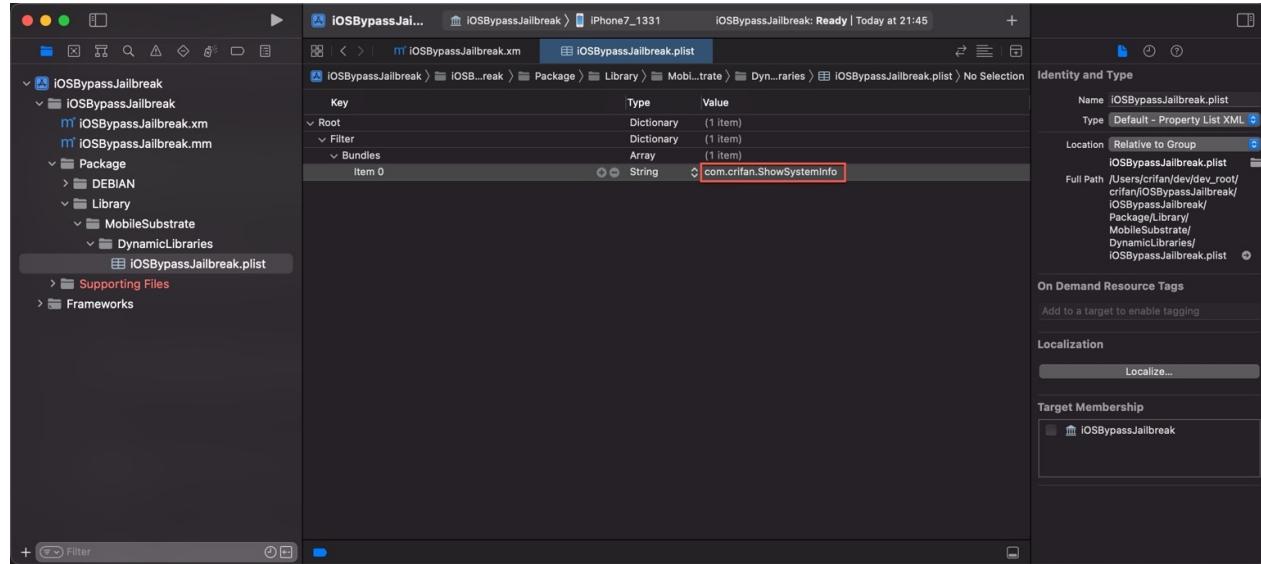
在 Root -> Filter -> Bundles，会看到 Item 0 :

- Type : String

- `Value` : 填入你要hook的app的包名

- 举例

- `com.crifan.ShowSystemInfo`



- 另外

- 如果要新增一行

- 移动到 `Item 0` 所在的行，会看到出现个 = 加号，点击 加号，会新增一行

设置iPhone的IP

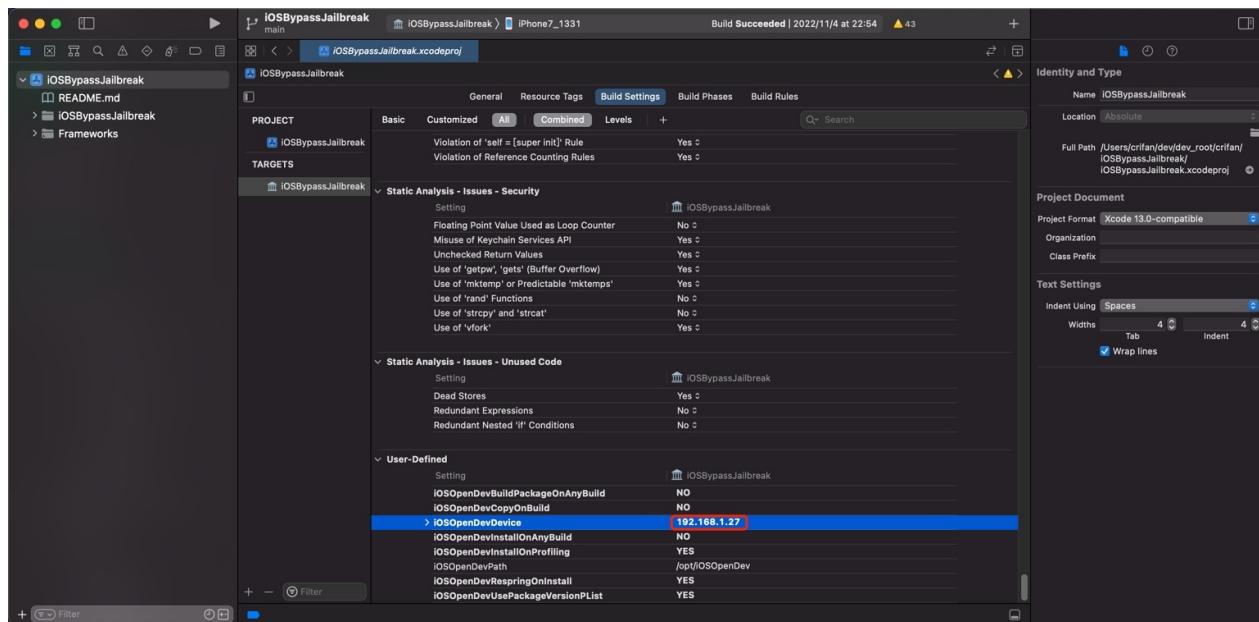
把此处要把iOS调试设备（iPhone）中的IP地址：



此处是：`192.168.1.27`

去加到配置中去：

- `iOSOpenDevDevice = 192.168.1.27`



附上原默认的更新后的配置：

```
iOSOpenDevBuildPackageOnAnyBuild = NO
iOSOpenDevCopyOnBuild = NO
iOSOpenDevDevice = 192.168.1.27
iOSOpenDevInstallOnAnyBuild = NO
iOSOpenDevInstallOnProfiling = YES
iOSOpenDevPath = /opt/iOSOpenDev
iOSOpenDevRespringOnInstall = YES
iOSOpenDevUsePackageVersionPList = YES
```

另外，理论上，去把对应变量加到环境变量：

```
→ ~ cat ~/.zshrc | grep iOSOpenDevDevice
export iOSOpenDevDevice 192.168.1.27
```

效果应该也是一样的。

确保ssh免密登录iPhone

此处iOSOpenDev内部在调试期间，会自动通过ssh访问iPhone设备，把生成的 .deb 插件的文件下载和安装到iPhone中

此时就需要先准备好环境：确保 Mac 中可以，ssh的免密登录iPhone

此处ssh免密登录的具体步骤是：

- 先用ssh登录一次iPhone
 - 命令

```
ssh root@192.168.1.27
```

- 输入密码

- OpenSSH 的默认密码是： alpine
- 即可登录到iPhone中
- 把ssh的key拷贝到iPhone中
 - 命令

```
ssh-copy-id root@192.168.1.27
```

- 输入密码： alpine

即可实现， ssh免密登录：

以后ssh直接可以访问iPhone， 而无需输入密码

常见问题

如果没有ssh免密登录，则常会看到对应的错误提示：

```
Preparing to run Xcode Build Phase...
Signing /Users/crifan/Library/Developer/Xcode/DerivedData/iOSBypassJailbreak-bfqgivvncccwmeaykh
tbvgylkkq/Build/Products/Release-iphoneos/iOSBypassJailbreak.dylib with ldid... Done.
Copying /Users/crifan/Library/Developer/Xcode/DerivedData/iOSBypassJailbreak-bfqgivvncccwmeaykh
tbvgylkkq/Build/Products/Release-iphoneos/iOSBypassJailbreak.dylib to package directory at /Us
ers/crifan/dev/dev_root/crifan/iOSBypassJailbreak/iOSBypassJailbreak/Package/Library/MobileSubs
trate/DynamicLibraries...
Preparing to build package...
Setting control file /Users/crifan/dev/dev_root/crifan/iOSBypassJailbreak/iOSBypassJailbreak/Pa
ckage/DEBIAN/control Version field to 1.0-1 using /Users/crifan/dev/dev_root/crifan/iOSBypassJa
ilbreak/iOSBypassJailbreak/PackageVersion.plist... Done.
Building package ... Done.
Creating zip /Users/crifan/dev/dev_root/crifan/iOSBypassJailbreak/Packages/com.crifan.iOSBypass
Jailbreak_1.0-1_iphoneos-arm.zip... Done.
Host key verification failed.
Failed to create directory /var/root/iOSOpenDevPackages on device 192.168.1.27
Command PhaseScriptExecution failed with a nonzero exit code
```

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 16:08:13

写hook插件代码

iOSOpenDev中的hook插件代码的逻辑是：

- `.xm` : 原始的hook插件的代码
 - 写hook插件，是改动 `.xm` 文件
 - 而不需要，也不应该改动 `.mm` 文件
- `.mm` : 从 `.xm` 自动（在 Build后）自动生成的文件
 - 后续真正编译的文件是 `.mm` 文件
 - 可以在 `Compiled Sources` 中看到 `.mm` 文件（而不是 `.xm` 文件）

新建 iOSOpenDev 的项目中的 `.xm` 文件（此处是 `iOSBypassJailbreak.xm`）生成的默认代码，来自模板，一般是：

```
// Logos by Dustin Howett
// See http://iphonedevwiki.net/index.php/Logos

#error iOSOpenDev post project creation from template requirements (remove these lines after completed) -- \
Link to libsubstrate.dylib: \
(1) go to TARGETS > Build Phases > Link Binary With Libraries and add /opt/iOSOpenDev/lib/libsubstrate.dylib \
(2) remove these lines from *.xm files (not *.mm files as they're automatically generated from *.xm files)

hook ClassName

+ (id)sharedInstance
{
    log;

    return %orig;
}

- (void)messageWithNoReturnAndOneArgument:(id)originalArgument
{
    log;

    %orig(originalArgument);

    // or, for example, you could use a custom value instead of the original argument: %orig(customValue);
}

- (id)messageWithReturnAndNoArguments
{
    log;

    id originalReturnOfMessage = %orig;

    // for example, you could modify the original return value before returning it: [SomeOtherClass doSomethingToThisObject:originalReturnOfMessage];
}
```

```
    return originalReturnOfMessage;
}

End
```

去删除掉，或注释掉，改自己的hook的代码。

比如此处仅用于演示的代码：

The screenshot shows the Xcode interface with the project 'iOSBypassJailbreak' open. The file 'iOSBypassJailbreak.m' is selected in the center editor pane. The code in the file is as follows:

```
// return originalReturnOfMessage;
// */

%hook UIDevice
- (NSString *)name
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice name");
    return @"Tweaked_name";
}

- (NSString *)systemName
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice systemName");
    return @"Tweaked_systemName";
}

- (NSString *)systemVersion
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice systemVersion");
    return @"Tweaked_systemVersion";
}

- (NSString *)model
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice model");
    return @"Tweaked_model";
}
%end
```

```
#import <UIKit/UIKit.h>
//#import <SpringBoard/SpringBoard.h>
// #import <Preferences/Preferences.h>
#import <os/log.h>

%hook UIDevice

- (NSString *)name
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice name");
    return @"Tweaked_name";
}

- (NSString *)systemName
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice systemName");
    return @"Tweaked_systemName";
}

- (NSString *)systemVersion
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice systemVersion");
    return @"Tweaked_systemVersion";
}
```

```
- (NSString *)model
{
    os_log(OS_LOG_DEFAULT, "test hook UIDevice model");
    return @"Tweaked_model";
}

end
```

如何新增(.xm 和 .mm)文件

有时候，需要去新增文件： .xm 和 .mm

具体步骤是：

- 新建 .xm 文件
 - 选中要新增文件所属的位置 -> 右键 -> Add File -> iOS -> Other -> Empty -> 输入文件名： yourFilename.xm -> Create
- 编译 -> 会生成对应 .mm 文件
 - Product -> Build
 - 会从 yourFilename.xm 生成 yourFilename.mm
- 把 .mm 文件加到 Compile Sources 中
 - 右键-> Add Files to {yourProjectName} -> 选择（刚新生成的）yourFilename.mm
 - 项目文件列表中，即可新增对应文件 yourFilename.mm
 - 项目的待编译的文件中，也包含了对应的 .mm 文件
 - Targets -> Build Phase -> Compile Sources 中有了刚加入的 .mm 文件
 - 如果文件，点击 加号 = ，去新增导入进来
 - 这样后续编译代码时，才能真正编译到对应hook代码

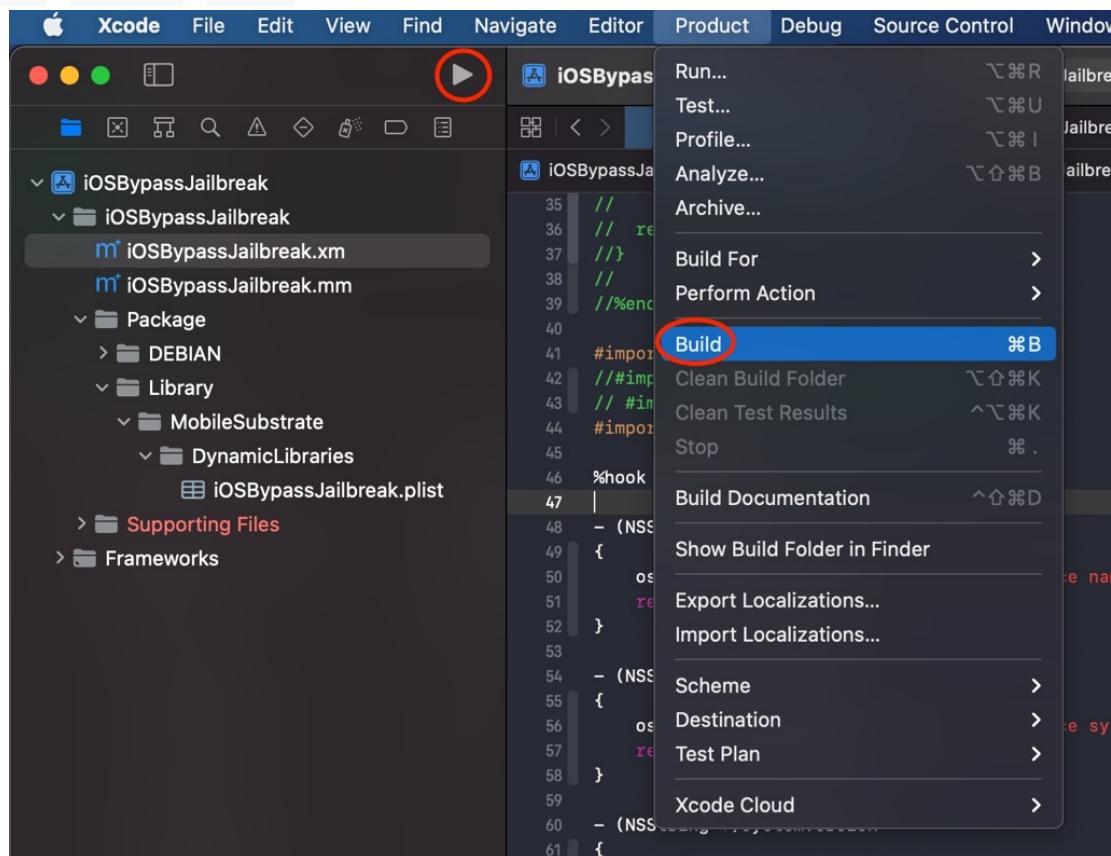
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：

2022-11-08 16:43:25

调试插件代码

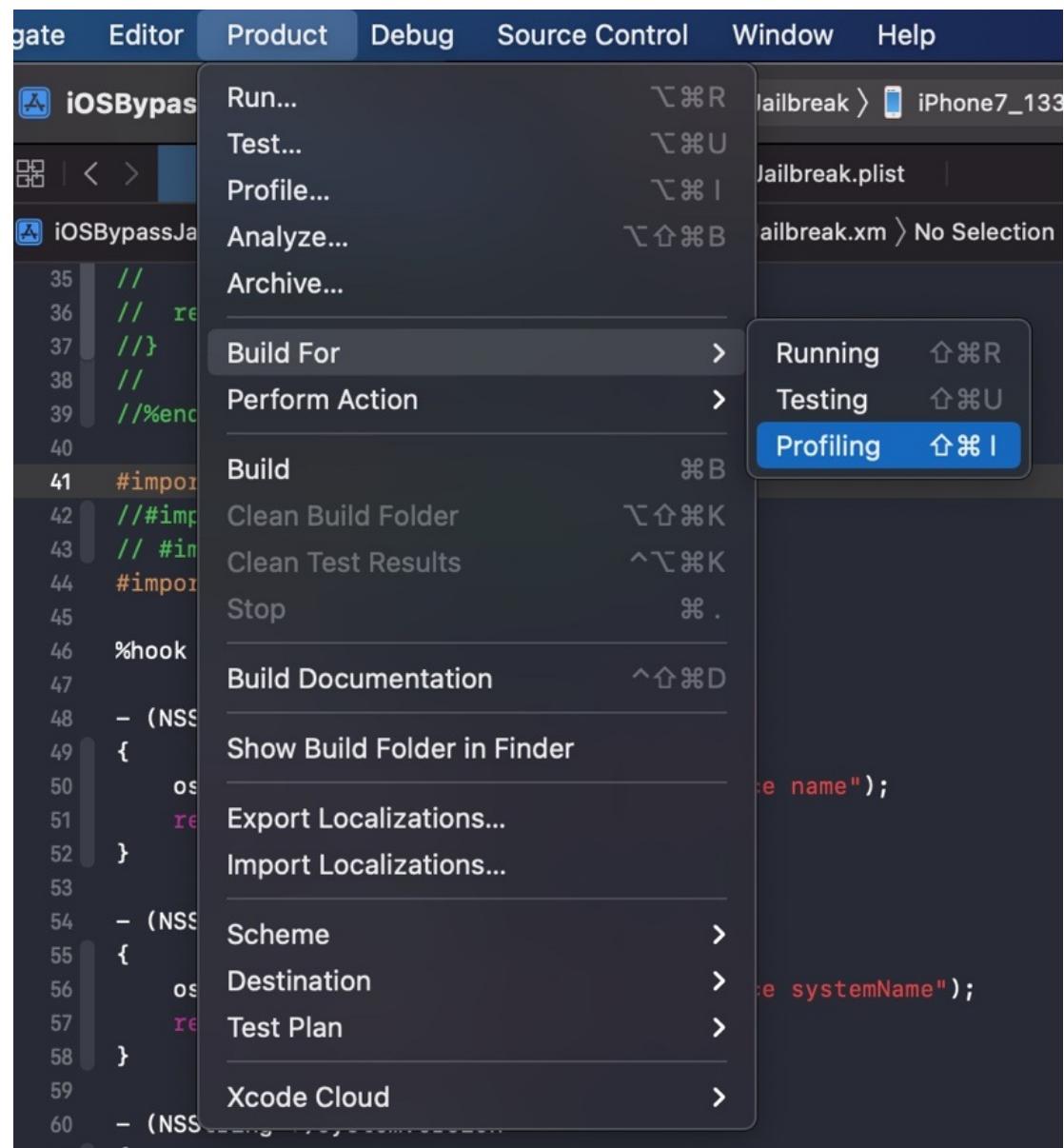
接下来就是典型的插件开发流程了：

- 写代码 = 写hook代码= 写tweak插件代码 = 改动 .xm 文件
- 编译代码 -> 确保语法没错，可以正常编译
 - Xcode -> Product -> Build



- 调试和运行 -> 把hook插件代码编译所生成的插件(.deb 文件)安装到iOS设备(iPhone)中，测试插件效果

- Product -> Build For -> Profiling



确认插件安装成功

- iPhone中看到自己的插件
 - Cydia -> 已安装 -> 最近 能看到自己的插件:



- 点击插件，可以看到插件基本信息



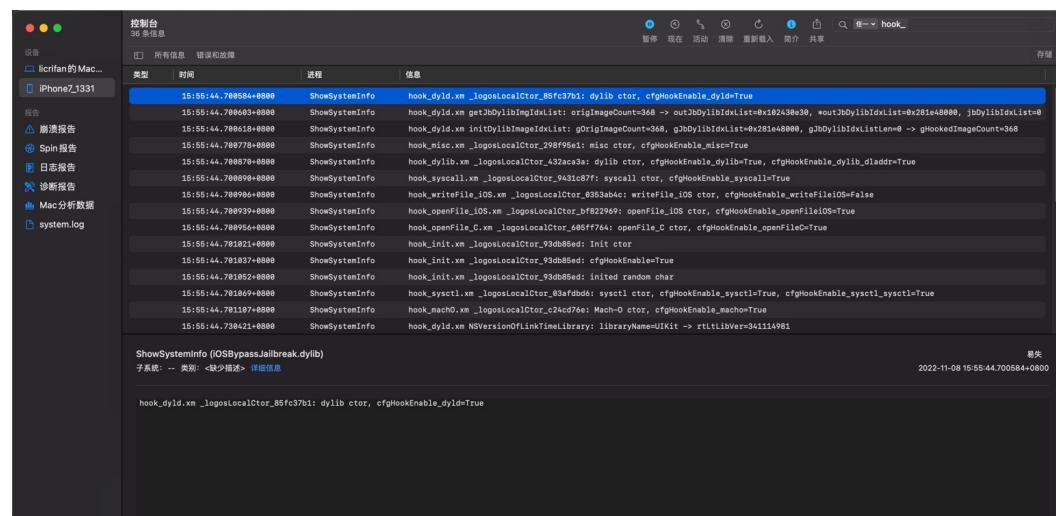
- 点击插件的文件，可以看到文件列表



确认插件的确正常工作

- 打开被测试的=被hook的app，看到此处测试代码：更改信息信息，显示是我们hook代码中的值，表示hook成功

-
- 查看对应log日志
 - Xcode -> Window -> Devices and Simulators -> Devices ->选中 Connected 中自己的iPhone设备-> Open Console ->打开 Console = 控制台，显示出对应iPhone的log日志
 - 其中就有你的插件的log日志
 - 如果没有，则自己去右上角，搜索对应关键字，即可搜到
 - 此处贴出，后续更新了代码后的相关log



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 16:44:50

带界面的插件开发流程

TODO:

- 【未解决】用XCode开发iOS的app用于配置改机软件参数
 - 【已解决】用iOSOpenDev去开发带GUI图形界面的iOS的app和tweak插件集成在一起的插件deb包
 - 【已解决】如何把普通iOS的app的XCode项目和iOSOpenDev的Logos插件tweak集成到一起
 - 【已解决】越狱iPhone中安装deb包后iOS的反越狱插件没生效
 - 【已解决】把iOS的app和iOS的tweak插件打包成独立的deb安装包
 - 【记录】越狱iPhone中安装iOS的app和tweak合并出的deb安装包
 - 【已解决】把iOSOpenDev的tweak插件和app合并打包成deb文件
 - 【已解决】越狱iOS如何用Theos开发带GUI图形界面的插件
 - 【记录】重新给iOSOpenDev的tweak加app打包deb看看app是否有权限写入Preferences目录
 - 【已解决】iOS的writeToURL报错：NSCocoaErrorDomain Code 513 You don't have permission to save the file in the folder Preferences
 - 【已解决】给iOSOpenDev的app和tweak用配置文件互相通信
 - 【已解决】iOSOpenDev的tweak中读取app保存出的配置文件参数
 - 【记录】重新给iOSOpenDev的tweak加app打包deb看看app是否有权限写入Preferences目录
 - 【已解决】iOSOpenDev的XCode编译报错：An empty identity is not valid when signing a binary for the product type Application
 - 【记录】确认iPhone中安装后的tweak加app是否正常使用
 - 【已解决】iOSOpenDev的XCode去Build For Profiling安装后iPhone桌面上找不到iOS的app的图标
 - 【已解决】把iOSOpenDev的tweak加app的deb文件安装到已越狱的iPhone中
 - 【已解决】用Filza安装tweak加app的deb后iPhone桌面中仍没出现iOS的app的logo图标
-

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 10:57:57

心得

TODO:

- 【未解决】 XCode调试警告: was compiled with optimization stepping may behave oddly variables may not be available
 - 【未解决】 iOSOpenDev的XCode的tweak插件编译尝试去掉优化加上调试信息
 -
 - 【已解决】 iOSOpenDev的XCode调试iPhone6报错: Unable to install The application could not be verified
 - 【已解决】 XCode中删除掉User-Defined的自定义参数
 - 【已解决】 XCode中删除用户自定义配置User-Defined中的 CODE_SIGNING_ALLOWED=NO
 -
 - 【已解决】 iOSOpenDev的XCode的iOS的tweak插件中实现ObjC的通用全局函数
 - 【已解决】 iOSOpenDev的XCode的iOS插件运行报错: ImageLoaderMachO doModInitFunctions和 _logosLocallInit
 - 【已解决】 iOS代码报错: objc Class is implemented in both app and dylib One of the two will be used Which one is undefined
 -
 - 【已解决】 iOSOpenDev的XCode项目编译报错: iPhone Developer no identity found
 - 【已解决】 调试iOSOpenDev的XCode的iOS的app
 - 【已解决】 研究iOSOpenDev的XCode项目编译过程以确保如何链接自定义.c文件的.o文件
 - 【已解决】 iOSOpenDev的XCode项目偶尔编译非常慢卡死
 - 【已解决】 iOSOpenDev的XCode中新增.c和.h文件并正常编译
 - 【已解决】 如何把XCode的iOS的app项目转换成iOSOpenDev的项目
 - 【已解决】 对比研究FakeWeChatLoc和自己的XCode项目的目录结构区别
 - 【已解决】 iOSOpenDev的XCode调试iPhone7报错: Unable to install A system application with the given bundle identifier is already installed on the device and cannot be replaced
 - 【记录】 更新iOSOpenDev的Logos插件的code signing签名配置
 - 【已解决】 XCode中iOSOpenDev开发插件代码报错: No matching function for call to strcpy
 - 【已解决】 XCode中iOSOpenDev的Tweak项目中Build Settings中User-Defined中添加和引用变量 THEOS
 - 【记录】 研究XCode中clang编译mm文件的过程和编译参数
 - 【记录】 深究为何此处XCode编译strcpy会报错No matching function for call to
 - 【未解决】 把之前theos的tweak改机剩余功能移植到iOSOpenDev的XCode中
 - 【已解决】 XCode中iOSOpenDev中修改control的Version版本号无效会被重置
 -
-

.xm 文件和 .mm 文件

TODO:

- 【已解决】 Xcode中xm源码中无法看到和添加断点
- 【已解决】 iOSOpenDev的XCode中.xm文件包含.c中函数找不到报错: Undefined symbols for

architecture arm64 referenced from

- 【已解决】XCode的iOSOpenDev项目报错：Failed Logos Processor Could not open xm
- 【已解决】iOSOpenDev的XCode中xm代码%hookf编译报错：Expected unqualified-id
- 【已解决】iOSOpenDev的XCode中如何把Tweak的xm代码拆分成多个文件模块
- 【未解决】iOSOpenDev的iosod的bug修复：Logos的预处理不支持group子目录中的xm文件

.xm 文件

- .xm 文件， 默认会被 Xcode 识别为 音频文件
 - 需要去改变文件类型为 Objective c (或 objective c++) 文件，再去显示为源代码

代码高亮

- 【已解决】iOSOpenDev的XCode中新增xm文件设置为Logos语法高亮但无效
- 【已解决】让XCode的iOSOpenDev中Logos的xm文件支持语法高亮

iOSOpenDev内部逻辑和过程

TODO:

- 【未解决】研究iOSOpenDev的XCode项目编译过程以确保如何链接自定义.c文件的.o文件
- 【已解决】XCode编译iOSOpenDev的Logo Tweak项目报错： Command PhaseScriptExecution failed with a nonzero exit code Failed to locate Logos Processor
- 【未解决】XCode中编译iOSOpenDev的Logos的Tweak时shell从sh换为zsh
- 【已解决】给iOS的XCode项目中新增iOSOpenDev的Project Navigator的目录和文件
- 【已解决】XCode项目中新增iOSOpenDev的Package目录到Target目录中
- 【已解决】XCode中如何把libsubstrate.dylib动态库导入到Link Binary With Libraries

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 17:05:53

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：
2022-11-08 10:39:15

参考资料

- 【已解决】ssh登录iPhone失败：Host key verification failed
- 【已解决】Mac中安装iOSOpenDev
- 【已解决】Mac中安装iOSOpenDev报错：安装器遇到了一个错误，导致安装失败
- 【已解决】Mac中初始化iOSOpenDev环境并新建插件项目
- 【已解决】iOSOpenDev设置SDK报错：File not found XCode Specifications iPhoneOSPackageTypes.xcspec
- 【已解决】iOSOpenDev设置SDK报错：PrivateFramework directory not found XCode iPhoneOS15.0.sdk
-
- iosOpenDev-install 失败官方wiki无法解决看这里（尝试有效） - PoloKey - 博客园 (cnblogs.com)
- zhangkn/knPrivateFrameworks:
[/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS9.2.sdk/System/Library/PrivateFrameworks \(github.com\)](https://github.com/zhangkn/knPrivateFrameworks)
- iosopendev专用Specifications.zip
- 越狱开发:用iosOpenDev配置越狱开发环境 编写第一个hello world_我的杯洗具的博客-CSDN博客
-

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 17:00:29