

之前从未接触过iOS越狱和逆向方面的相关知识, 前段时间刘培庆的<iOS应用逆向与安全>一书首发, 我买了一本. 在学习过程中的一些小Tips, 分享给大家, 希望能一起学习进步. 文章将不断更新完善.



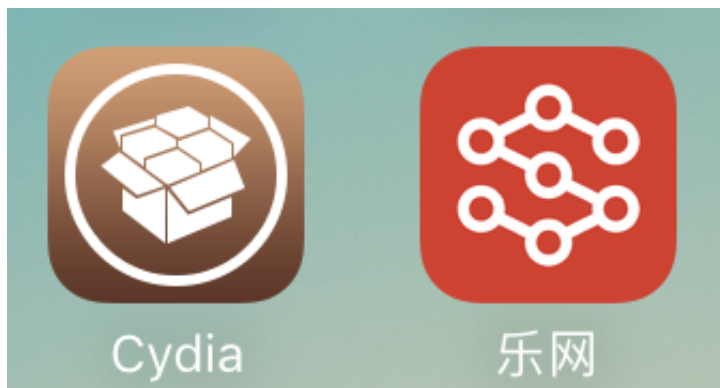
配置：

- iPhone6 乞丐版 , iOS10.3.2
- iMac macOS High Sierra 10.13.5
- Xcode 9.4.1

第一步: 越狱

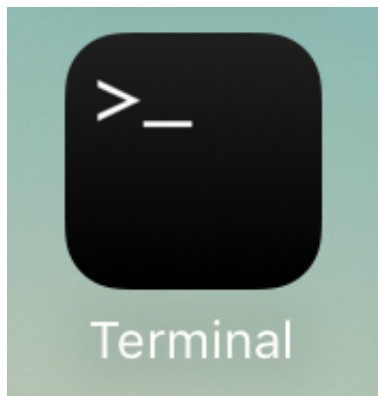
越狱教程, 网上有很多. 个人推荐最新版的爱思助手, 一键越狱, 傻瓜式操作, 简单粗暴.

出现的问题: 越狱后,Cydia无法联网. 解决方案: 爱思助手下载安装乐网, 搞定.. 爱思很强大, 依旧简单粗暴.



第二步: 按书上写的, 安装相关的插件或应用.

- Cydia中搜索并安装MTerminal 和 adv-cmds.



打开 Terminal , 运行 `ps aux | grep dropbear` , 出现下图的内容, 表示默认支持USB

```
last login: Fri Jun 29 13:42:14 on ttys000
just-bibodeiPhone:~ mobile$ su
Password:
just-bibodeiPhone:/var/mobile root# cd /
just-bibodeiPhone:/ root# ps aux | grep dropbear
root   14583  0.0  0.0  1069488   8  ??  Ss   Tue02PM   0:00.08 /usr/local/bin/dropbear -F -R -p 22
root   17948  0.0  0.1   633680   8  s000  R*   1:44PM   0:00.01 grep dropbear
just-bibodeiPhone:/ root#
```

连接.

- 在Mac终端运行命令:

```
iproxy 2222 22
```

```
ssh -p 2222 root@localhost
```

第一次操作, 要验证密码, 相关内容书中都有, 还有修改密码等等操作 不再这里赘述.

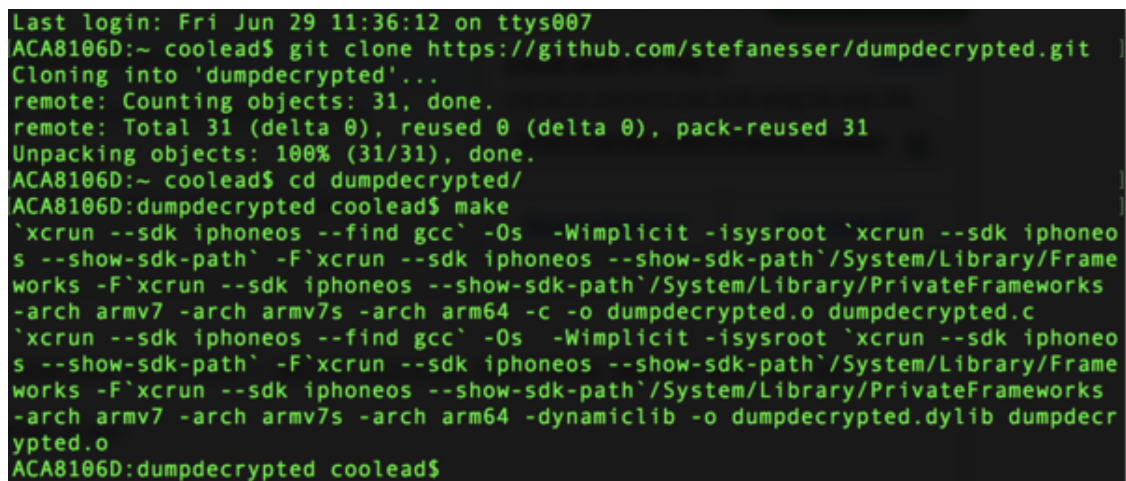
书中第10页写成了 iproxy 22 2222, 已有很多人在 [iOSREBook-issues](#) 反馈了, 大家有什么问题, 可以向作者反馈

- 安装浏览文件系统工具, Mac上安装某某助手, 或者 fFunBox. iPhone 安装iFile. 如果iFile安装失败, 可以百度搜索其它源, 注意兼容版本.
- Cydia 安装 Substrate, appsync(如果安装失败, 爱思助手可以帮你 -); scp(或者 rsync).

第三步骤: 开始逆向.

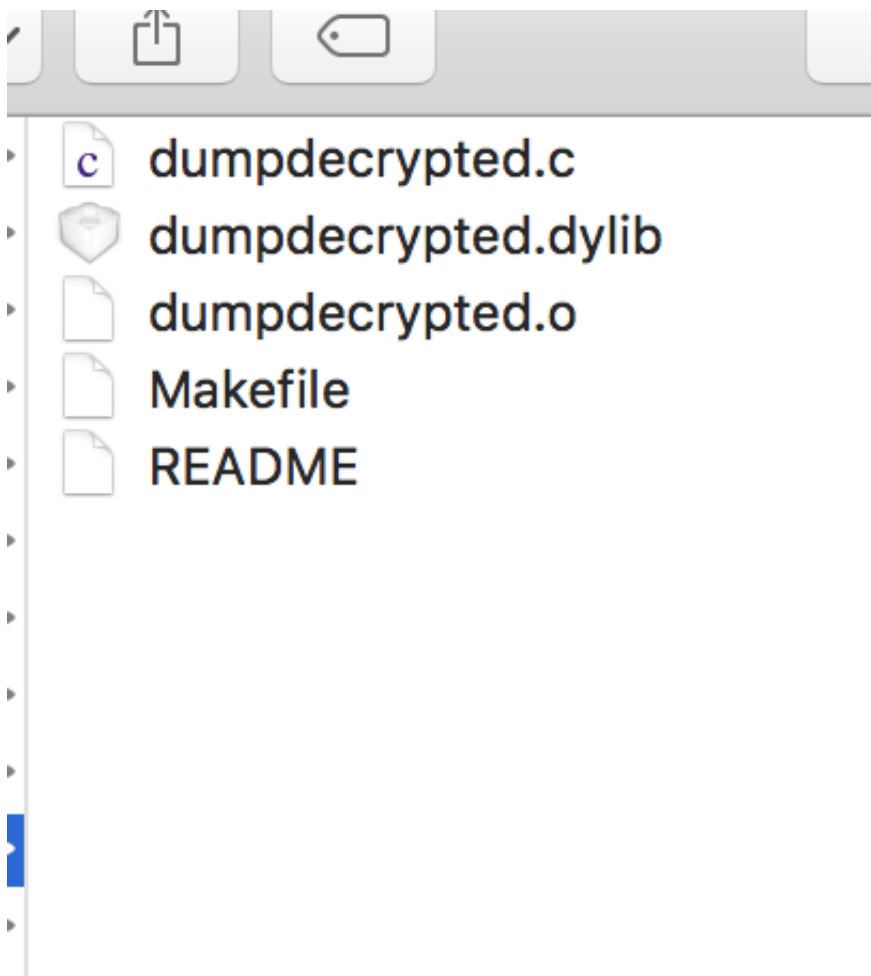
- 应用解密 1. dumpdecrypted. 从GitHub下载并编译.

```
git clone https://github.com/stefanesser/dumpdecrypted.git
cd dumpdecrypted
make
```



```
Last login: Fri Jun 29 11:36:12 on ttys007
ACA8106D:~ coolead$ git clone https://github.com/stefanesser/dumpdecrypted.git
Cloning into 'dumpdecrypted'...
remote: Counting objects: 31, done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 31
Unpacking objects: 100% (31/31), done.
ACA8106D:~ coolead$ cd dumpdecrypted/
ACA8106D:dumpdecrypted coolead$ make
`xcrun --sdk iphoneos --find gcc` -Os -Wimplicit -isysroot `xcrun --sdk iphoneos --show-sdk-path` -F`xcrun --sdk iphoneos --show-sdk-path`/System/Library/Frameworks -F`xcrun --sdk iphoneos --show-sdk-path`/System/Library/PrivateFrameworks -arch armv7 -arch armv7s -arch arm64 -c -o dumpdecrypted.o dumpdecrypted.c
`xcrun --sdk iphoneos --find gcc` -Os -Wimplicit -isysroot `xcrun --sdk iphoneos --show-sdk-path` -F`xcrun --sdk iphoneos --show-sdk-path`/System/Library/Frameworks -F`xcrun --sdk iphoneos --show-sdk-path`/System/Library/PrivateFrameworks -arch armv7 -arch armv7s -arch arm64 -dynamiclib -o dumpdecrypted.dylib dumpdecrypted.o
ACA8106D:dumpdecrypted coolead$
```

执行完后, 文件夹内生成dumpdecrypted.dylib 和 dumpdecrypted.o 两个文件



书中说使用 dumpdecrypted.dylib 进行注入解密. 实际操作有问题, 这个后面再说.

2. 手机运行需要解密的TargetApp (以某信为例), 然后在终端执行命令 ps -e

```
18047 ??      0:00.09 /usr/libexec/ptpd -t usb
18045 ??      0:00.70 /System/Library/PrivateFrameworks/SyncedDefaults.Framework/Support/syncdefaultsd
18047 ??      0:00.13 /usr/local/bin/dropbear -F -R -p 22
18051 ??      0:00.21 /System/Library/PrivateFrameworks/SafariSafeBrowsing.framework/com.apple.Safari.SafeBrowsing.Service
18054 ??      0:07.70 /var/containers/Bundle/Application/F68AA473-CCB2-4872-85CB-7DE710FD214D/TargetApp.app/TargetApp
18048 ttys000    0:00.11 -sh
18055 ttys000    0:00.01 ps -e
just-bibodeiPhone:/ root#
```

然后获取TargetApp 的 Bundle id, 在终端执行以下命令:

```
cat /var/containers/Bundle/Application/F68AA473-CCB2-4872-85CB-7DE710FD214D/TargetApp.app/Info.plist | grep CFBundleIdentifier -A 1
```

```
just-bibodeiPhone:/ root# cat /var/containers/Bundle/Application/F68AA473-CCB2-4872-85CB-7DE710FD214D/TargetApp.app/Info.plist | grep CFBundleIdentifier -A 1
<key>CFBundleIdentifier</key>
<string>com. TargetApp</string>
just-bibodeiPhone:/ root#
```

3. Xcode新建一个iOS App (我命名为MonkeyApp), 在 `- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions` 中执行以下代码:

```
NSString *bundleID = TargetApp 的 Bundle id. ;
NSURL *dataURL = [[NSClassFromString(@"LSApplicationProxy")
performSelector:@selector(applicationProxyForIdentifier:)
withObject:bundleID] performSelector:@selector(dataContainerURL)];
NSLog(@"%@ ", [dataURL.absoluteString
stringByAppendingString:@"/Documents"]);
```

终端打印结果为:

file:///private/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents,

cd 到 MonkeyApp 文件夹下.

```
root@bibi001:~# cd /var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents
root@bibi001:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents#
```

4. Mac终端执行命令 将 dumpdecrypted.dylib 拷贝到
/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-
DABE22887F1B/Documents 文件夹下

```
ACA8106D:dumpdecrypted coollead$ scp -P 2222 ./dumpdecrypted.dylib root@localhost:
/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Do
cuments
root@localhost's password:
dumpdecrypted.dylib                                100% 193KB  2.2MB/s  00:00
ACA8106D:dumpdecrypted coollead$
```

- 5.解密 终端在MonkeyApp 的Documents文件夹下, 执行以下命令:

```
DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib  
/var/containers/Bundle/Application/F68AA473-CCB2-4872-85CB-  
7DE710FD214D/TargetApp.app/arGetApp
```

```

just-b1bodeiPhone: / root# cd /var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents
just-b1bodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root# DYLD_INSERT_LIBRARIES=dumpcrypt
ted.dylib /var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root# DYLD_INSERT_LIBRARIES=dumpcrypt
dyld: could not load inserted library 'dumpcrypt.ted.dylib' because no suitable image found. Did find:
    dumpdecrypted.dylib: required code signature missing for 'dumpdecrypted.dylib'

/private/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents/dumpdecrypted.dylib: required code signa
ture missing for '/private/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents/dumpdecrypted.dylib'

Abort trap: 6
just-b1bodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root#

```

执行完后,失败,并没有出现书中描述的场景,观察终端输出结果,关键在图中划红线

的这句话: `dumpdecrypted.dylib: required code signature missing for 'dumpdecrypted.dylib'`

这句话的意思是 `dumpdecrypted.dylib` 需要 code signature , 那我们就给他 code signature .. Mac终端执行命令:

```
## 列出可签名证书
security find-identity -v -p codesigning
## 为dumpdecrypted.dylib签名
codesign --force --verify --verbose --sign "iPhone Developer: xxx
xxxx (xxxxxxxxxx)" dumpdecrypted.dylib
```

```
ACA8106D:dumpdecrypted coollead$ codesign --force --verify --verbose --sign "iPhone Developer: [redacted]
[redacted] ./dumpdecrypted.dylib
./dumpdecrypted.dylib: signed Mach-O universal (armv7 armv7s arm64) [dumpdecrypted]
ACA8106D:dumpdecrypted coollead$
```

注意, 你的 MonkeyApp 必须是同一个 iPhone Developer 签名, Xcode 运行 MonkeyApp 必须是 release 模式

然后重复上面的 第 4 步 和 第 5 步, 出现下图中的内容则解密成功

```
just-bibodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root# DYLD_INSERT_LIBRARIES=dumpdecryp
ted.dylib /var/mobile/Containers/Data/Application/F68AA473-CCB2-4872-85CB-7DE710FD214D/[redacted].app/[redacted]
mach-o decryption dumper

DISCLAIMER: This tool is only meant for security research purposes, not for application crackers.

[+] detected 64bit ARM binary in memory.
[+] offset to cryptid found: 0x1000bcca8(from 0x1000bc000) = ca8
[+] Found encrypted data at address 00004000 of length 58949632 bytes - type 1.
[+] Opening /private/var/mobile/Containers/Data/Application/F68AA473-CCB2-4872-85CB-7DE710FD214D/[redacted].app/[redacted] for reading.
[+] Reading header
[+] Detecting header type
[+] Executable is a plain MACH-O image
[+] Opening WeChat.decrypted for writing.
[+] Copying the not encrypted start of the file
[+] Dumping the decrypted data into the file
[+] Copying the not encrypted remainder of the file
[+] Setting the LC_ENCRYPTION_INFO->cryptid to 0 at offset ca8
[+] Closing original file
[+] Closing dump file
just-bibodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root#
```

这时候 在MonkeyApp的Documents下 会生成一个 TargetApp.decrypted 文件

```
just-bibodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root# ls -l
total 73212
drwxr-xr-x  6 mobile mobile   192 Jun 15 13:30 00000000000000000000000000000000
drwxr-xr-x 36 mobile mobile  1152 Jun 15 12:22 0acc6dc19533a96a9014435a401161d4
drwxr-xr-x  3 mobile mobile   96 Jun 29 08:12 CrashReport
-rw-r--r--  1 mobile mobile   310 Apr 25 14:28 Ksid
-rw-r--r--  1 mobile mobile  1313 Jun 29 14:15 LocalInfo.lst
-rw-r--r--  1 mobile mobile   237 Jun 15 13:30 LoginInfo2.dat
drwxr-xr-x 16 mobile mobile   512 Jun 20 08:53 MMResourceMgr
drwxr-xr-x 68 mobile mobile  2176 Jun 15 12:14 MappedKV
drwxr-xr-x  5 mobile mobile   160 Jun 29 14:15 MemoryStat
drwxr-xr-x  2 mobile mobile    64 Apr 20 12:49 OpenImResource
-rw-r--r--  1 root mobile   64320 Jun 29 12:05 [redacted]
-rw-r--r--  1 mobile mobile    15 Jun 29 14:28 SafeNode.dat
-rw-r--r--  1 root mobile 74664880 Jun 29 14:47 [redacted].decrypted
drwxr-xr-x  2 mobile mobile    64 Apr 20 16:44 app_tutt
-rw-r--r--  1 mobile mobile    8 Jun 29 14:15 db.globalconfig
-rw-r--r--  1 root mobile 207776 Jun 29 14:47 dumpdecrypted.dylib
drwxr-xr-x 49 mobile mobile  1568 Jun 29 14:28 fd4bf8f4d297500cballicad16ed6bba
-rw-r--r--  1 mobile mobile   592 Jun 7 13:43 heavy_user_id_mapping.dat
-rw-r--r--  1 mobile mobile   448 Apr 20 14:26 mmupdateinfo.archive
just-bibodeiPhone:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents root#
```

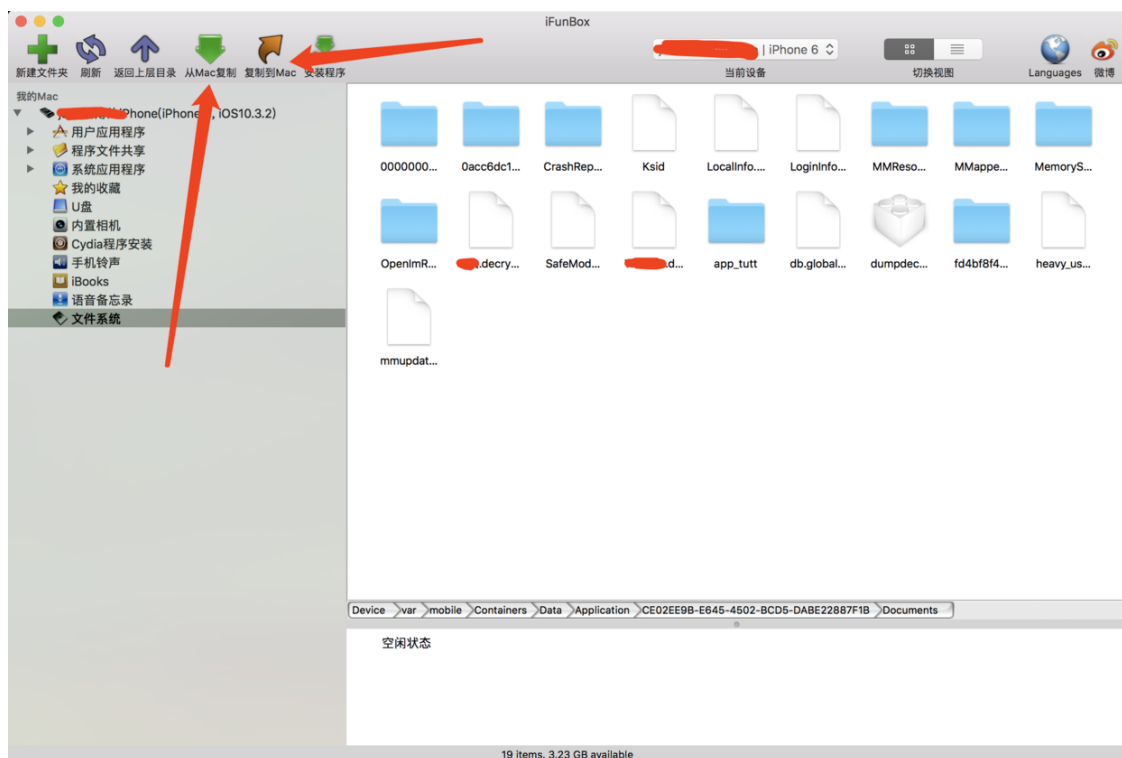
解密成功的文件

把这个文件拷贝到Mac中: 终端执行命令:

```
scp -P 2222
root@localhost:/var/mobile/Containers/Data/Application/CE02EE9B-
E645-4502-BCD5-DABE22887F1B/Documents/TargetApp.decrypted
```

```
ACA8106D:dumpdecrypted coollead$ scp -P 2222 root@localhost:/var/mobile/Containers/Data/Application/CE02EE9B-E645-4502-BCD5-DABE22887F1B/Documents/TargetApp.decrypted root@localhost:
root@localhost's password:
TargetApp.decrypted 100% 71MB 11.1MB/s 00:06
ACA8106D:dumpdecrypted coollead$
```

拷贝文件 其实不一定非要 scp 命令, 记得之前我们装过iFunBox么? 直接用 iFunBox操作就好了.... 但是 用终端命令逼格高一些, 不是吗? 😄😄😄😄😄

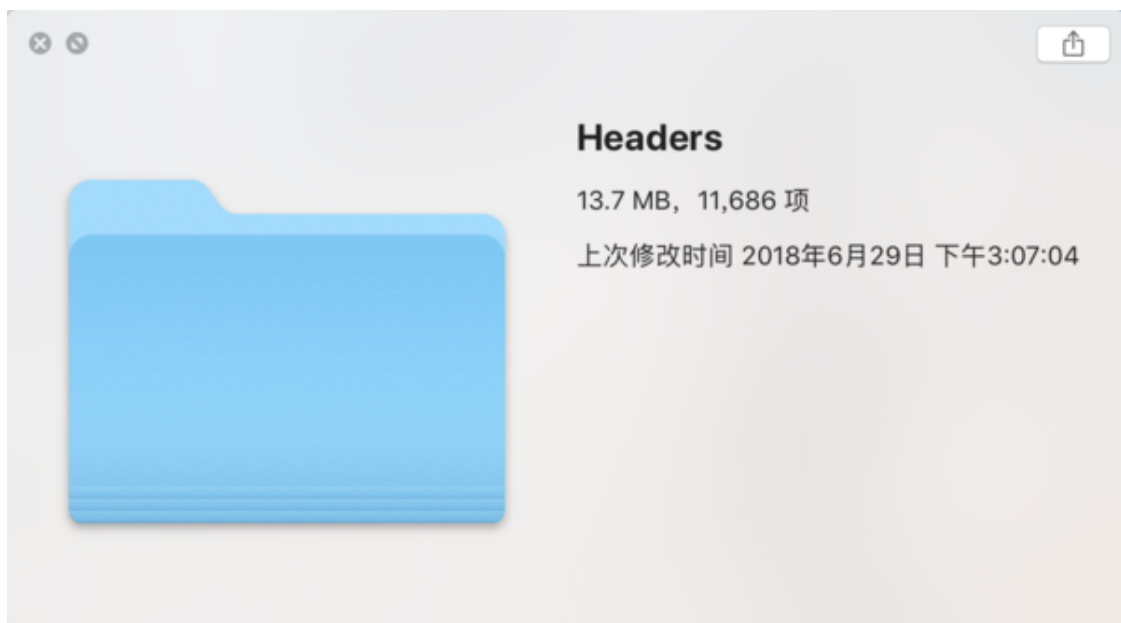
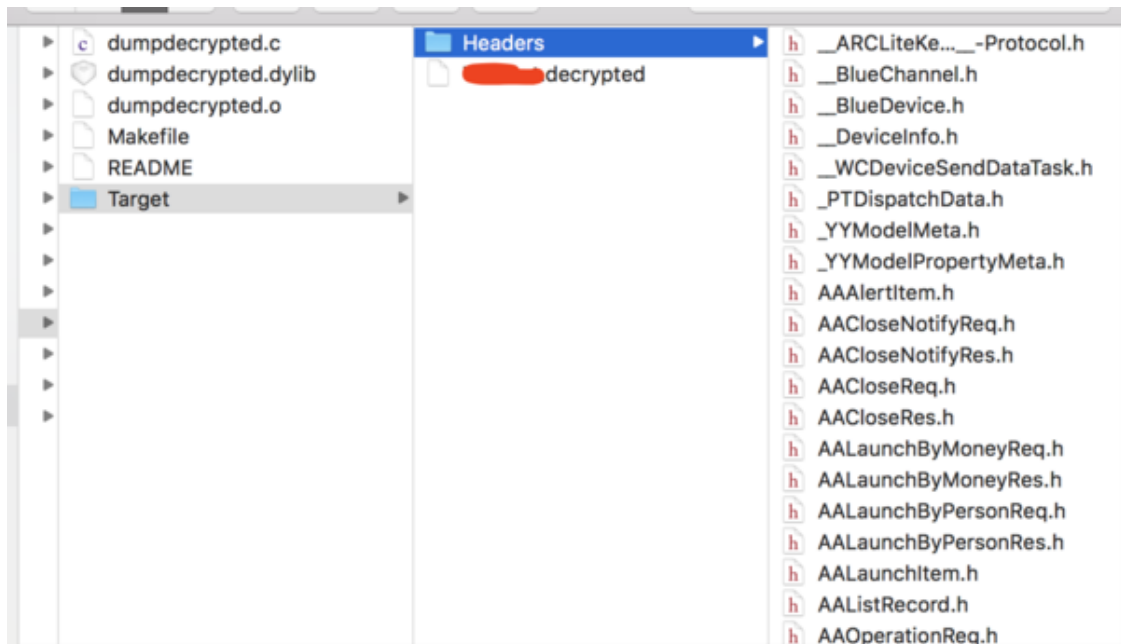


第四步 : class-dump

- 安装 class-dump.. 不赘述
 - dump出TargetApp的头文件:
1. cd 到 TargetApp.decrypted 所在的文件夹;
 2. 执行命令:

```
class-dump --arch arm64 TargetApp.decrypted -H -o ./Headers/
```


然后耐心等待, 执行成功后, 看到 TargetApp 有11686 个 头文件.



目前只学习到了这里, 下周学习后面的内容后再更新.