

目录

前言	1.1
Hopper概览	1.2
下载和安装Hopper	1.3
Hopper使用	1.4
Hopper举例	1.5
AwemeCore	1.5.1
Thunder	1.5.2
打开应用	1.5.2.1
分析逻辑	1.5.2.2
Hopper心得	1.6
Hopper vs IDA	1.7
附录	1.8
参考资料	1.8.1

iOS逆向工具：Hopper

- 最新版本: v1.0.0
- 更新时间: 20231008

简介

介绍iOS逆向的常用工具之前：Hopper。先是Hopper概览；然后是下载和安装，接着是如何使用；并且给出了具体例子，比如AwemeCore、Thunder等；且整理了相关心得；以及和IDA的对比。

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_tool_hopper: iOS逆向工具：Hopper](#)

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逆向工具：Hopper book.crifan.org](#)
- [iOS逆向工具：Hopper crifan.github.io](#)

离线下载阅读

- [iOS逆向工具：Hopper PDF](#)
- [iOS逆向工具：Hopper ePub](#)
- [iOS逆向工具：Hopper 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](mailto:admin@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：2023-10-08 16:18:47

Hopper概览

- Hopper = Hopper Disassembler
 - 偶尔缩写为: hd
 - 是什么: iOS逆向工具
 - Hopper is a reverse engineering tool for OS X and Linux, lets you disassemble, decompile and debug your applications
 - This tool will let you disassemble any binary you want, and provide you all the information about its content, like imported symbols, or the control flow graph! Hopper can retrieve procedural information about the disassembled code like the stack variables, and lets you name all the objects you want.
- 作用: 主要用于二进制的静态逆向分析代码逻辑
 - disassemble
 - decompile
- 对标: IDA
- 支持
 - 运行平台: Mac 、 Linux
 - 目标架构: 32/64bits Intel / Apple Silicon Mac, Linux, Windows and iOS executables
- 主页
 - [Hopper \(hopperapp.com\)](http://hopperapp.com)
- 截图

The screenshot shows the Hopper Disassembler interface with the following details:

- File Information:**
 - Path: /Users/bsr/Desktop/Crypto Tools.a
 - Loader: Mach-O
 - CPU: intel/x86_64
 - Calling Convention: System V
- Instruction Encoding:** 6A 00
- Graphic Views:** Type: Entropy, From: 0, To: 302 136, Cur. Pos.: 302 136
- Format:** Argument: Default, Signed, Negate, Leading Zeros, Type: , Field path: , Manage Types
- Comment:**
- Colors and Tags:**
- Cross References:**
- Procedure:** 4 basic blocks, void func()

The main window displays assembly code for the 'start' procedure:

```

0000000100000c30 db    "System/Library/Frameworks/AppKit.framework/Versions/C/AppK
0000000100000c6e db    0x00 ; '
0000000100000c6f db    0x00 ; '
0000000100000c70 db    3136 dup (0x00)

; Section: text
; RamAddr: 0x1000018b0; 0x10002ed40| (183184 bytes)
; File offset: [6320; 189584[ (183184 bytes)
; Flags: 0x80000400
; S_REGULAR
; S_ATTR_PURE_INSTRUCTIONS
; S_ATTR_SOME_INSTRUCTIONS

; ===== BEGINNING OF PROCEDURE =====
start:
    push    0x0                   ; DATA XREF=0x100000d8
    rbp    ,rsp
    and    rbp, 0xfffffffffffffff0
    mov    rdi, qword [rsi+rpb+8]
    mov    rsi, qword [rsi+rpb+10]
    mov    rdx, 0x0
    add    edx, 0x1
    shl    edx, 0x3
    add    rdx, rs1
    mov    rcx, rdx
    jmp    loc_1000018d5

loc_1000018d1:
    add    rcx, 0x8               ; CODE XREF=start+41
loc_1000018d5:
    cmp    qword [ds:rcx], 0x0     ; CODE XREF=start+31
    jne    loc_1000018d1

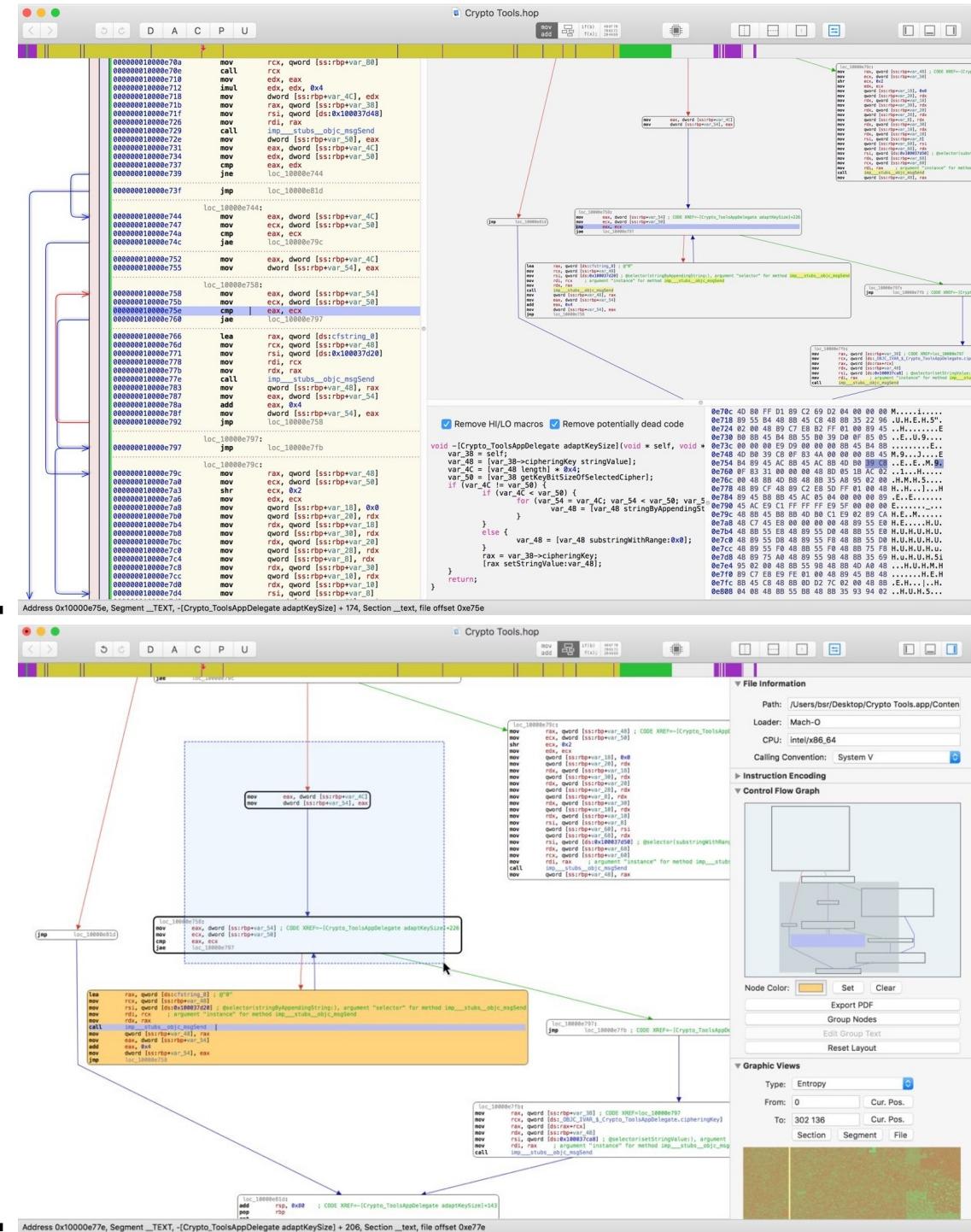
00000001000018d9 48833900
00000001000018d9 75F6
00000001000018db 4883C108
00000001000018d9 E80C000000
00000001000018e4 B9C7
00000001000018e6 EB8CSD0200
00000001000018ec F4
00000001000018eb F4
00000001000018ec db    4 dup (0x90)

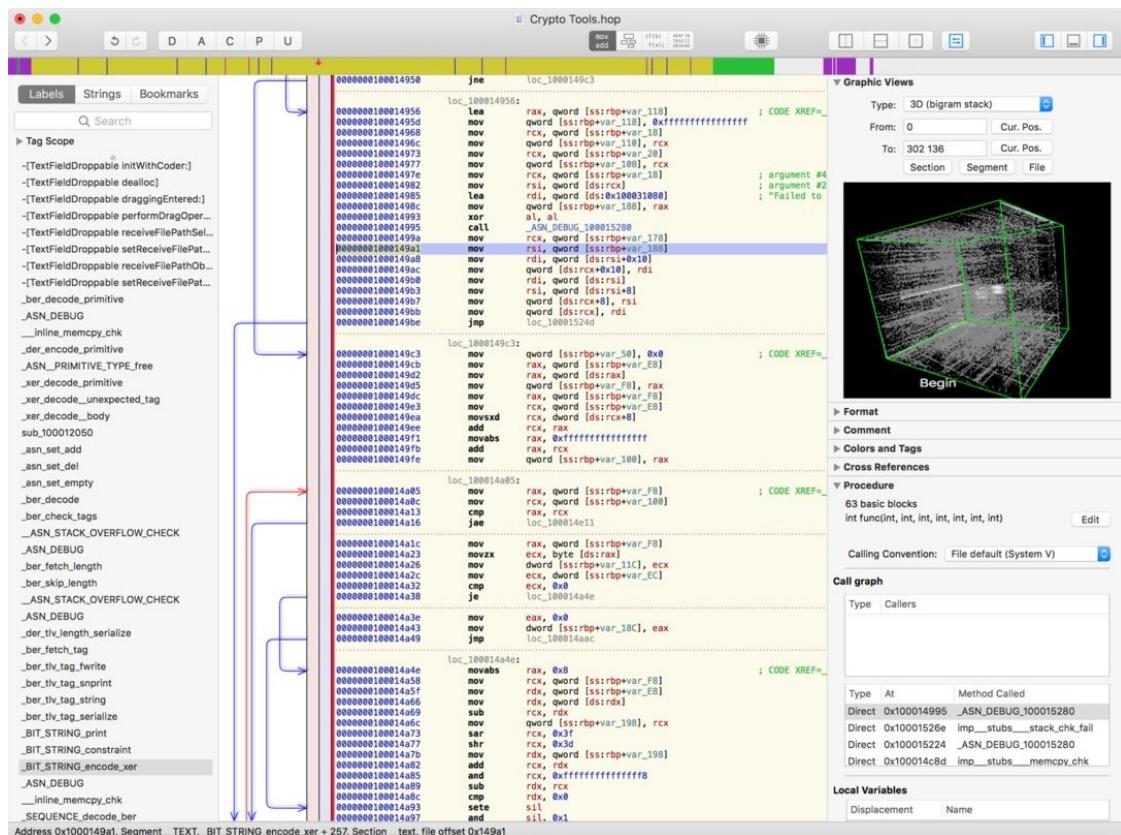
; ===== BEGINNING OF PROCEDURE =====
_main:
    ; Variables:
    ; var_4: -4
    ; var_8: -8
    ; var_10: -16

>>> Python Command

```

Address 0x1000018b0, Segment __TEXT, start + 0, Section __text, file offset 0x18b0





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: 2023-10-08 10:50:22

下载和安装Hopper

请支持正版

警告⚠：此处Hopper破解版仅限于技术研究使用，不准用于非法目的，否则后果自负。

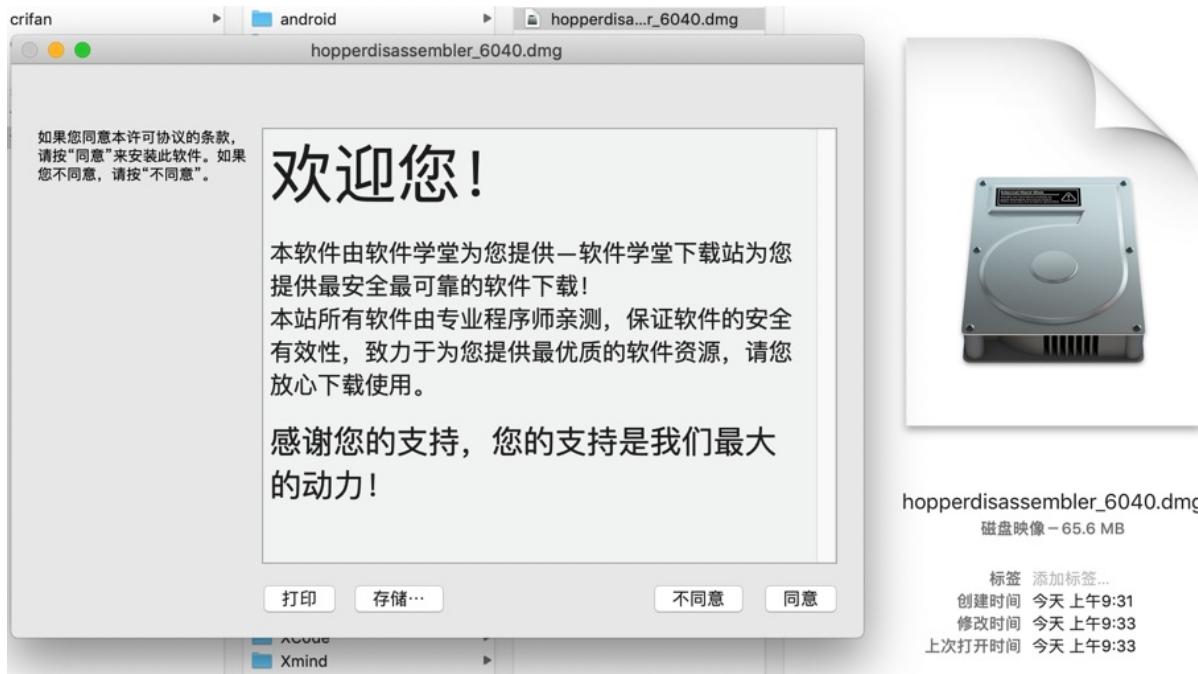
如有侵权，请联系笔者删除。

下载

下载破解版的Hopper Disassembler：

[hopper disassembler for mac v4破解版下载\(免授权文件/序列号\) v4.0.8 - 软件学堂](#)

-> 得到：[hopperdisassembler_6040.dmg](#)



安装

双击 `dmg`，继续，进入界面：

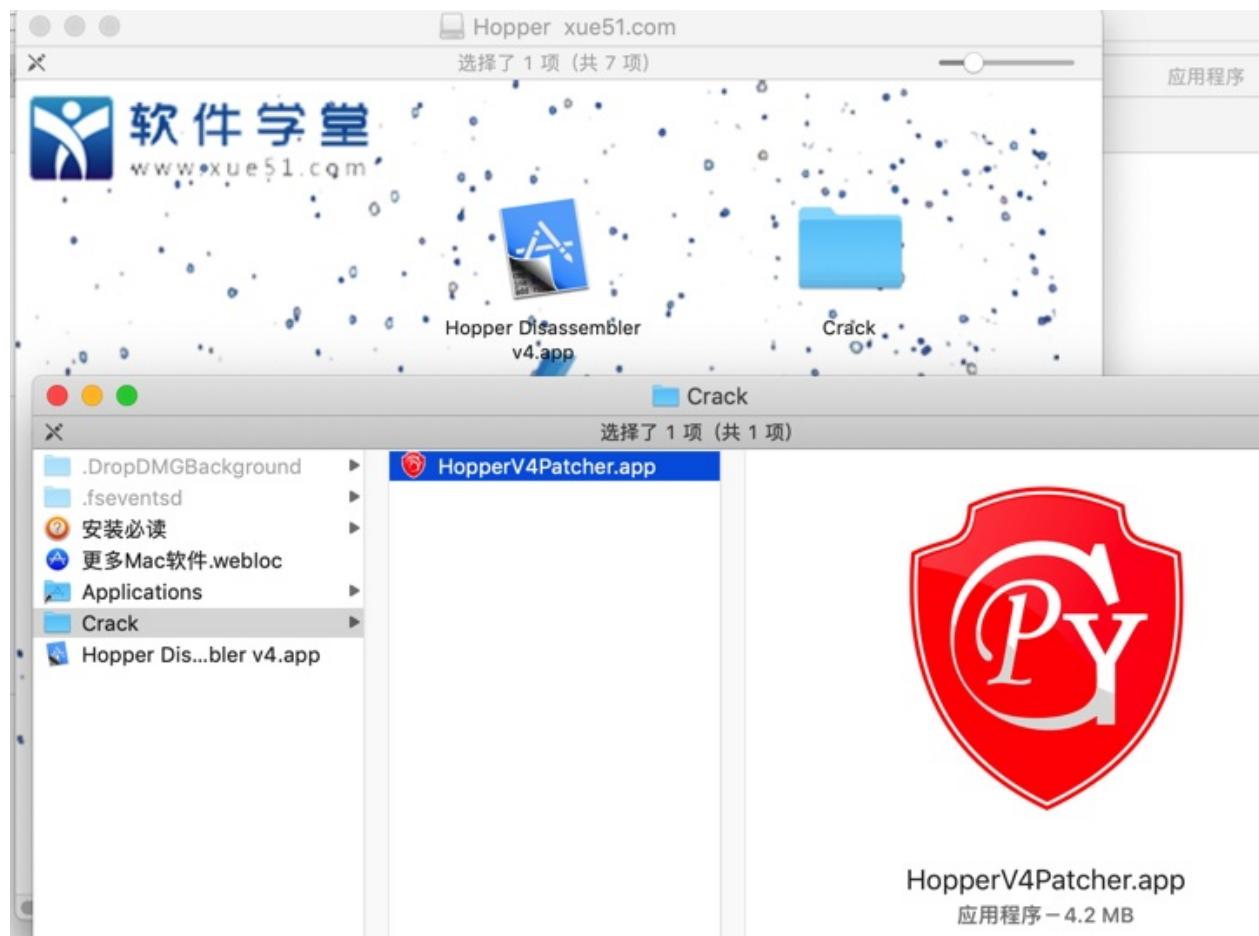


- 注：不要参考教程中说的，双击，否则会直接运行的。

去把 Hopper Disassembler v4.app 拖动到 应用程序 中：



另外 crack 中有个： HopperV4Patcher.app



双击运行，出现提示：

- 不明身份开发者

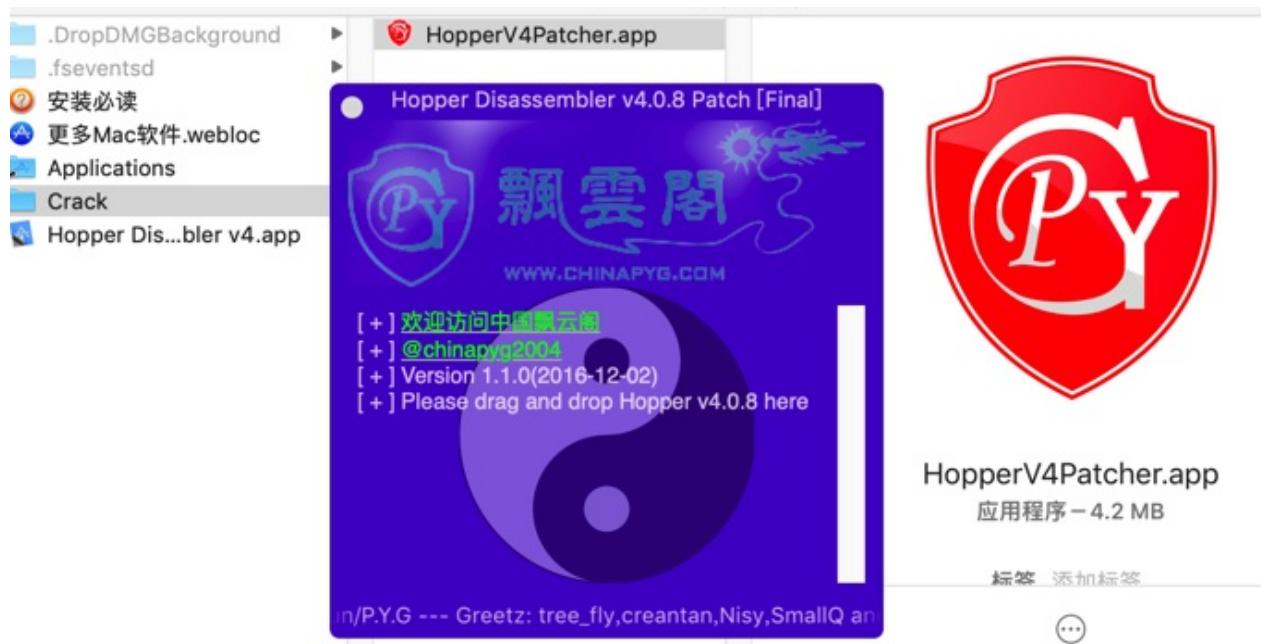
。

去隐私中允许：

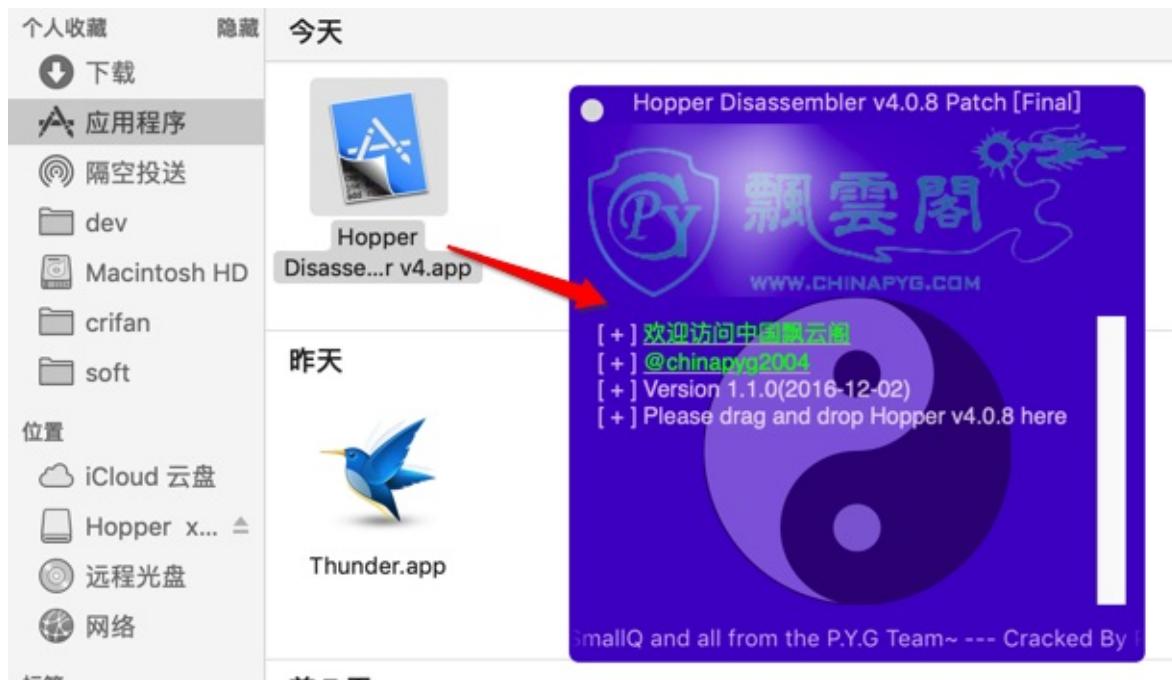
- 仍要打开

。

即可出现界面：



再去把应用程序中的： Hopper Disassembler v4.app 拖动到这个patch界面中：

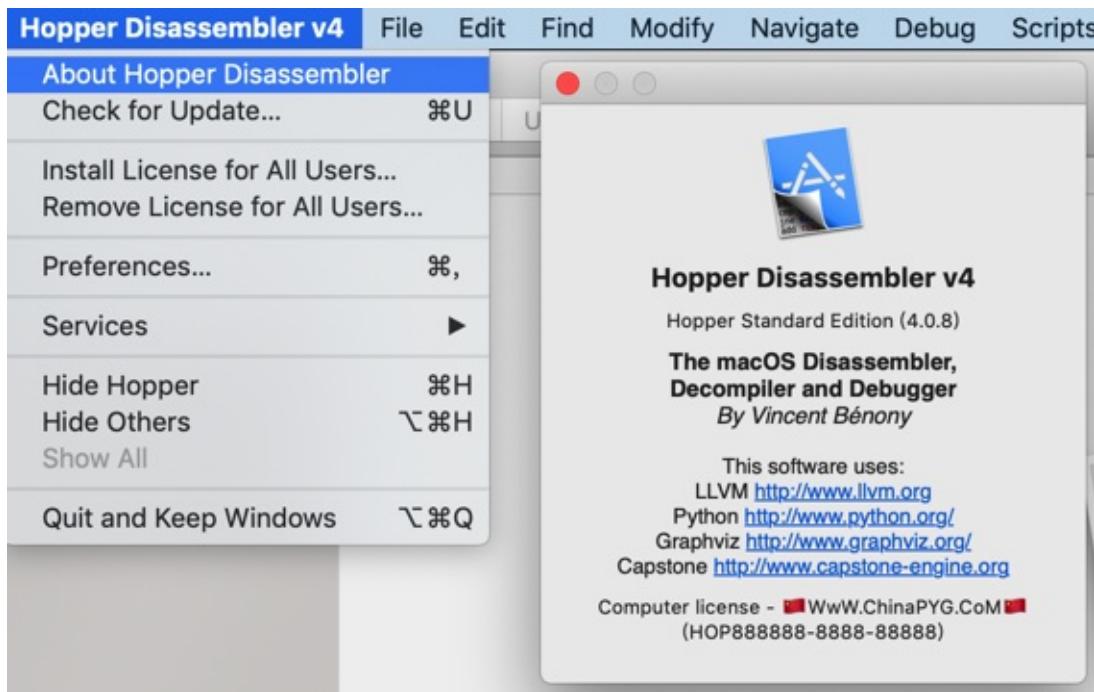


瞬间就破解好了：

- 显示 patch success

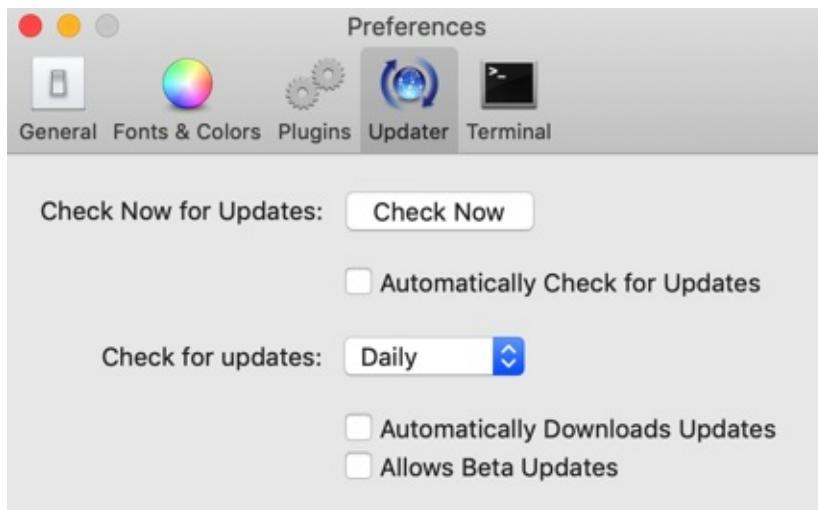
。

如此，即可安装完毕，即可正常打开，已破解的Hopper：



的确没有注册弹框，可以看到 computer license 了，说明破解成功。

另外：为了防止更新导致破解失效，去看看更新设置：

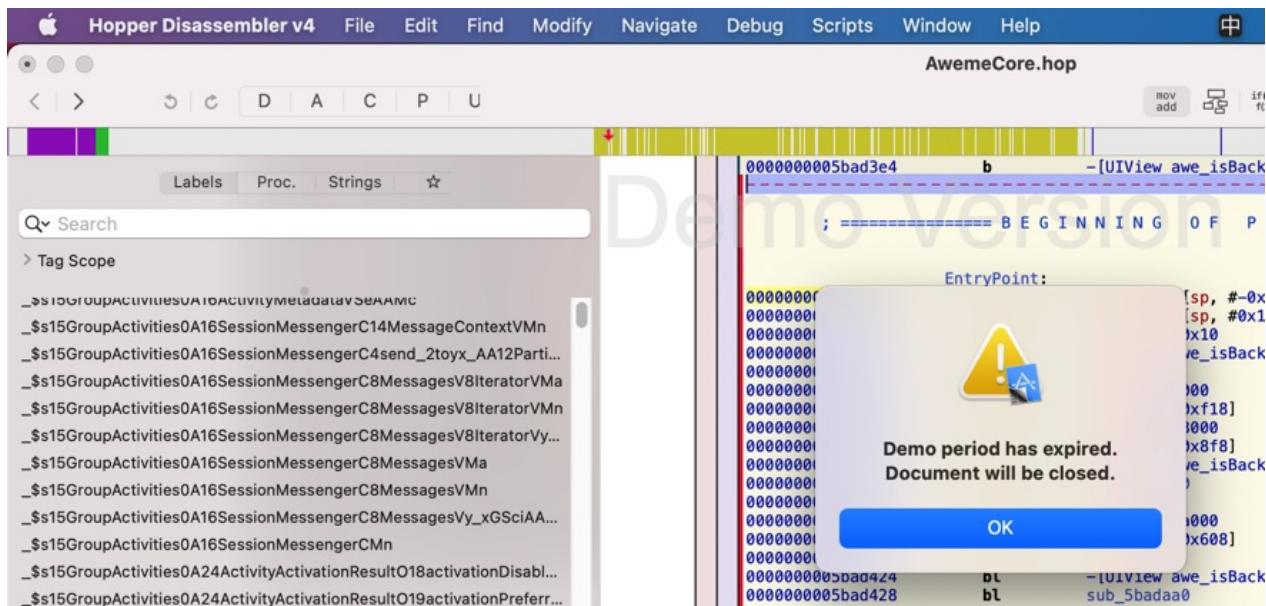


此处已经是关闭自动升级更新了，是我们希望的：不要开启自动更新。

后记：

结果用了几天后，还是会过期，无法继续使用：

Demo period has expired
Document will be closed



点击OK，就退出了。

另外，此处感觉Hopper也比较卡顿，且对于大型app也会卡死，所以放弃。

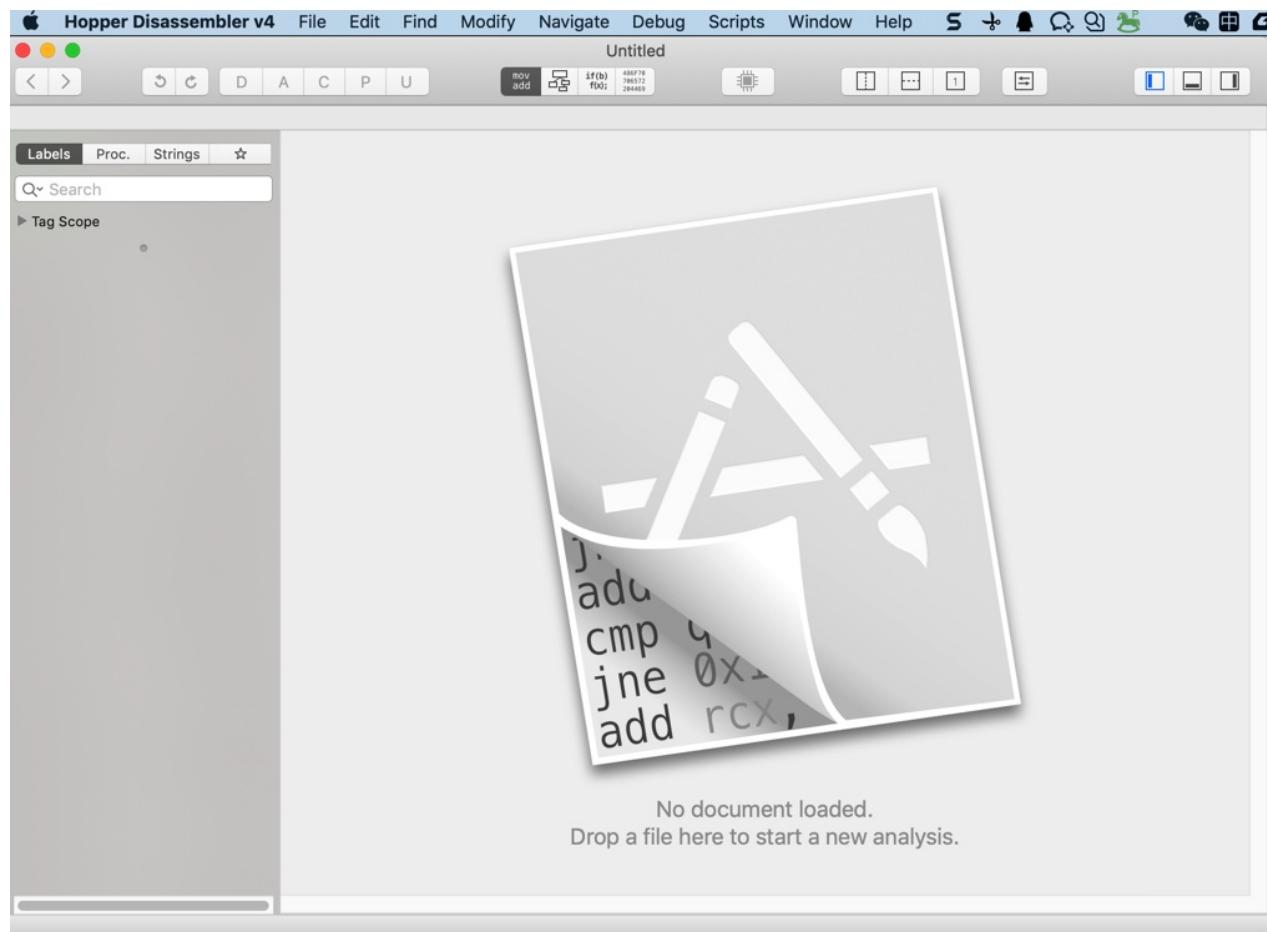
继续转用IDA吧。

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: 2023-10-08 14:27:55

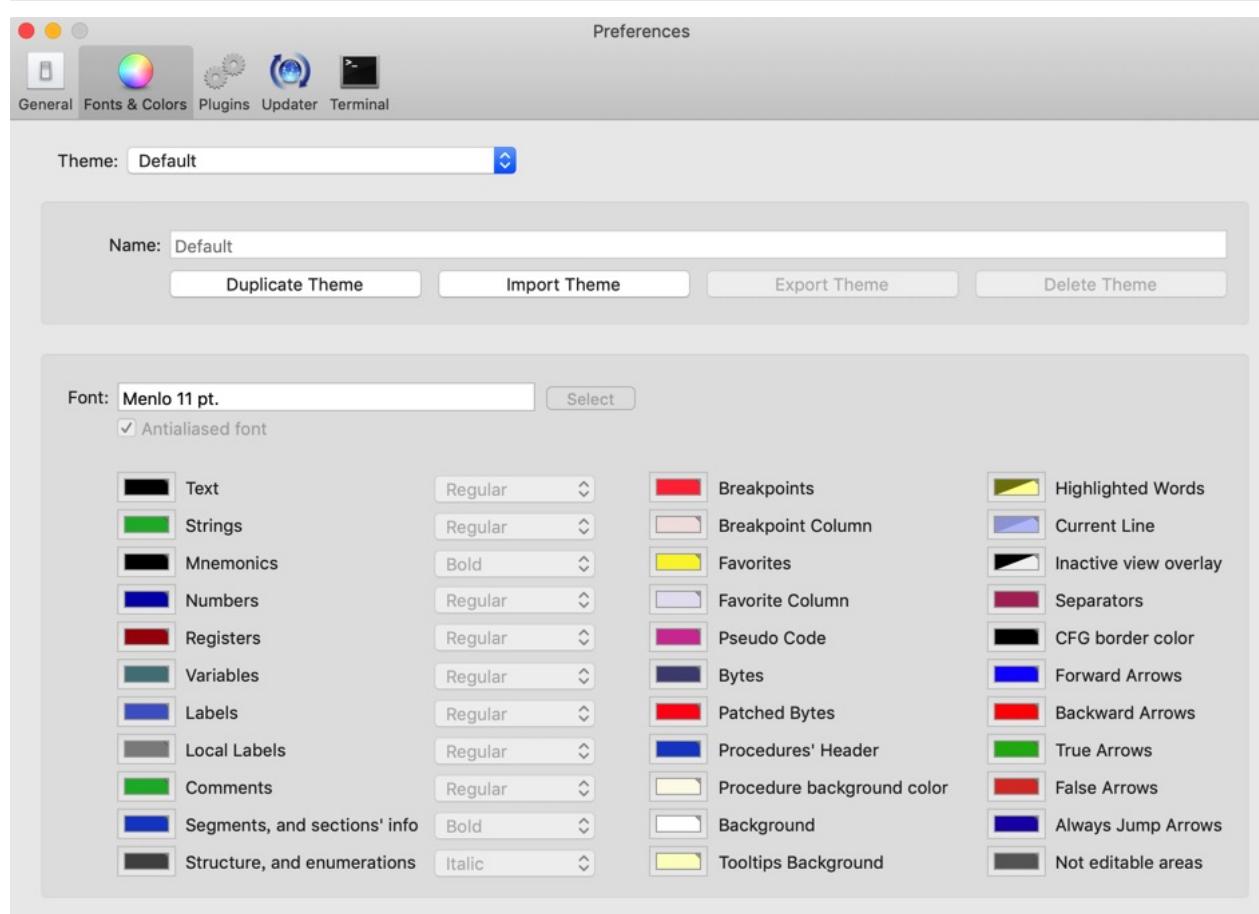
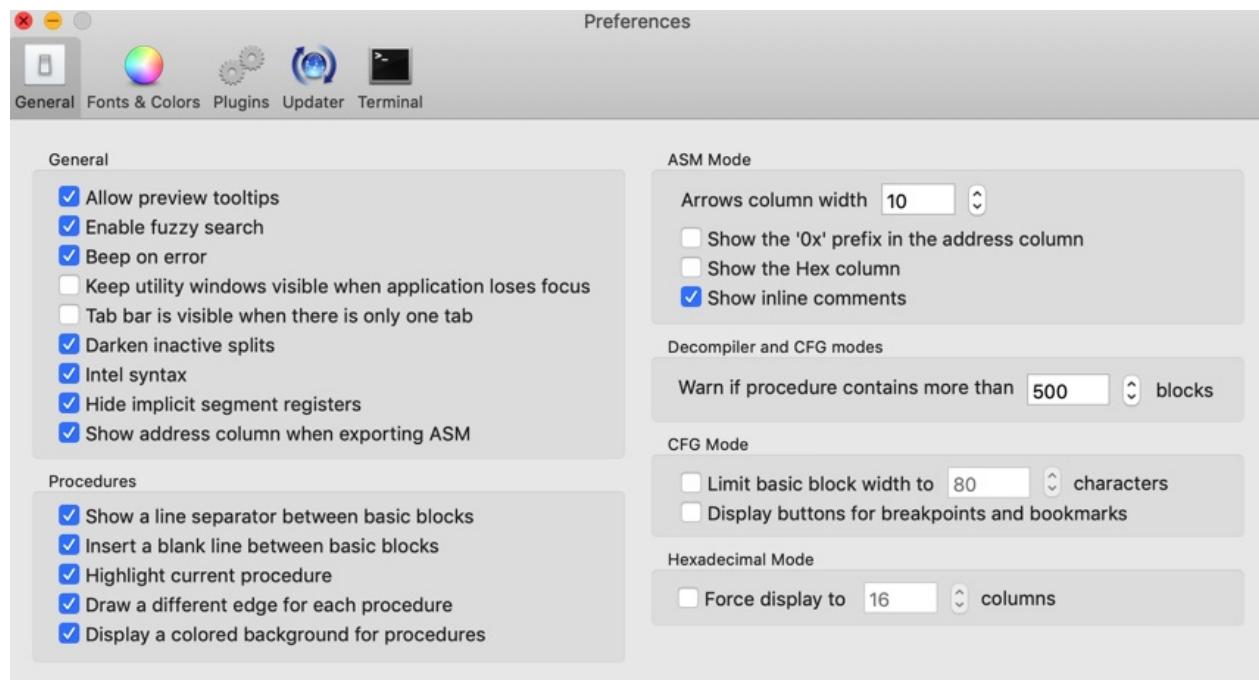
Hopper使用

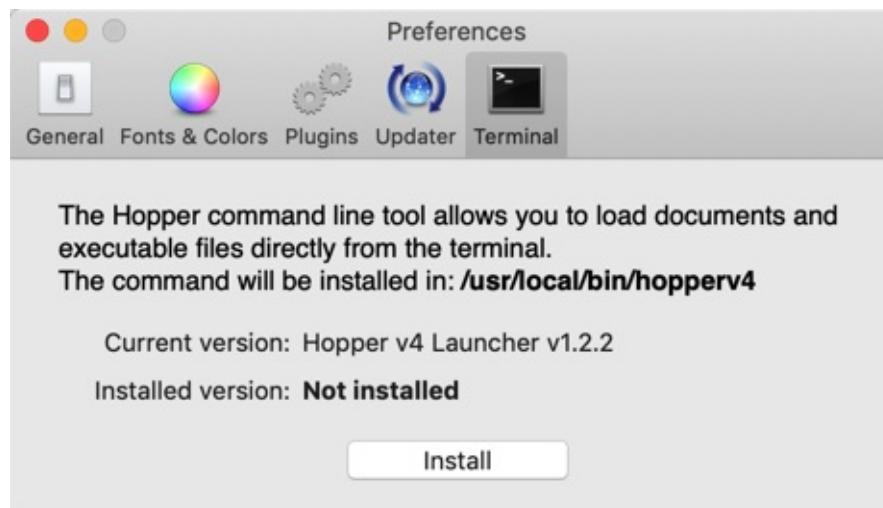
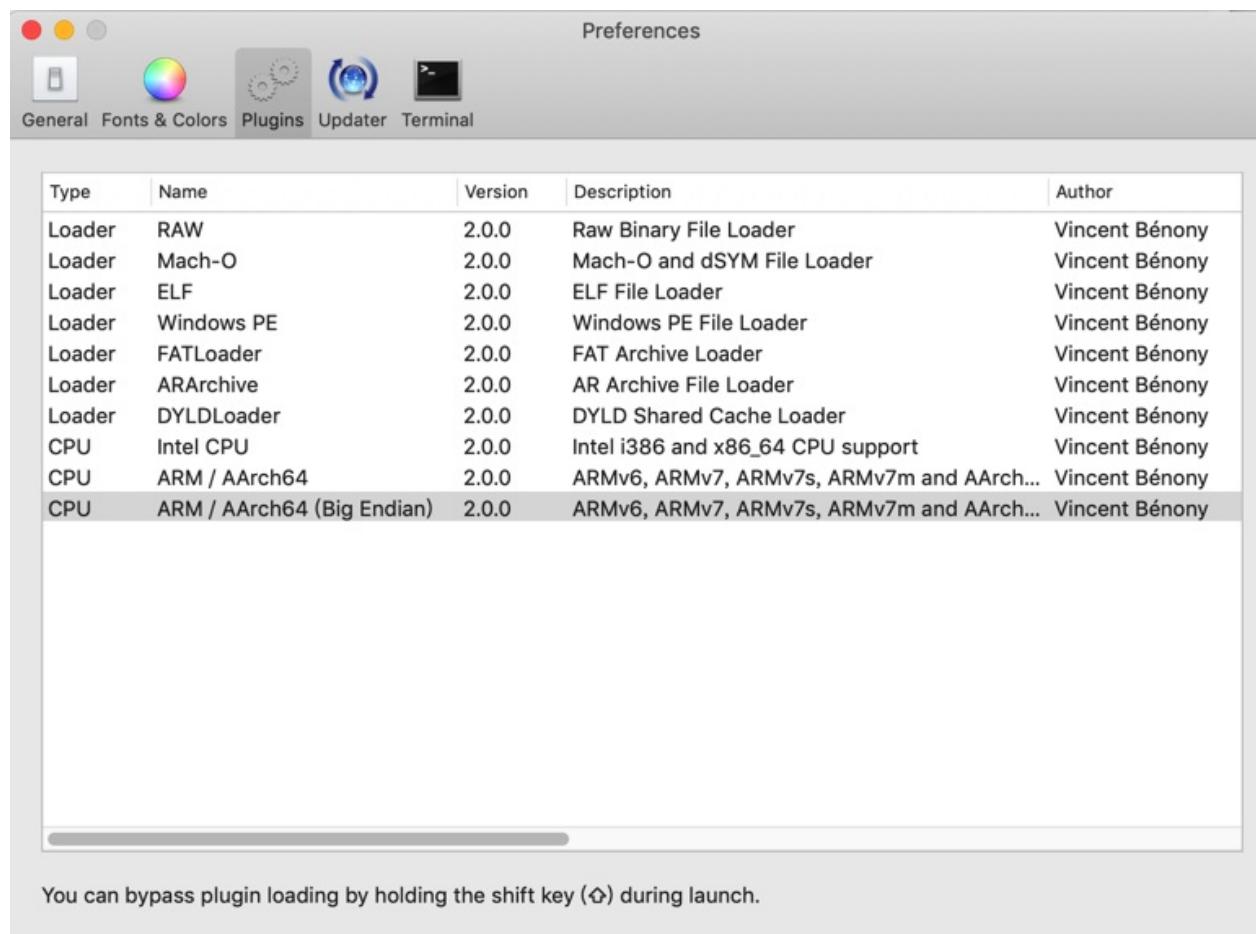
Hopper基本界面

打开后的Hopper主界面是：



Preferences设置界面





Hopper功能介绍



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: 2023-10-08 16:07:32

Hopper举例

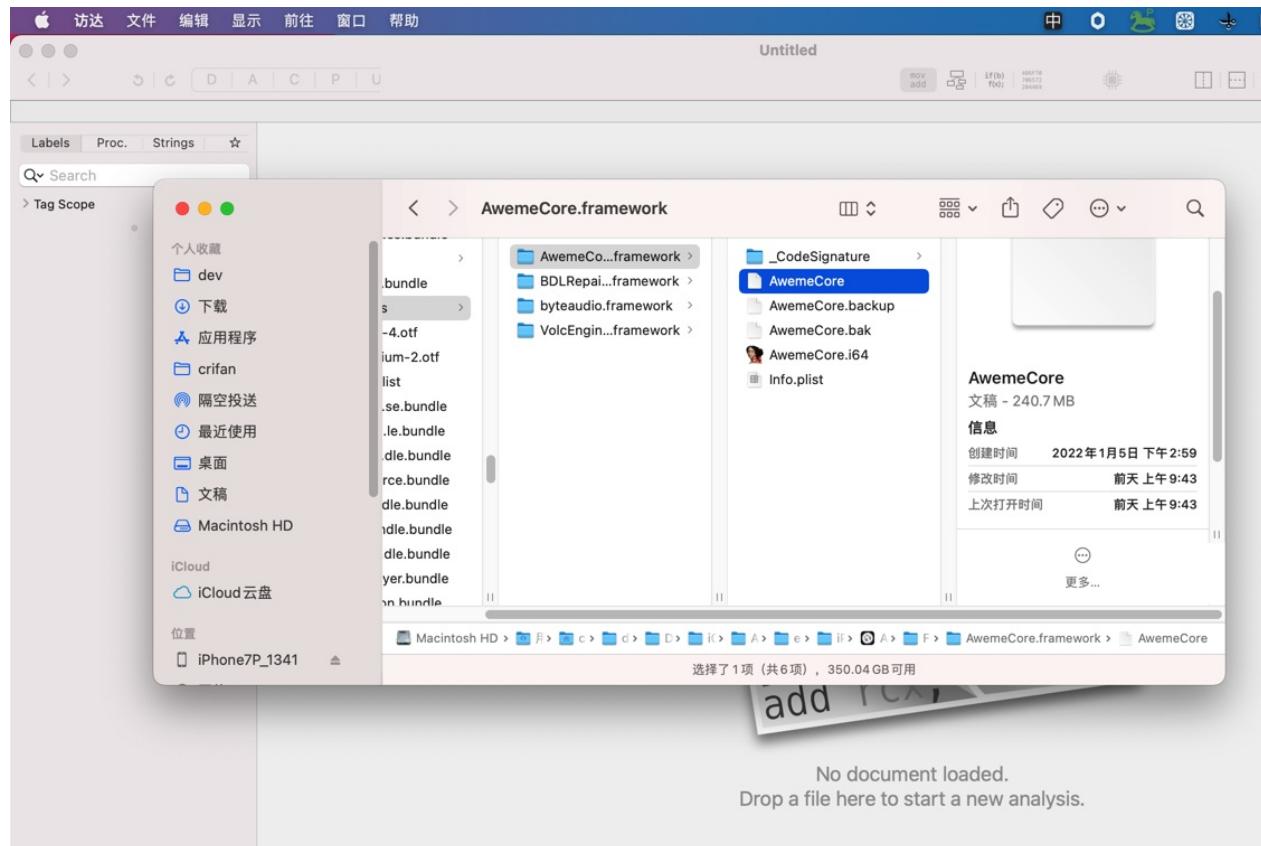
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2023-10-08 11:27:43

Hopper使用举例：AwemeCore

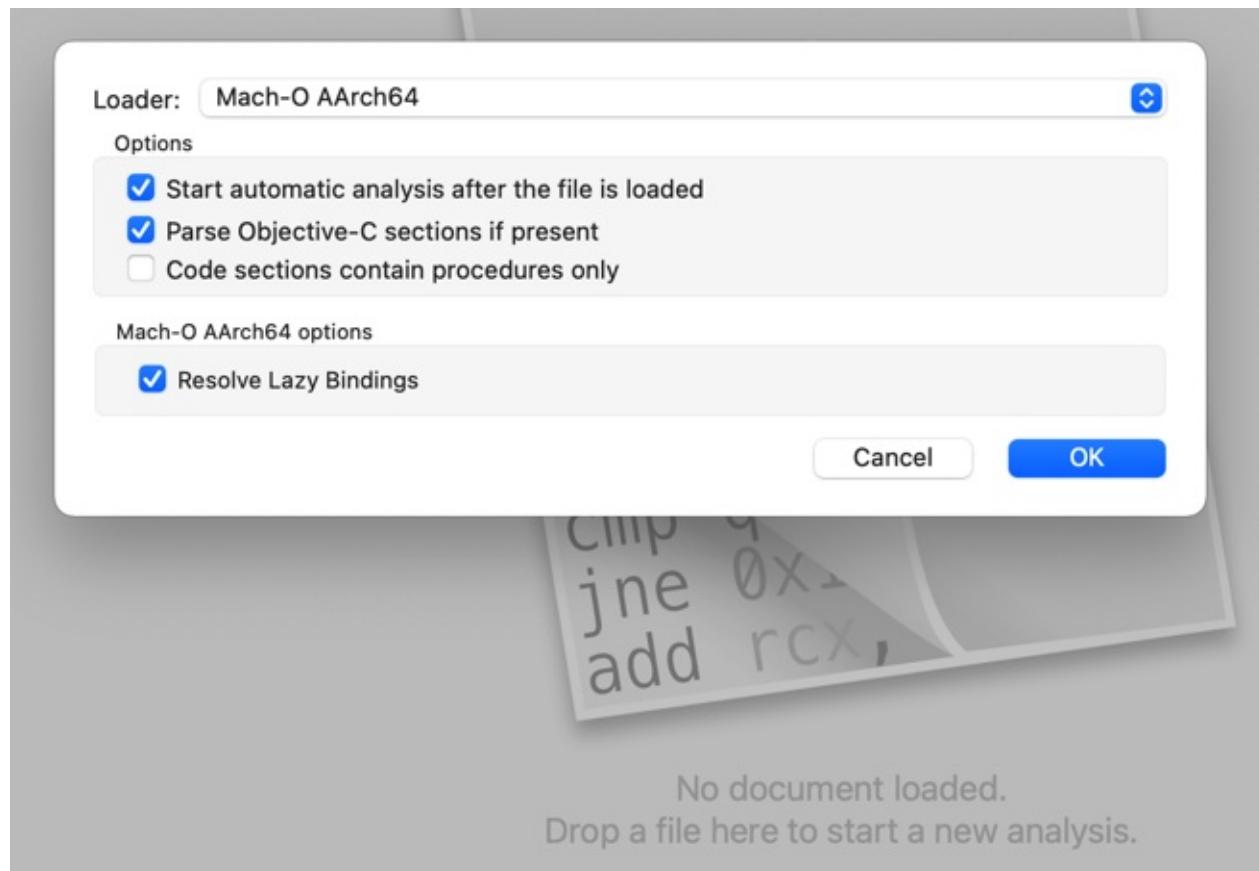
Hopper加载AwemeCore

- 输入文件: AwemeCore

然后把 AwemeCore 拖进 Hopper

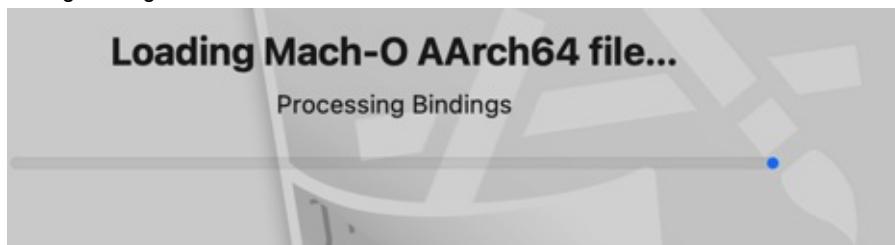


出现Loader弹框：

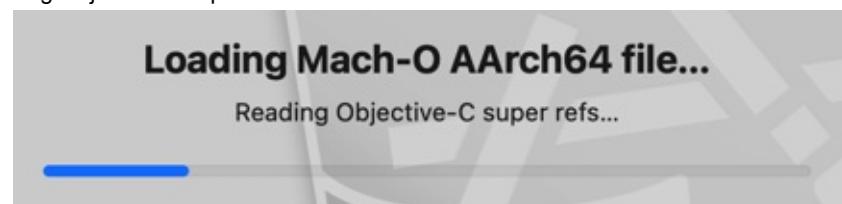


然后开始加载和分析：

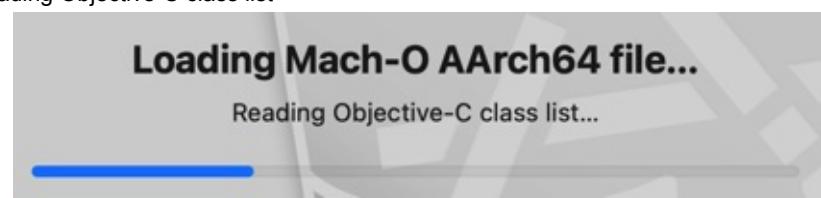
- Loading Mach-O AArch64 file
 - Processing bindings



- Reading Objective-C
 - Reading Objective-C super refs



- Reading Objective-C class list



加载完毕，进入主页面：

```

AwemeCore.hop
; ===== BEGINNING OF PROCEDURE =====

EntryPoint:
0000000005bad3e8    stp    x20, x19, [sp, #-0x20]! ; DATA XREF=0x30c9d0
0000000005bad3e9    stp    x20, x30, [sp, #0x10]
0000000005bad3e9    add    x20, sp, #0x10
0000000005bad3f0    bl    -[UIView awe_isBackground]+1529944
0000000005bad3f1    mov    x19, x0
0000000005bad3f2    adrp   x8, #0x947000
0000000005bad3f3    ldr    x8, [x8, #0xf18]
0000000005bad3f4    adrp   x8, #0x3893000
0000000005bad3f5    ldr    x1, [x8, #0x8f8]
0000000005bad3f6    bl    -[UIView awe_isBackground]+1529720
0000000005bad3f7    ldp    x20, x19, [sp], #0x20
0000000005bad3f8    b     loc_d523fb8
0000000005bad3f9    db    0x4f ; 'z'
0000000005bad3fa    db    0x4f ; '0'
0000000005bad3f9    db    0xbe ; '!'
0000000005bad410    db    0x99 ; ' '
0000000005bad411    db    0x19 ; '9'
0000000005bad412    db    0x10 ; '0'
0000000005bad413    db    0x43 ; 'C'
0000000005bad414    db    0x00 ; ' '
0000000005bad415    db    0x91 ; '1'
0000000005bad416    db    0x88 ; '8'
0000000005bad417    db    0x99 ; ' '
0000000005bad418    db    0x00 ; ' '
0000000005bad419    db    0x91 ; '1'
0000000005bad420    db    0x88 ; '8'
0000000005bad421    db    0x99 ; ' '
0000000005bad422    db    0x00 ; ' '
0000000005bad423    db    0x19 ; '9'
0000000005bad424    db    0x10 ; '0'
0000000005bad425    db    0x43 ; 'C'
0000000005bad426    db    0x00 ; ' '
0000000005bad427    db    0x91 ; '1'
0000000005bad428    db    0x88 ; '8'
0000000005bad429    db    0x99 ; ' '
0000000005bad430    db    0x00 ; ' '
0000000005bad431    db    0x19 ; '9'
0000000005bad432    db    0x10 ; '0'
0000000005bad433    db    0x43 ; 'C'
0000000005bad434    db    0x00 ; ' '
0000000005bad435    db    0x91 ; '1'
0000000005bad436    db    0x88 ; '8'
0000000005bad437    db    0x99 ; ' '
0000000005bad438    db    0x00 ; ' '
0000000005bad439    db    0x19 ; '9'
0000000005bad440    db    0x10 ; '0'
0000000005bad441    db    0x43 ; 'C'
0000000005bad442    db    0x00 ; ' '
0000000005bad443    db    0x91 ; '1'
0000000005bad444    db    0x88 ; '8'
0000000005bad445    db    0x99 ; ' '
0000000005bad446    db    0x00 ; ' '
0000000005bad447    db    0x91 ; '1'
0000000005bad448    db    0x88 ; '8'
0000000005bad449    db    0x99 ; ' '
0000000005bad450    db    0x00 ; ' '
0000000005bad451    db    0x91 ; '1'
0000000005bad452    db    0x88 ; '8'
0000000005bad453    db    0x99 ; ' '
0000000005bad454    db    0x00 ; ' '
0000000005bad455    db    0x69 ; '9'
0000000005bad456    db    0x47 ; 'G'
0000000005bad457    db    0x79 ; ' '
0000000005bad458    db    0x00 ; ' '

```

缺点：目前总体有点卡顿

分析代码逻辑

接下来，就是如何具体分析逻辑了

```

AwemeCore.hop
; ===== BEGINNING OF PROCEDURE =====

sub_5bad3dc:
0000000005bad3dc    mov    x0, x19 ; CODE XREF=sub_5bad29c+108, sub_5bad29c+148
0000000005bad3e0    mov    x1, x21
0000000005bad3e4    b     -[UIView awe_isBackground]+1529720
; ===== BEGINNING OF PROCEDURE =====

EntryPoint:
0000000005bad3e8    stp    x20, x19, [sp, #-0x20]! ; DATA XREF=0x30c9d0
0000000005bad3e9    stp    x20, x30, [sp, #0x10]
0000000005bad3e9    add    x20, sp, #0x10
0000000005bad3f0    bl    -[UIView awe_isBackground]+1529944
0000000005bad3f1    mov    x19, x0
0000000005bad3f2    adrp   x8, #0x947000
0000000005bad3f3    ldr    x8, [x8, #0xf18]
0000000005bad3f4    adrp   x8, #0x3893000
0000000005bad3f5    ldr    x1, [x8, #0x8f8]
0000000005bad3f6    bl    -[UIView awe_isBackground]+1529720
0000000005bad3f7    ldp    x20, x19, [sp], #0x20
0000000005bad3f8    b     loc_d523fb8
0000000005bad3f9    db    0x4f ; 'z'
0000000005bad3fa    db    0x4f ; '0'
0000000005bad3f9    db    0xbe ; '!'
0000000005bad410    db    0x99 ; ' '
0000000005bad411    db    0x19 ; '9'
0000000005bad412    db    0x10 ; '0'
0000000005bad413    db    0x43 ; 'C'
0000000005bad414    db    0x00 ; ' '
0000000005bad415    db    0x91 ; '1'
0000000005bad416    db    0x88 ; '8'
0000000005bad417    db    0x99 ; ' '
0000000005bad418    db    0x00 ; ' '
0000000005bad419    db    0x91 ; '1'
0000000005bad420    db    0x88 ; '8'
0000000005bad421    db    0x99 ; ' '
0000000005bad422    db    0x00 ; ' '
0000000005bad423    db    0x19 ; '9'
0000000005bad424    db    0x10 ; '0'
0000000005bad425    db    0x43 ; 'C'
0000000005bad426    db    0x00 ; ' '
0000000005bad427    db    0x91 ; '1'
0000000005bad428    db    0x88 ; '8'
0000000005bad429    db    0x99 ; ' '
0000000005bad430    db    0x00 ; ' '
0000000005bad431    db    0x19 ; '9'
0000000005bad432    db    0x10 ; '0'
0000000005bad433    db    0x43 ; 'C'
0000000005bad434    db    0x00 ; ' '
0000000005bad435    db    0x91 ; '1'
0000000005bad436    db    0x88 ; '8'
0000000005bad437    db    0x99 ; ' '
0000000005bad438    db    0x00 ; ' '
0000000005bad439    db    0x19 ; '9'
0000000005bad440    db    0x10 ; '0'
0000000005bad441    db    0x43 ; 'C'
0000000005bad442    db    0x00 ; ' '
0000000005bad443    db    0x91 ; '1'
0000000005bad444    db    0x88 ; '8'
0000000005bad445    db    0x99 ; ' '
0000000005bad446    db    0x00 ; ' '
0000000005bad447    db    0x91 ; '1'
0000000005bad448    db    0x88 ; '8'
0000000005bad449    db    0x99 ; ' '
0000000005bad450    db    0x00 ; ' '
0000000005bad451    db    0x91 ; '1'
0000000005bad452    db    0x88 ; '8'
0000000005bad453    db    0x99 ; ' '
0000000005bad454    db    0x00 ; ' '
0000000005bad455    db    0x69 ; '9'
0000000005bad456    db    0x47 ; 'G'
0000000005bad457    db    0x79 ; ' '
0000000005bad458    db    0x00 ; ' '

```

此处可以看到函数名： awe_isBackground

双击后，跳转到函数实现：

```

AwemeCore.hop
Labels Proc. Strings ⌂
Q Search
> Tag Scope
+[UIFont tc21_pingFangSCMediu...
+[UIFont tc21_pingFangSCsemib...
+[UIFont tc21_fontWithNamesize:]
-[UIImageViewView_awa_addBezierPat...
-[UIViewController_awa_exceptio...
-[UIViewController setAwe_excep...
-[CALayer flex_node]
-[CALayer flex_makeLayout:]
-[UIView flex_node]
-[UIView flex_makeConstraints:]
-[NSArray mus_stringWithIndex:]
-[NSArray mus_dictionaryWithInd...
-[NSMutableArray mus_addObjec...
-[NSArray mus_JSONString]
-[NSArray mus_JSONStringWith...
-[NSBundle setFakeBundleID:]
-[NSString awe_stringWithMaxim...
-[NSString awe_stringWithMaxim...
+[NSURLPatternString]
+[NSString awe_hashTagURLPatt...
+[NSString awe_phonePatternStr...
-[NSString safeStringByReplacin...
+[NSURL HTS_URLWithString:par...
+[NSURL URLWithStringWithPar...
sub_d2bbfe0
sub_d2bce88
sub_d2bcf1c
sub_d2bcf40
sub_d2bcf70
EntryPoint_1931
-[UIView awe_isBackground]
sub_d524038
sub_d5303cc
sub_d5303d0
sub_d5303d4
sub_d5303d8
Address 0xd3ae774, Segment __BD_TEXT, -[UIView awe_isBackground] + 0, Section __text, file offset 0x8a0e774

```

点击 尝试别人说的，切换到 伪代码 ObjC 的

```

AwemeCore.hop
P U
if(b) f(x);
Pseudo-code mode
00000000d3ae770 db 0xa8 ; ...
00000000d3ae771 db 0x00 ; ...
00000000d3ae772 db 0xf7 ; ...
00000000d3ae773 db 0x14 ; ...
-[UIView awe_isBackground]:
00000000d3ae774 db 0xf4 ; ...
00000000d3ae775 db 0x4f ; '0'

```

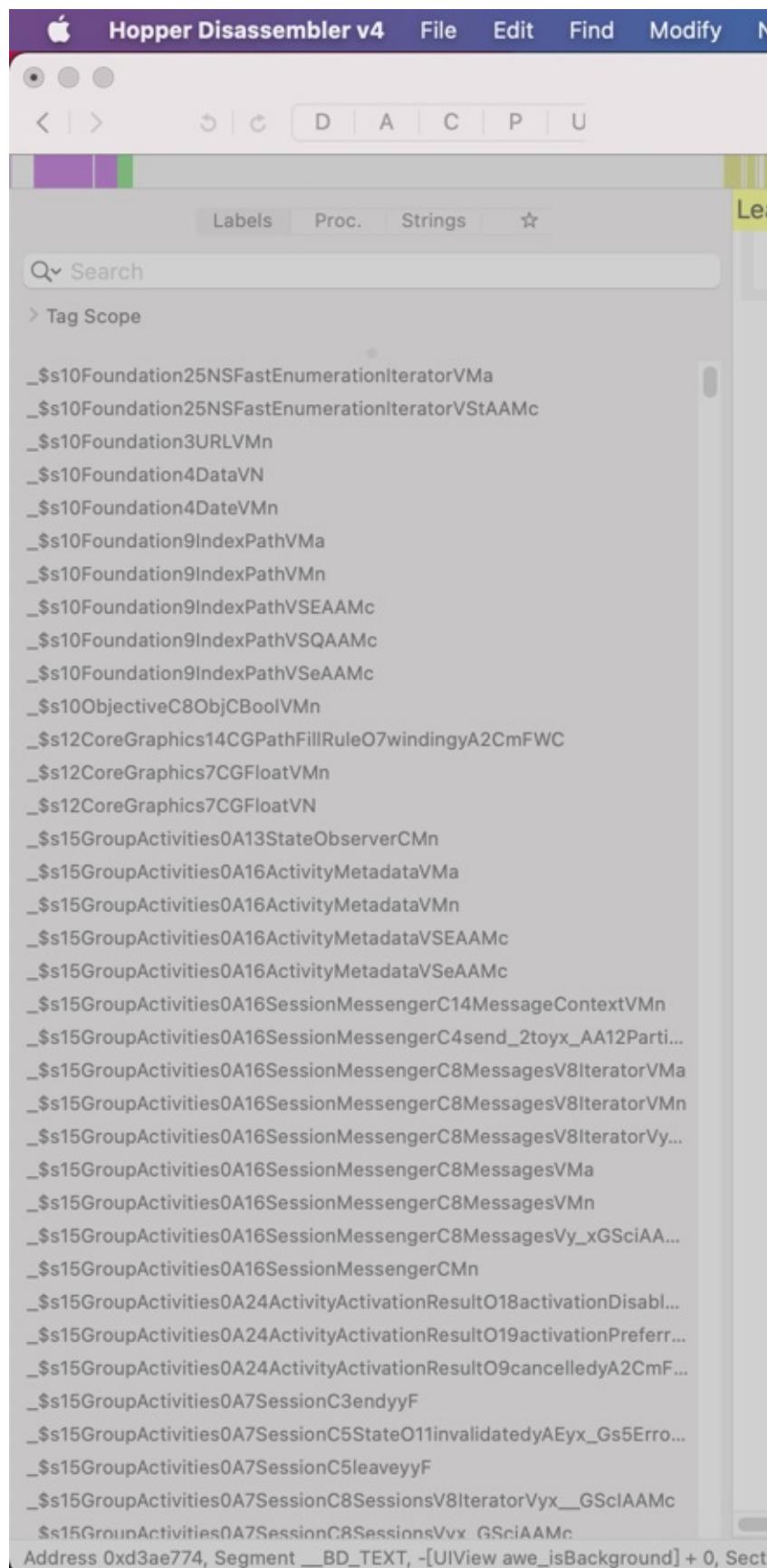
此处出现警告：

- No procedure at this address

◦

继续去找其他逻辑：比如，是否有越狱相关内容。

从左边的函数列表：



找找哪些和启动相关的、初始化相关的

比如之前看到的，靠近entry的 `AWELaunchMainPlaceholder` 这种函数：

```

AwemeCore.hop

; ===== BEGINNING OF PROCEDURE =
sub_5bad3dc:
    mov    x0, x19
    mov    x1, x21
    b     -[UIView awe_isBackground]+1529720

; ===== BEGINNING OF PROCEDURE =
EntryPoint:
    stp    x20, x19, [sp, #-0x20]!
    stp    x29, x30, [sp, #0x10]
    add    x29, sp, #0x10
    bl    -[UIView awe_isBackground]+1529944
    mov    x19, x0
    adrp   x8, #0x947000
    ldr    x0, [x8, #0xf18]
    adrp   x8, #0x3893000
    ldr    x1, [x8, #0x8f8]
    bl    -[UIView awe_isBackground]+1529720
    sub   sub_5bada60
    bl    x20, x0
    adrp   x8, #0x38aa000
    ldr    x1, [x8, #0x608]
    movz   w2, #0x3
    bl    -[UIView awe_isBackground]+1529720
    sub   sub_5badaa0
    bl    x0, x19
    ldp   x29, x30, [sp, #0x10]
    ldp   x20, x19, [sp]!, #0x20
    b     loc_d523fb8

; ===== BEGINNING OF PROCEDURE =
sub_5bad43c:
    stp    x20, x19, [sp, #-0x20]!
    stp    x29, x30, [sp, #0x10]
    add    x29, sp, #0x10
    adrp   x8, #0x947000
    ldr    x19, [x8, #0xea8]
    nop
    ldr    x0, [x8, #0xed0]
    adrp   x8, #0x38aa000
    ldr    x1, [x8, #0x310]
    bl    -[UIView awe_isBackground]+1529720
    mov    x2, x0
    adrp   x8, #0x38aa000
    ldr    x1, [x8, #0x568]
    mov    x0, x19
    ldp   x29, x30, [sp, #0x10]
    ldp   x20, x19, [sp]!, #0x20

EntryPoint
+[AWELaunchMainPlaceholder _g...]
sub_5bada60
Address 0x5bad438, Segment __BD_TEXT, EntryPoint + 80, Section __text, file offset 0x120d438

```

就很值得好好研究看看

另外继续研究越狱相关：

继续研究左边函数列表：

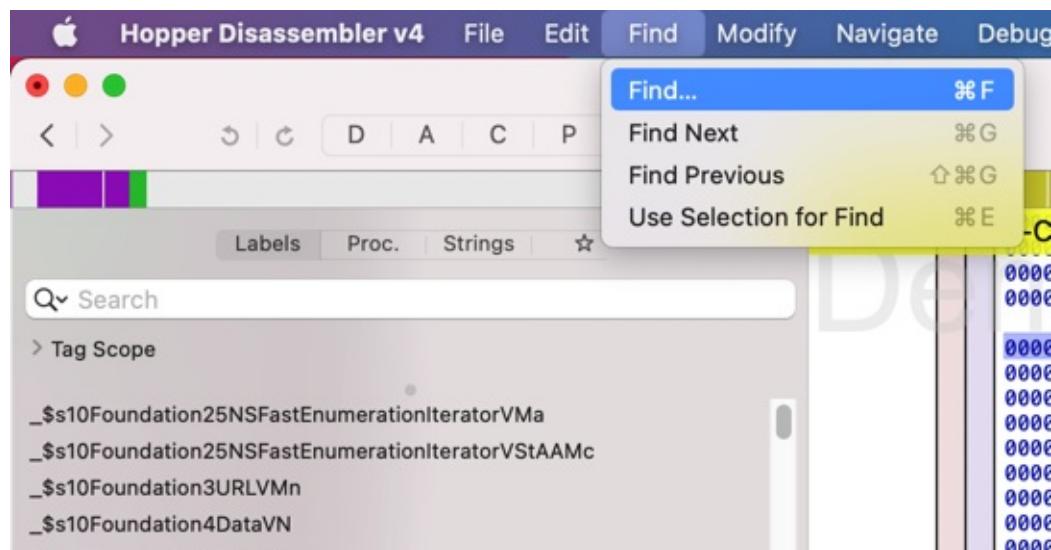


The screenshot shows the Xcode debugger interface with the assembly dump view selected. The assembly code is displayed in a scrollable list, showing various Objective-C method implementations. At the bottom of the assembly dump, there is a memory dump window showing the memory address 0xd3ae774, segment __BD_TEXT, and the instruction -[UIView awe_isBackground] + 0, followed by some assembly code.

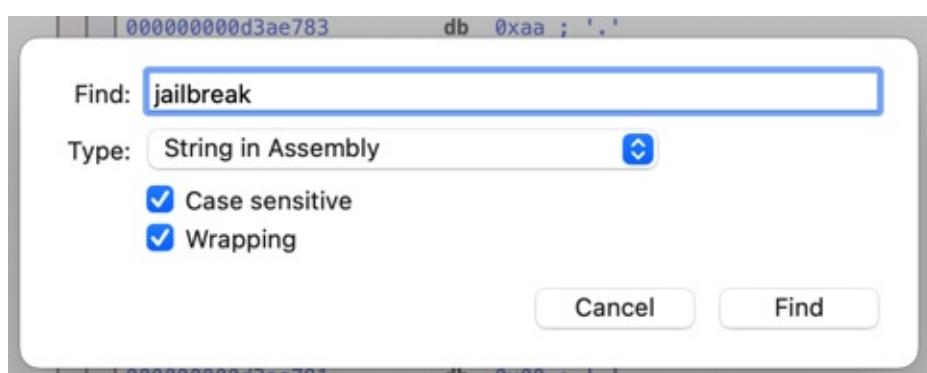
```
-[UIImage awe_generateUserAvatarPublishImage]
-[UIImage awe_generateUserAvatarPublishImageWithSize:]
-[UIImage awe_generateSizeFixedUserAvatarPublishImage]
-[UIImage awe_generateProfileCoverPublishImage]
-[UIImage awe_generateUserAvatarPublishImageWithNickName:locatio...
-[UIImage awe_drawText:withFontSize:weight:atHeightProportion:]
-[UIImage awe_drawText:withFontSize:weight:atHeightProportion:toTh...
-[UIView afd_colorCircleLottieView]
-[UIView setAfd_colorCircleLottieView:]
-[UIView afd_showColorCircleWithLottieAnimation:radius:bundle:]
-[UIView afd_hideColorCircle]
-[UIViewController isIronManTransitionContainer]
+[UIImage awehg_timorImageNamed:]
EntryPoint_1932
-[NSURLRequest setIgnoreErrorTips:]
-[NSURLRequest ignoreErrorTips]
-[NSURLRequest needCommonParams]
-[NSURLRequest setNeedCommonParams:]
-[NSURLRequest requestSerializerClass]
-[NSURLRequest setRequestSerializerClass:]
-[NSArray my.httpClientEncode]
-[UIView im_showUnreadViewAtPoint:frame:]
-[UIView im_showUnreadViewAtPoint:]
-[UIView im_showUnreadViewWithUnreadCount:atPoint:]
-[UIView im_showUnreadViewWithUnreadCount:atPoint:dotColor:]
-[UIView im_showUnreadViewWithText:atPoint:bgColor:textColor:]
-[UIView im_showCustomUnreadView:withFrame:]
-[UIView im_hideUnreadView]
-[UIView im_isUnreadDotShown]
-[UIView im_updateDotColor:]
-[UIView im_updateDotViewLocation:]
-[UIView im_updateCountDotViewLocation:]
-[UIView im_dotView]
-[UIView setIm_dotView:]
-[UIView im_countDotView]
-[UIView setIm_countDotView:]
```

Address 0xd3ae774, Segment __BD_TEXT, -[UIView awe_isBackground] + 0, S

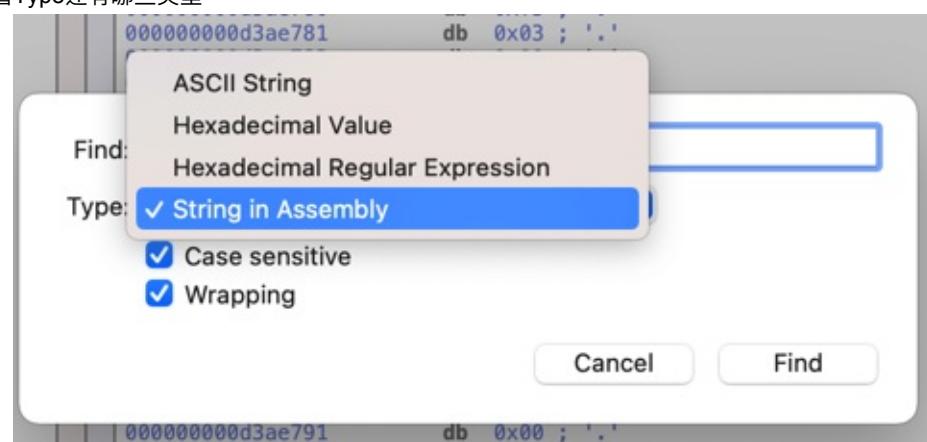
去试试，搜索 Find -> Find



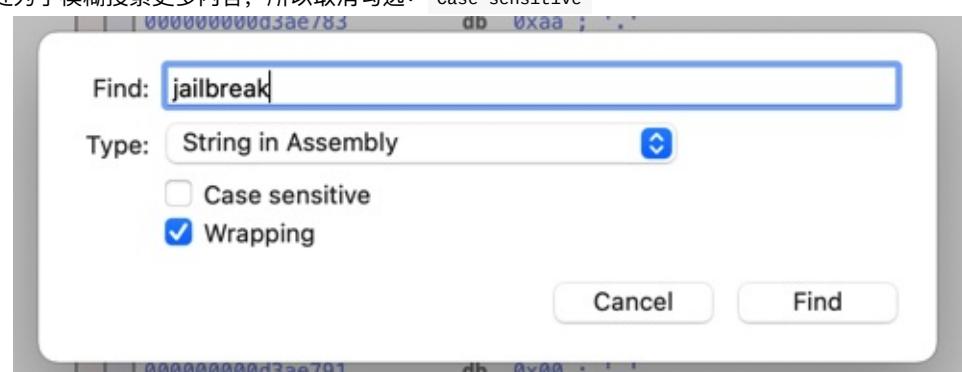
越狱: jailbreak



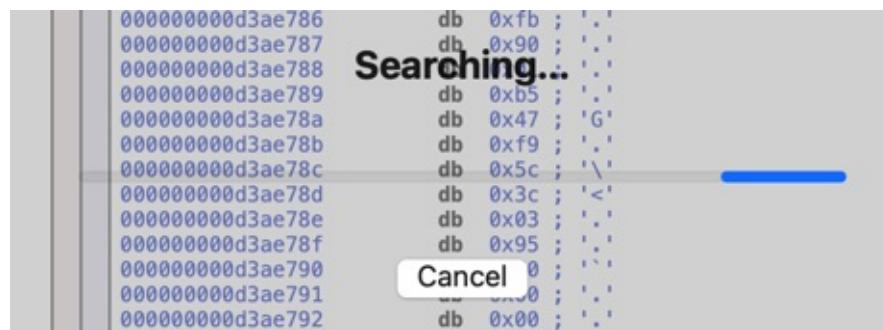
- 看看Type还有哪些类型



- 此处为了模糊搜索更多内容, 所以取消勾选: Case sensitive



开始搜索:

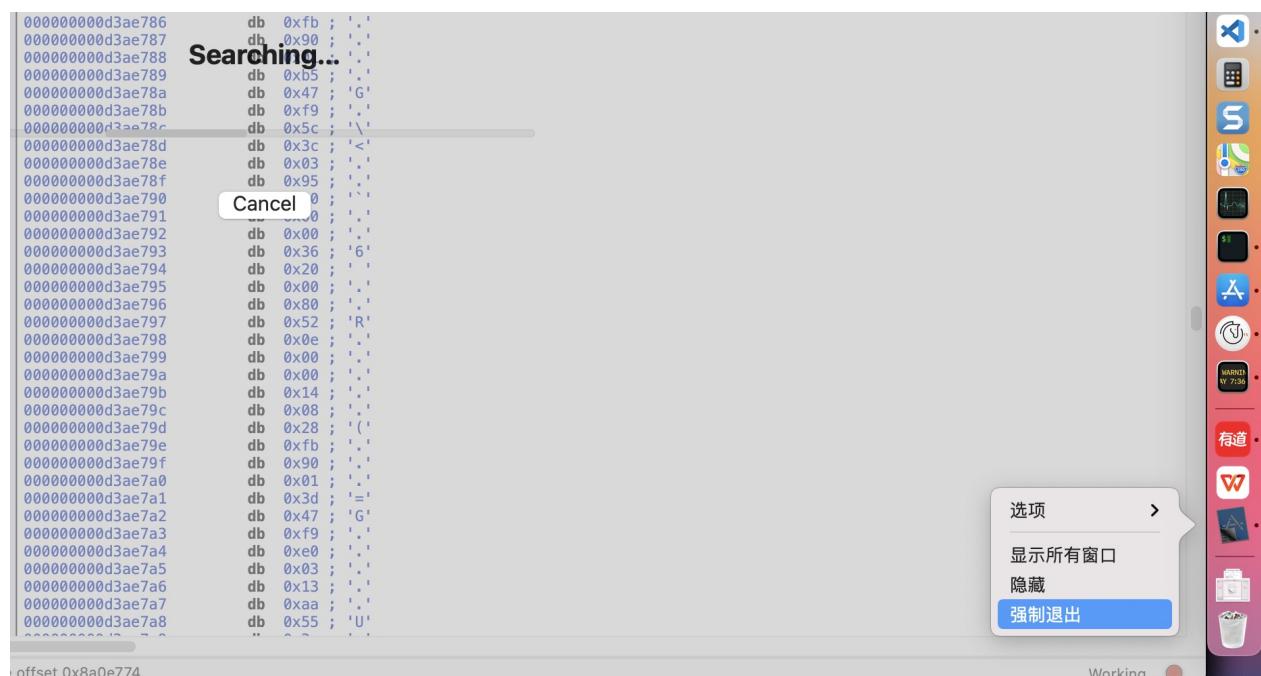


此处搜索了很多分钟，仍没有结束。

后来是，等待了2天多，依旧没结束，所以放弃。

尝试点击Cancel时，已无法点击。

索性强制退出：



->

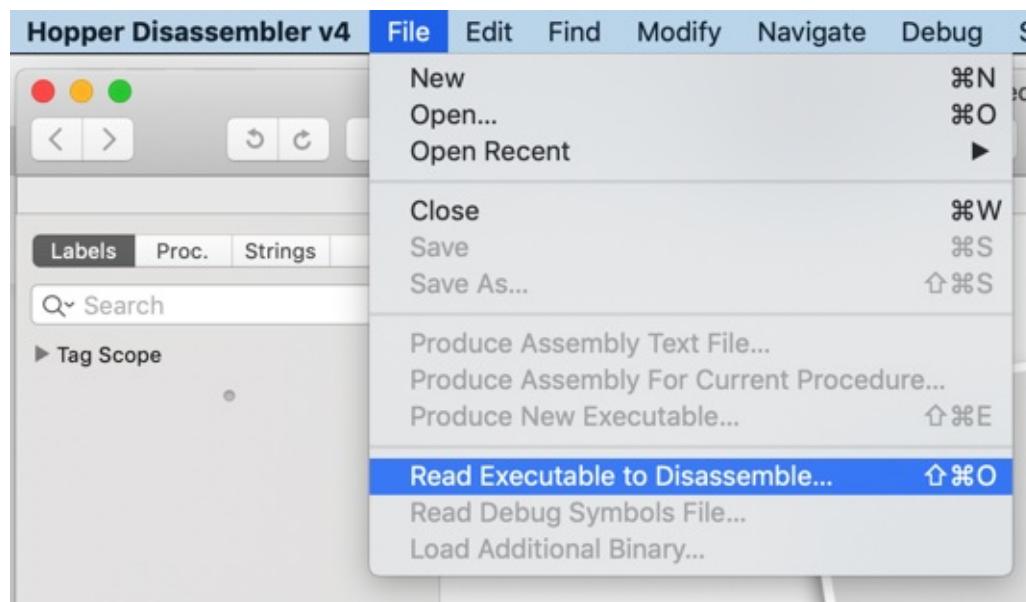
- 目前的结论
 - Hopper对于（包含逻辑和内容很多的）大的二进制，基本上无法正常使用。

Thunder迅雷

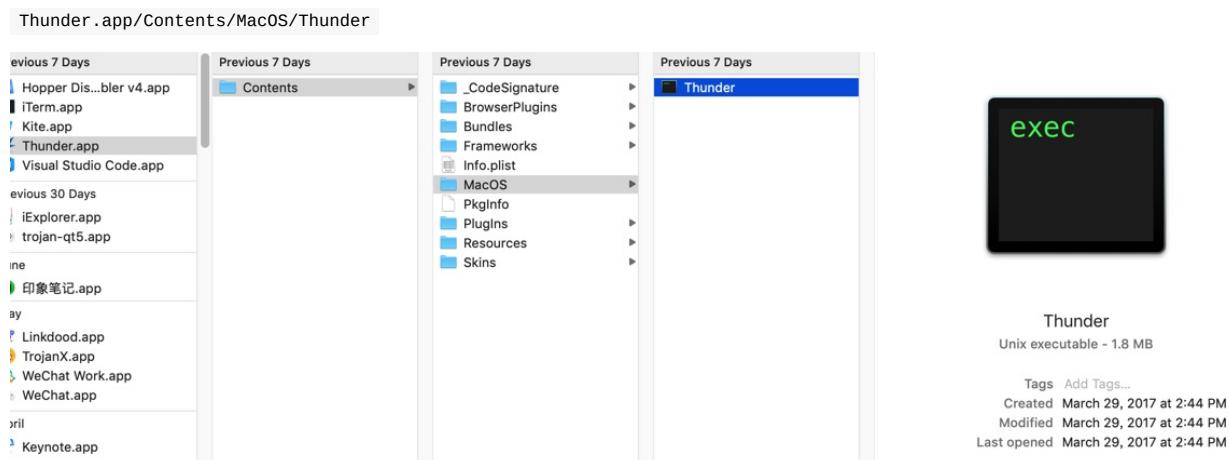
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2023-10-08 14:53:21

Hopper打开Thunder迅雷

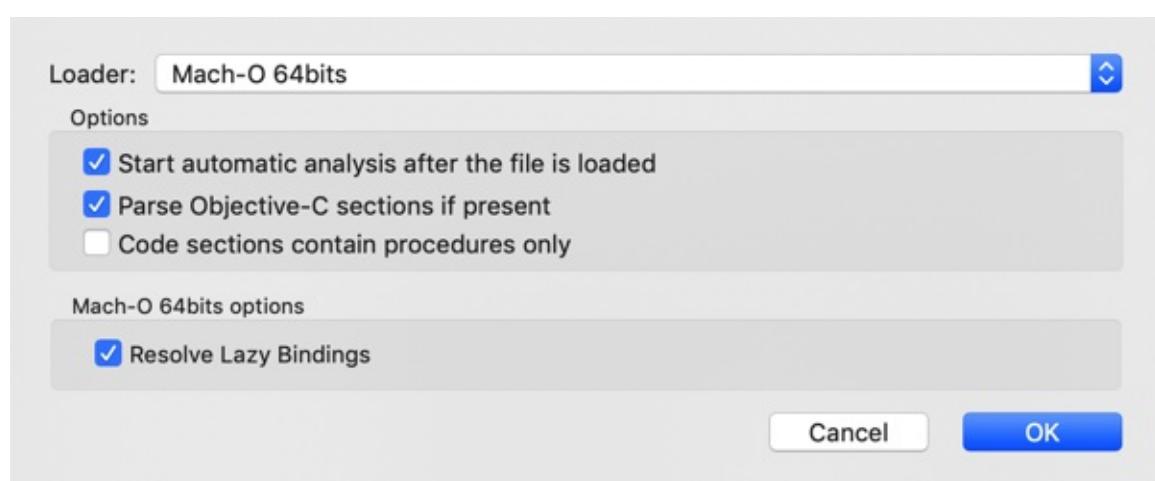
Hopper Disassembler v4 -> File -> Read Executable to Disassemble :



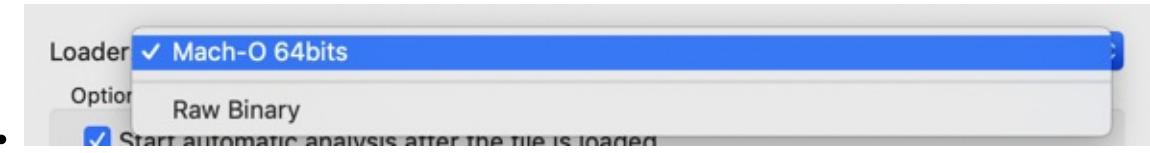
尝试去打开 Mac 的app: 迅雷



弹框显示加载选项:



顺带看看Loader参数的其他可选性:



- Loader可选项

- Mach-O 64bits
- Raw Binary

点击OK，即可加载分析：

```

; ===== BEGINNING OF PROCEDURE =====

_main:
    push    rbp
    mov     rbp, rsp
    pop    rbp
    jmp    imp_main
; endp

+[NSString base64StringFromData:length:]:
    db 0x55 ; 'U'
    db 0x48 ; 'H'
    db 0x89 ; '.'
    db 0xe5 ; ','
    db 0x41 ; 'A'
    db 0x57 ; 'W'
    db 0x41 ; 'A'
    db 0x56 ; 'V'
    db 0x41 ; 'A'
    db 0x55 ; 'U'
    db 0x41 ; 'A'
    db 0x54 ; 'T'
    db 0x53 ; 'S'
    db 0x48 ; 'H'
    db 0x83 ; '.'
    db 0xec ; '.'
    db 0x48 ; 'H'
    db 0x48 ; 'H'
    db 0x48 ; 'H'
    db 0x89 ; '.'
    db 0xd7 ; '.'
    db 0xff ; '.'
    db 0x15 ; '.'
    db 0xb6 ; '.'
    db 0x6f ; 'o'
    db 0x07 ; '.'
    db 0x00 ; '.'
    db 0x49 ; 'I'
    db 0x89 ; '.'
    db 0xc6 ; '.'
    db 0x48 ; 'H'
    db 0x8b ; '.'
    db 0x35 ; '5'

Address 0x100049286, Segment _TEXT, _main + 0, Section __text, file offset 0x49286
Working...

```

开始加载和分析。很快分析完毕，效果是：

打开应用

Address 0x000049286, Segment _TEXT, _main + 0, Section _text, file offset 0x49286

```
0000000100049281 jmp imp_stubs__objc_autoreleaseReturnValue ; endp ; ===== BEGINNING OF PROCEDURE ====== ; _main: 0000000100049286 push rbp 0000000100049287 mov rbp, rsp 000000010004928a pop rbp 000000010004928b jmp imp_stubs__NSApplicationMain ; endp ; ===== BEGINNING OF PROCEDURE ====== ; Variables: ; var_29: -41 ; var_2A: -42 ; var_2B: -43 ; var_2C: -44 ; var_2D: -45 ; var_2E: -46 ; var_2F: -47 ; var_34: -52 ; var_40: -64 ; var_48: -72 ; var_50: -80 ; var_58: -88 ; var_60: -96 ; var_68: -104 ; [NSString base64StringFromData:length::] 0000000100049290 push rbp 0000000100049291 mov rbp, rsp 0000000100049294 push r15 0000000100049296 push r14 0000000100049298 push r13 000000010004929a push r12 000000010004929c push rbx 000000010004929d sub rsp, 0x48 00000001000492a1 mov rdi, rdx 00000001000492a4 call qword [_objc_retain_1000c0260] ; argument "instance" ; _objc_retain 00000001000492aa mov r14, rax 00000001000492ad mov rsi, qword [0x1000f5340] ; @selector(length), a ; Objective C Implementation defined
```

再去把右边和底部也开启显示：

Address 0x000049286, Segment _TEXT, _main + 0, Section _text, file offset 0x49286

```
0000000100049281 jmp imp_stubs__objc_autoreleaseReturnValue ; endp ; ===== BEGINNING OF PROCEDURE ====== ; _main: 0000000100049286 push rbp 0000000100049287 mov rbp, rsp 000000010004928a pop rbp 000000010004928b jmp imp_stubs__NSApplicationMain ; endp ; ===== BEGINNING OF PROCEDURE ====== ; Variables: ; var_29: -41 ; var_2A: -42 ; var_2B: -43 ; var_2C: -44 ; var_2D: -45 ; var_2E: -46 ; var_2F: -47 ; var_34: -52 ; var_40: -64 ; var_48: -72 ; var_50: -80 ; var_58: -88 ; var_60: -96 ; var_68: -104 ; [NSString base64StringFromData:length::] 0000000100049290 push rbp 0000000100049291 mov rbp, rsp 0000000100049294 push r15 0000000100049296 push r14 0000000100049298 push r13 000000010004929a push r12 000000010004929c push rbx 000000010004929d sub rsp, 0x48 00000001000492a1 mov rdi, rdx 00000001000492a4 call qword [_objc_retain_1000c0260] ; argument "instance" ; _objc_retain 00000001000492aa mov r14, rax 00000001000492ad mov rsi, qword [0x1000f5340] ; @selector(length), a ; Objective C Implementation defined
```

> analysis section __objc_classrefs
> analysis section __objc_superrefs
> analysis section __objc_ivar
> analysis section __objc_data
> analysis section __data
> analysis section __common
> analysis section __common
Analysis segment _LINKEDIT
Analysis segment External Symbols
> dataflow analysis of procedures in __TEXT
> dataflow analysis of procedures in __DATA
> dataflow analysis of procedures in _LINKEDIT
> dataflow analysis of procedures in External Symbols
Background analysis ended in 10712ms

>>> Python Command

底部的log是：

```
Hopper is ready  
Mach-O 64bits file loaded
```

```

Starting background analysis
Analysis segment __TEXT
> analysis section __text
> analysis section __stubs
> analysis section __stub_helper
> analysis section __gcc_except_tab
> analysis section __objc_methname
> transform section __objc_methname to C strings
> analysis section __cstring
> transform section __cstring to C strings
> analysis section __const
> analysis section __objc_classname
> transform section __objc_classname to C strings
> analysis section __objc_methtype
> transform section __objc_methtype to C strings
> analysis section __ustring
> analysis section __swift3_typeref
> analysis section __swift3_refistr
> analysis section __swift3_fieldmd
> analysis section __swift3_assocty
> analysis section __swift2_types
> analysis section __swift2_proto
> analysis section __swift3_capture
> analysis section __unwind_info
> analysis section __eh_frame
Analysis segment __DATA
> analysis section __nl_symbol_ptr
> analysis section __got
> analysis section __la_symbol_ptr
> analysis section __const
> analysis section __cfstring
> analysis section __objc_classlist
> analysis section __objc_nlclslist
> analysis section __objc_catlist
> analysis section __objc_protolist
> analysis section __objc_imageinfo
> analysis section __objc_const
> analysis section __objc_selrefs
> analysis section __objc_protorefs
> analysis section __objc_classrefs
> analysis section __objc_superrefs
> analysis section __objc_ivar
> analysis section __objc_data
> analysis section __data
> analysis section __bss
> analysis section __common
Analysis segment __LINKEDIT
Analysis segment External Symbols
  mark procedures
Analysis segment __TEXT
  analysis section __text
  disassemble section __text
  searching additional procedures in section __text
  analysis section __stubs
  disassemble section __stubs
  searching additional procedures in section __stubs
  analysis section __stub_helper
  disassemble section __stub_helper
  searching additional procedures in section __stub_helper
  analysis section __gcc_except_tab
  analysis section __objc_methname
  analysis section __cstring
  analysis section __const
  analysis section __objc_classname
  analysis section __objc_methtype
  analysis section __ustring
  analysis section __swift3_typeref
  analysis section __swift3_refistr
  analysis section __swift3_fieldmd
  analysis section __swift3_assocty
  analysis section __swift2_types
  analysis section __swift2_proto
  analysis section __swift3_capture
  analysis section __unwind_info
  analysis section __eh_frame
Analysis segment __DATA
  analysis section __nl_symbol_ptr
  analysis section __got

```

```

> analysis section __la_symbol_ptr
> analysis section __const
> analysis section __cfstring
> analysis section __objc_classlist
> analysis section __objc_nlclslist
> analysis section __objc_catlist
> analysis section __objc_protolist
> analysis section __objc_imageinfo
> analysis section __objc_const
> analysis section __objc_selrefs
> analysis section __objc_protorefs
> analysis section __objc_classrefs
> analysis section __objc_superrefs
> analysis section __objc_ivar
> analysis section __objc_data
> analysis section __data
> analysis section __bss
> analysis section __common
Analysis segment __LINKEDIT
Analysis segment External Symbols
> dataflow analysis of procedures in __TEXT
> dataflow analysis of procedures in __DATA
> dataflow analysis of procedures in __LINKEDIT
> dataflow analysis of procedures in External Symbols
Background analysis ended in 10712ms

```

显示的是加载和分析的过程。

看到有 `_swift3_typeref`，或许表示此处Mac版Thunder是用Swift写的？

继续看看其他的：

- File Information

◦

- Calling Convention 除了 System v 外还有 Microsoft x64

中间部分解析出的内容：

```

; ████████████████████ B E G I N N I N G   O F   P R O C E D U R E ████████████████████

; Variables
; var_29  41
; var_2A  42
; var_2B  43
; var_2C  44
; var_2D  45
; var_2E  46
; var_2F  47
; var_34  52
; var_40  64
; var_48  72
; var_50  80
; var_58  88
; var_60  96
; var_68  104

```

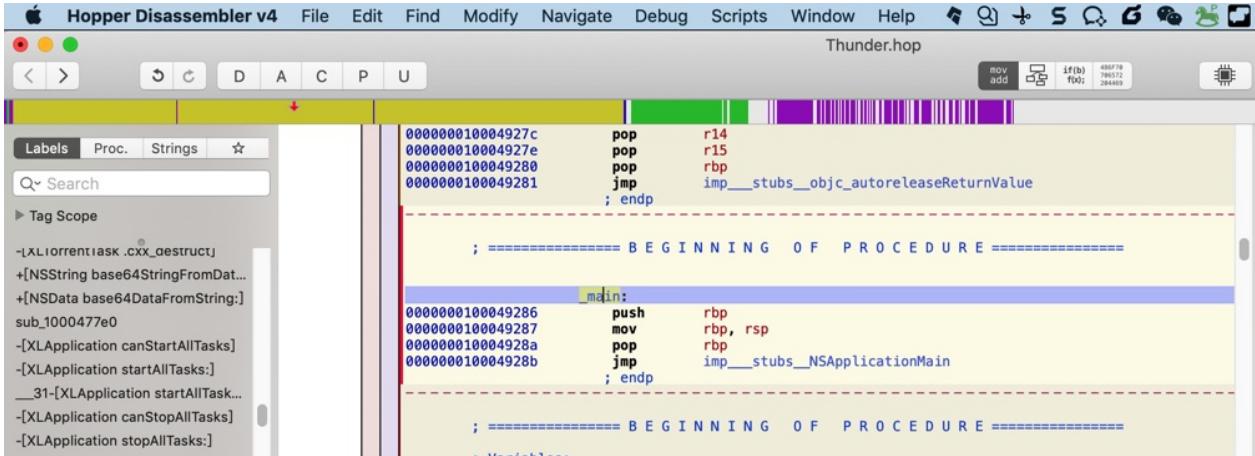
从 Beginning of procedure 感觉是：main函数 入口处

看到了，上面就是 _main :

```

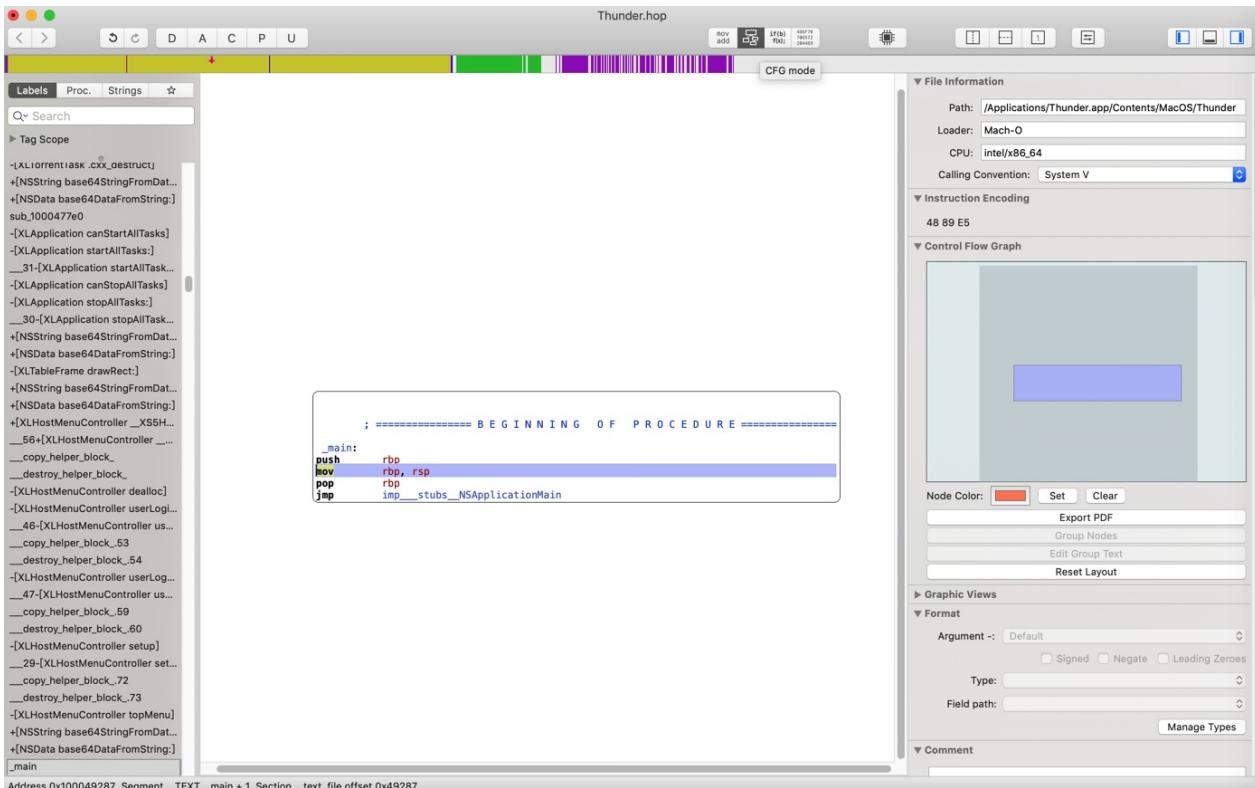
; ===== BEGINNING OF PROCEDURE =====

_main:
0000000100049286    push    rbp
0000000100049287    mov     rbp, rsp
000000010004928a    pop    rbp
000000010004928b    jmp     imp_stubs__NSApplicationMain
;
```



感觉是：Hopper自动帮我们定位到了main入口所在的位置

从 main 切换到 ctf mode , 结果显示的内容，很简单：



没有我希望的：函数调用顺序

不过后来搞懂了：是此处函数本身简单。

换个下面的函数代码段：

Thunder.hop

```

Labels Proc. Strings ☆
Search
Tag Scope

+[NSString base64StringFromData...]
+[NSData base64DataFromString:]
+[LocalTasksMgr sharedMgr]
__26+[LocalTasksMgr sharedMg...
__copy_helper_block_
__destroy_helper_block_
-[LocalTasksMgr autoRunTaskList]
-[LocalTasksMgr _newTasksWithT...
-[LocalTasksMgr init]
-[LocalTasksMgr run]
_task_change_state_callback
_etc_file_name_changed_callback
-[LocalTasksMgr autoRunUncom...
-[LocalTasksMgr _indexWithTaski...
-[LocalTasksMgr _updateIndexes]
-[LocalTasksMgr taskStateChang...
-[LocalTasksMgr _updateTimerEv...
-[LocalTasksMgr moveTasksToRe...
-[LocalTasksMgr recoverTasks:]
-[LocalTasksMgr destroyTasks:de...
-[LocalTasksMgr taskWithTaskid:]
-[LocalTasksMgr setAutoRunForT...
-[LocalTasksMgr checkAutoRunF...
-[LocalTasksMgr removeUncheck...
-[LocalTasksMgr _addUncheckTa...
__43-[LocalTasksMgr _addUnch...
__copy_helper_block_277
__destroy_helper_block_278
__43-[LocalTasksMgr _addUnch...
__copy_helper_block_281
__destroy_helper_block_282
-[LocalTasksMgr _isUncheckTask:]
-[LocalTasksMgr _setupUncheckL...
-[LocalTasksMgr postUncheckTas...
__54-[LocalTasksMgr postUnche...
__copy_helper_block_306

0000000100049280    pop    rbp
0000000100049281    jmp    imp_stubs_objc_autoreleaseReturnValue
; endp

; ====== B E G I N N I N G   O F   P R O C E D U R E ======
_main:
0000000100049286    push   rbp
0000000100049287    mov    rbp, rsp
000000010004928a    pop    rbp
000000010004928b    jmp    imp_stubs_NSApplicationMain
; endn

; ====== B E G I N N I N G   O F   P R O C E D U R E ======
; Variables:
; var_29: -41
; var_2A: -42
; var_2B: -43
; var_2C: -44
; var_2D: -45
; var_2E: -46
; var_2F: -47
; var_34: -52
; var_40: -64
; var_48: -72
; var_50: -88
; var_58: -88
; var_60: -96
; var_68: -104

+[NSString base64StringFromData:length:]: ; Objective C Implement
0000000100049290    push   rbp
0000000100049291    mov    rbp, rsp
0000000100049294    push   r15
0000000100049296    push   r14
0000000100049298    push   r13
000000010004929a    push   r12
000000010004929c    push   rbx
000000010004929d    sub    rsp, 0x48
00000001000492a1    mov    rdi, rdx
00000001000492a4    call   qword [_objc_retain_1000c0260] ; argument "instance"
00000001000492a6    mov    r14, rax
00000001000492ad    mov    rsi, qword [0x1000f5340]
00000001000492b4    mov    rdi, r14
00000001000492b7    call   qword [_objc_msghSend_1000c0248] ; argument "instance"
00000001000492bd    mov    rbx, rax
00000001000492c0    mov    qword [rbp+var_58], rbx
00000001000492c4    test   rbx, rbx
00000001000492c7    je    loc_100049470

00000001000492cd    mov    rdi, qword [objc_cls_ref_NSMutableString] ; argument "instance"
00000001000492d4    mov    rsi, qword [0x1000f5348] ; @selector(stringWi...
00000001000492db    mov    r12, qword [_objc_msghSend_1000c0248]

...

```

Address 0x100049290, Segment _TEXT, +[NSString base64StringFromData:length:] + 264984, Section _text, file offset 0x49290

切换到 CTF mode 后，就可以看到希望看到的：调用关系了：

Thunder.hop

```

Labels Proc. Strings ☆
Q Search
Tag Scope
+NSString base64StringFromData...
+NSData base64DataFromString...
+[LocalTasksMgr sharedMgr]
__26+[LocalTasksMgr sharedMgr...]
_copy_helper_block_
_destroy_helper_block_
-[LocalTasksMgr autoRunTaskList]
-[LocalTasksMgr _newTasksWithT...
-[LocalTasksMgr init]
-[LocalTasksMgr run]
_task_change_state_callback
_etc_file_name_changed_callback
-[LocalTasksMgr autoRunUncom...
-[LocalTasksMgr _indexWithTaski...
-[LocalTasksMgr _updateIndexes]
-[LocalTasksMgr taskStateChang...
-[LocalTasksMgr _updateTimerEv...
-[LocalTasksMgr moveTasksToRe...
-[LocalTasksMgr destroyTasksDe...
-[LocalTasksMgr taskWithTaskid]
-[LocalTasksMgr setAutoRunFor...
-[LocalTasksMgr checkAutoRunF...
-[LocalTasksMgr removeUncheck...
-[LocalTasksMgr _addUncheckTa...
__43-[LocalTasksMgr _addInch...
_copy_helper_block_277
_destroy_helper_block_278
__43-[LocalTasksMgr _addInch...
_copy_helper_block_281
_destroy_helper_block_282
-[LocalTasksMgr _isUncheckTask]
-[LocalTasksMgr _setupUncheckL...
-[LocalTasksMgr postUncheckTa...
__54-[LocalTasksMgr postUnche...
_copy_helper_block_306
Address 0x100049290, Segment __TEXT, +[NSString base64StringFromData:length] + 264984, Section __text, file offset 0x49290

```

File Information
Path: /Applications/Thunder.app/Contents/MacOS/Thunder
Loader: Mach-O
CPU: intel/x86_64
Calling Convention: System V

Control Flow Graph

以及右边有 Control Flow Graph = 控制流程图了

移动后，随时可以看到占整体的比例和位置：

Thunder.hop

```

Labels Proc. Strings ☆
Q Search
Tag Scope
+NSString base64StringFromData...
+NSData base64DataFromString...
+[LocalTasksMgr sharedMgr]
__26+[LocalTasksMgr sharedMgr...]
_copy_helper_block_
_destroy_helper_block_
-[LocalTasksMgr autoRunTaskList]
-[LocalTasksMgr _newTasksWithT...
-[LocalTasksMgr init]
-[LocalTasksMgr run]
_task_change_state_callback
_etc_file_name_changed_callback
-[LocalTasksMgr autoRunUncom...
-[LocalTasksMgr _indexWithTaski...
-[LocalTasksMgr _updateIndexes]
-[LocalTasksMgr taskStateChang...
-[LocalTasksMgr _updateTimerEv...
-[LocalTasksMgr moveTasksToRe...
-[LocalTasksMgr destroyTasksDe...
-[LocalTasksMgr taskWithTaskid]
-[LocalTasksMgr setAutoRunFor...
-[LocalTasksMgr checkAutoRunF...
-[LocalTasksMgr removeUncheck...
-[LocalTasksMgr _addUncheckTa...
__43-[LocalTasksMgr _addInch...
_copy_helper_block_277
_destroy_helper_block_278
__43-[LocalTasksMgr _addInch...
_copy_helper_block_281
_destroy_helper_block_282
-[LocalTasksMgr _isUncheckTask]
-[LocalTasksMgr _setupUncheckL...
-[LocalTasksMgr postUncheckTa...
__54-[LocalTasksMgr postUnche...
_copy_helper_block_306
Address 0x100049290, Segment __TEXT, +[NSString base64StringFromData:length] + 264984, Section __text, file offset 0x49290

```

File Information
Path: /Applications/Thunder.app/Contents/MacOS/Thunder
Loader: Mach-O
CPU: intel/x86_64
Calling Convention: System V

Control Flow Graph

重点再去看看，找找，左边的：

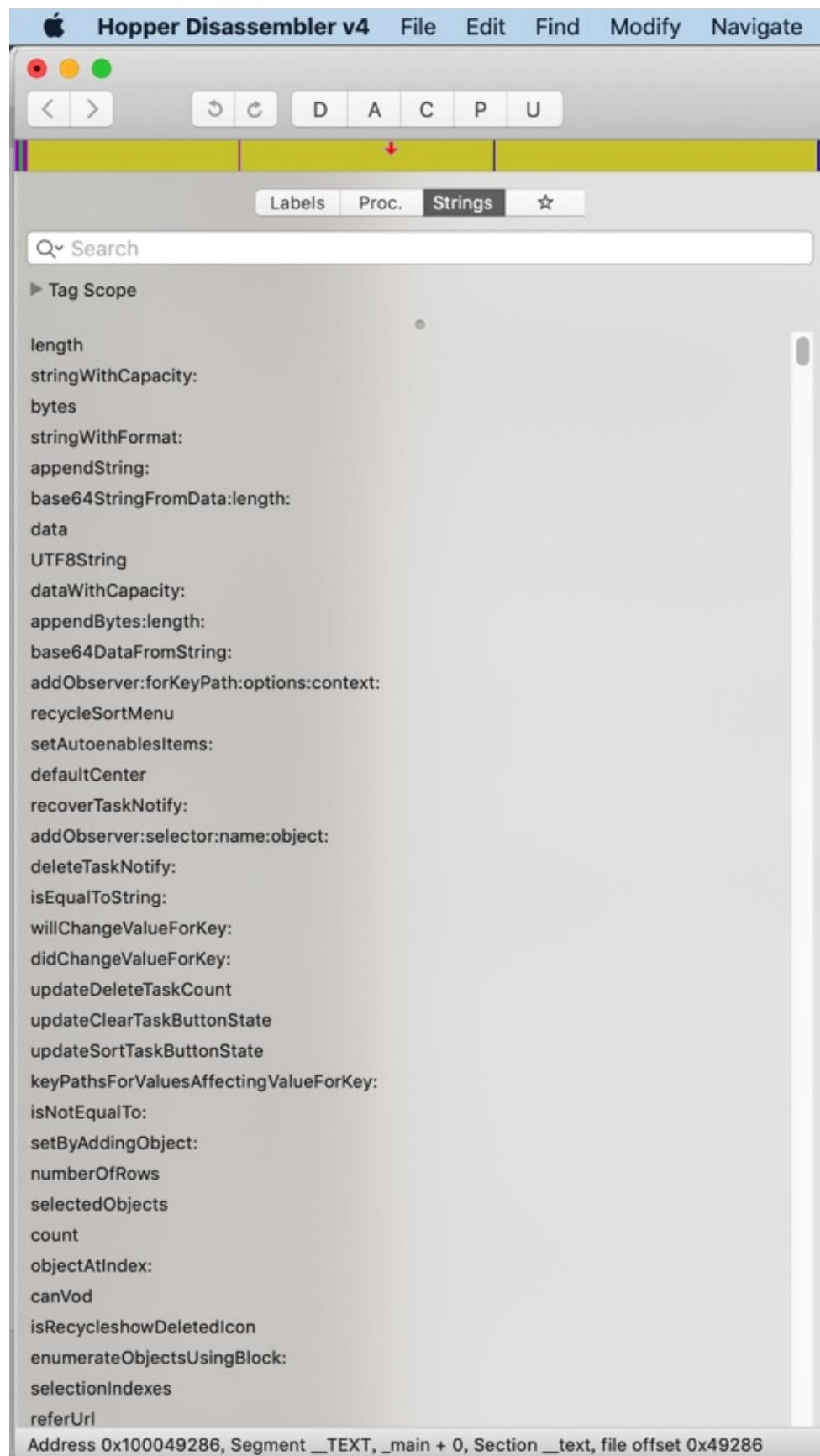
- 函数列表
- Labels



o Proc

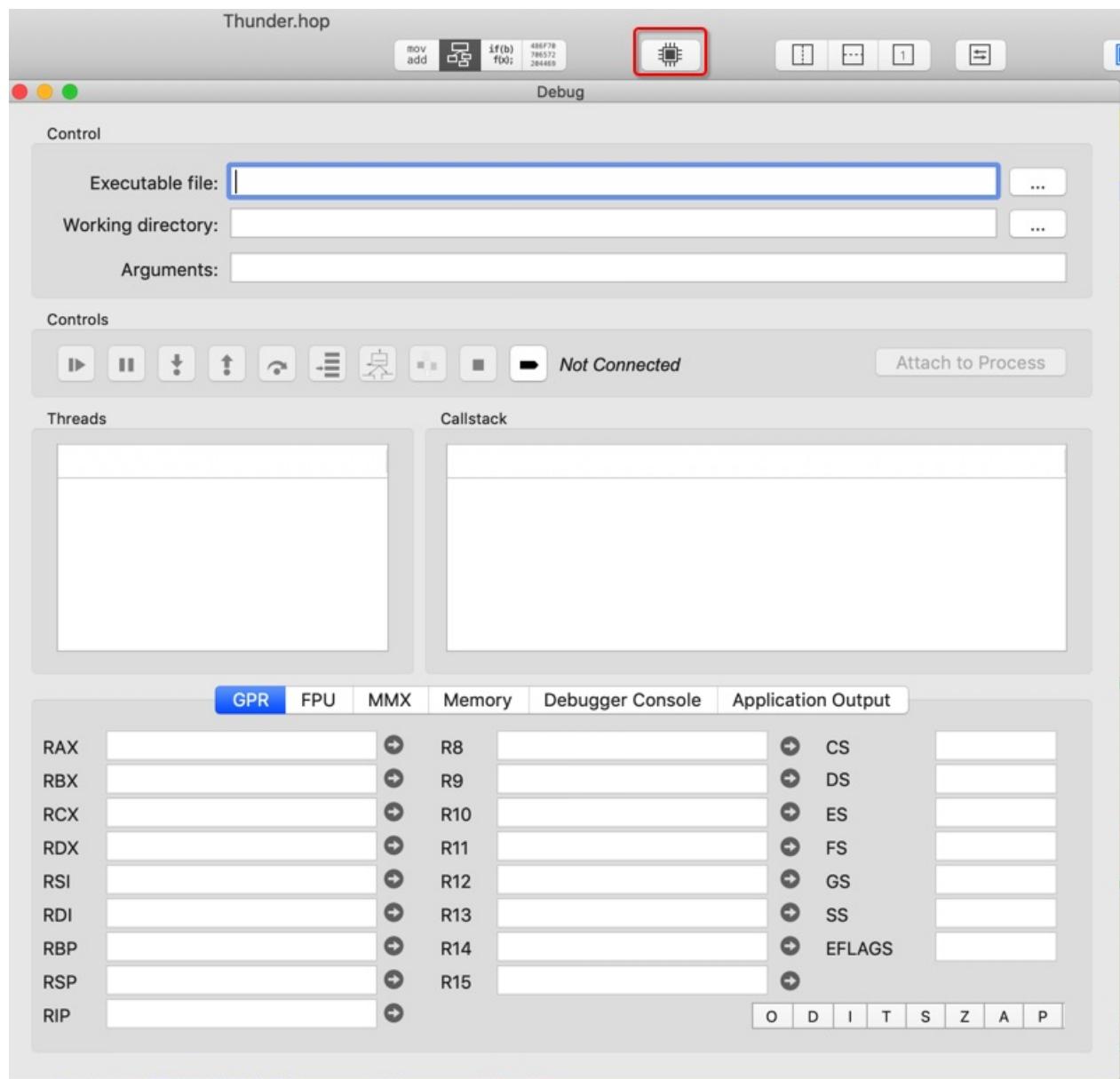


- o Strings



继续看看其他的：

点击 芯片图标 按钮，弹出 调试窗口：



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: 2023-10-08 14:53:56

Hopper分析Thunder逻辑

对于 Thunder迅雷 的基本界面：



想要去研究其中对应的 精选 、 搜索 、 应用 之类的内容。

去搜：

search

The screenshot shows the Hopper Disassembler interface with a search bar containing "search". The results list contains numerous method names, many of which are highlighted in yellow. Some of the visible method names include:

- [XLRecycleToolbarController clickSearchSubtitleMenu:]
- NSSearchPathForDirectoriesInDomains
- hash_search
- _NSSearchPathForDirectoriesInDomains_ptr
- _hash_search_ptr
- _NSSearchPathForDirectoriesInDomains
- _hash_search
- objc_cls_ref_NSKeyedArchiver
- objc_cls_ref_NSKeyedUnarchiver
- _OBJC_CLASS_\$_NSKeyedArchiver
- _OBJC_CLASS_\$_NSKeyedUnarchiver
- [UserNotificationOnCompleteController onTaskStateChanged:]
- +[XLChangeSkinController shareChangeSkinController]
- __51+[XLChangeSkinController shareChangeSkinController]_block_invoke
- [_TtC7Thunder30BrowserExtensionInstallManager bisChromeInstall]
- Thunder.BrowserExtensionInstallManager.bisChromeInstall.getter : Swift.Bool
- [_TtC7Thunder30BrowserExtensionInstallManager setBisChromeInstall:]
- Thunder.BrowserExtensionInstallManager.bisChromeInstall.setter : Swift.Bool
- [_TtC7Thunder30BrowserExtensionInstallManager bisChromeExtensionInst...]
- Thunder.BrowserExtensionInstallManager.bisChromeExtensionInstall.getter : ...
- [_TtC7Thunder30BrowserExtensionInstallManager setBisChromeExtensi...]
- Thunder.BrowserExtensionInstallManager.bisChromeExtensionInstall.setter : ...
- Thunder.BrowserExtensionInstallManager.checkAndOpenGuideDialog () -> ()
- [_TtC7Thunder30BrowserExtensionInstallManager checkAndOpenGuideDia...]
- Thunder.BrowserExtensionInstallManager.startTimerCheckExtension () -> ()
- [_TtC7Thunder30BrowserExtensionInstallManager startTimerCheckExtensi...]
- Thunder.BrowserExtensionInstallManager.bisChromeInstall.materializeForSe...
- Thunder.BrowserExtensionInstallManager.bisChromeExtensionInstall.mater...
- partial apply forwarder for Thunder.BrowserExtensionInstallManager.(startTi...
- _NSAppearanceNameVibrantLight
- _objc_metaclass_XLRecycleToolbarController_methods
- _objc_class_XLRecycleToolbarController_methods
- direct field offset for Thunder.BrowserExtensionInstallManager.bisChromeln...
- direct field offset for Thunder.BrowserExtensionInstallManager.bisChromeEx...
- _NSAppearanceNameVibrantLight
- [XLRecycleToolbarController observeValueForKeyPath:ofObject:change:co...]

Address 0x100049290, Segment _TEXT, +[NSString base64StringFromData:length]

第一个，看起来就是我们希望的要找的

```
-[XLRecycleToolbarController clickSearchSubtitleMenu]:
```

点击 搜索 子菜单

» 那顶部的5个：

精选 正在下载 已完成 搜索 应用

就应该叫做 主菜单了

看了看代码：

```

; ===== BEGINNING OF PROCEDURE =====
; -[XLRecycleToolbarController clickSearchSubtitleMenu];
push rbp ; Objective C Implementation defined at 0x1000cb108 (instance method), DATA XREF=0x1000cb108
mov rbp, rsp
push r15
push r14
push r12
push rbx
mov rbx, rdi
mov rdi, rdx ; argument "instance" for method _objc_retain
call qword [_objc_retain_1000c0260] ; _objc_retain
mov r14, rax
mov rax, qword [_OBJC_IVAR_$_XLRecycleToolbarController_arrayCtrl]
mov rdi, qword [rbx+rax]; argument "instance" for method _objc_msgSend
mov rsi, qword [0x1000f5410] ; @selector(selectedObjects), argument "selector" for method _objc_msgSend
mov r12, qword [_objc_msgSend_1000c0248]
call _objc_msgSend ; _objc_msgSend
mov rdi, rax ; argument "instance" for method imp_stubs__objc_retainAutoreleasedReturnValue
call imp_stubs__objc_retainAutoreleasedReturnValue
mov r15, rax
mov rsi, qword [0x1000f5418] ; @selector(count), argument "selector" for method _objc_msgSend
mov rdi, r15 ; argument "instance" for method _objc_msgSend
call _objc_msgSend ; _objc_msgSend
test rax, rax
je loc_100003f0b

```

Address 0x100003d3, Segment _TEXT, -[XLRecycleToolbarController clickSearchSubtitleMenu] + 78, Section _text, file offset 0x3ed3

也看不出什么头绪

十秒逆向九秒猜

再去找 精选

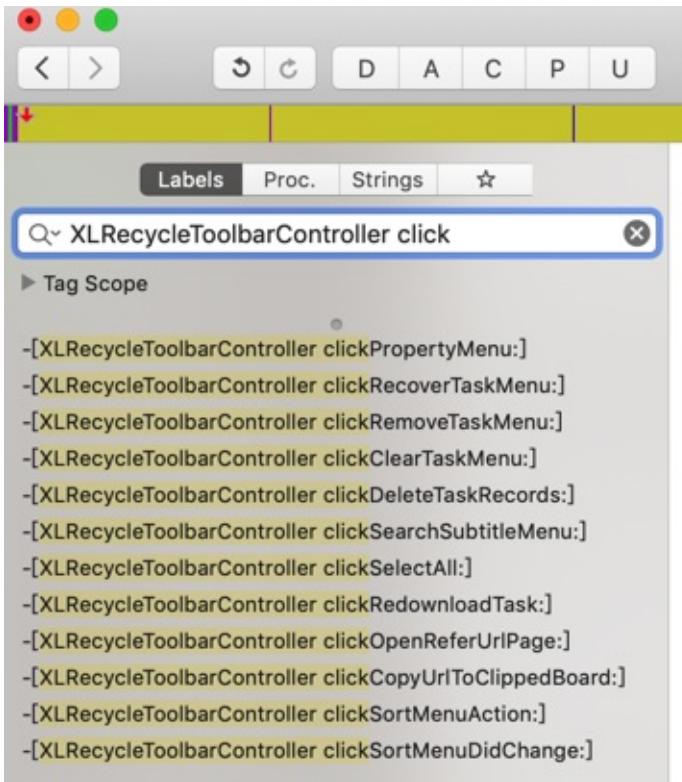
但是不知道英文如何翻译 精选 才能找到

不过想到了，去找类似的

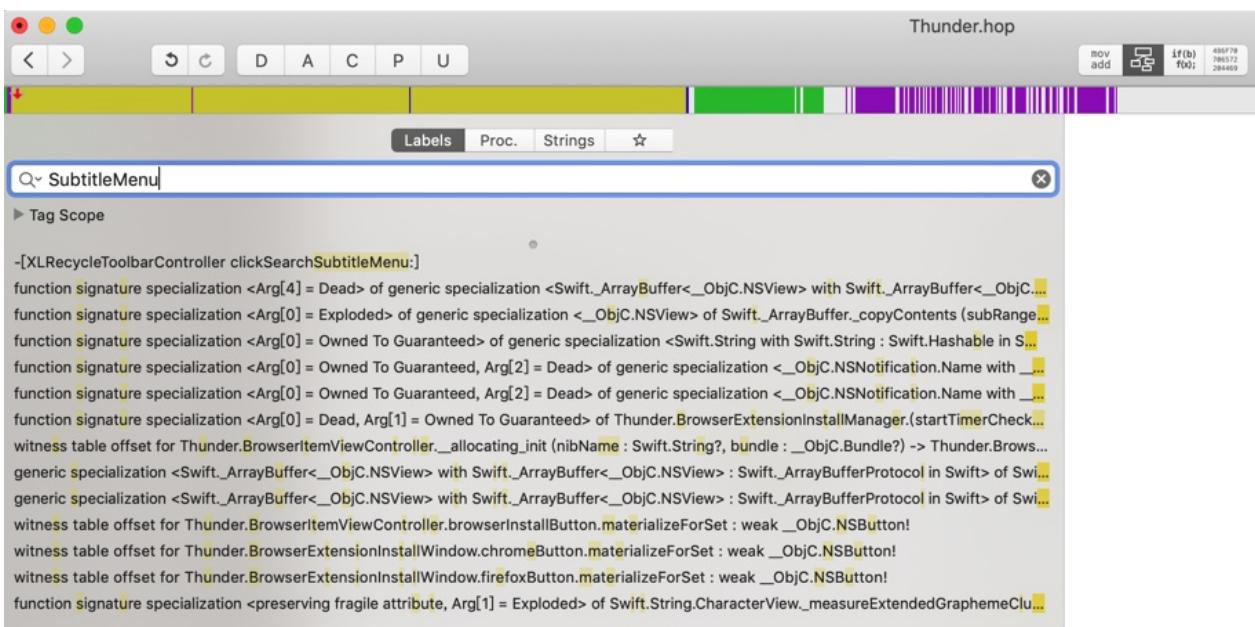
SubtitleMenu

或

XLRecycleToolbarController click



只有其他的一些 但不是我们要的

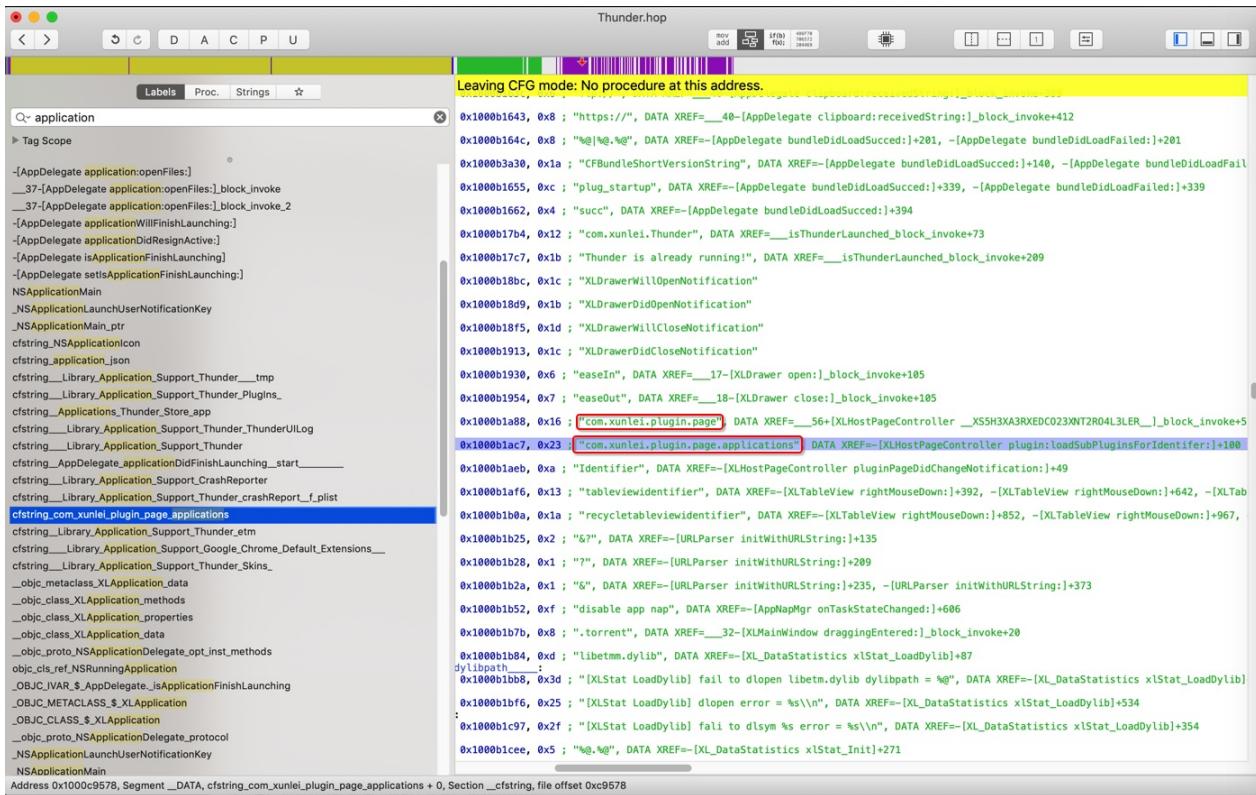


也没有我们要的

搜: jingxuan , 也没有。

再去找找其他的: application

找到:



```

00000001000c9558 dq __CFConstantStringClassReference, 0x7c8, 0x1000b1a88, 0x16 ; "com.xunlei.plugin.pa
ge", DATA XREF __56 [XLHostPageController __X55H3XA3RXEDC023XNT2R04L3LER__]_block_invoke 55
cfstring_com_xunlei_plugin_page_applications
00000001000c9578 dq __CFConstantStringClassReference, 0x7c8, 0x1000b1ac7, 0x23 ; "com.xunlei.plugin.pa
ge.applications", DATA XREF __57 [XLHostPageController plugin loadSubPluginsForIdentifier ] 100
cfstring_Identifier

```

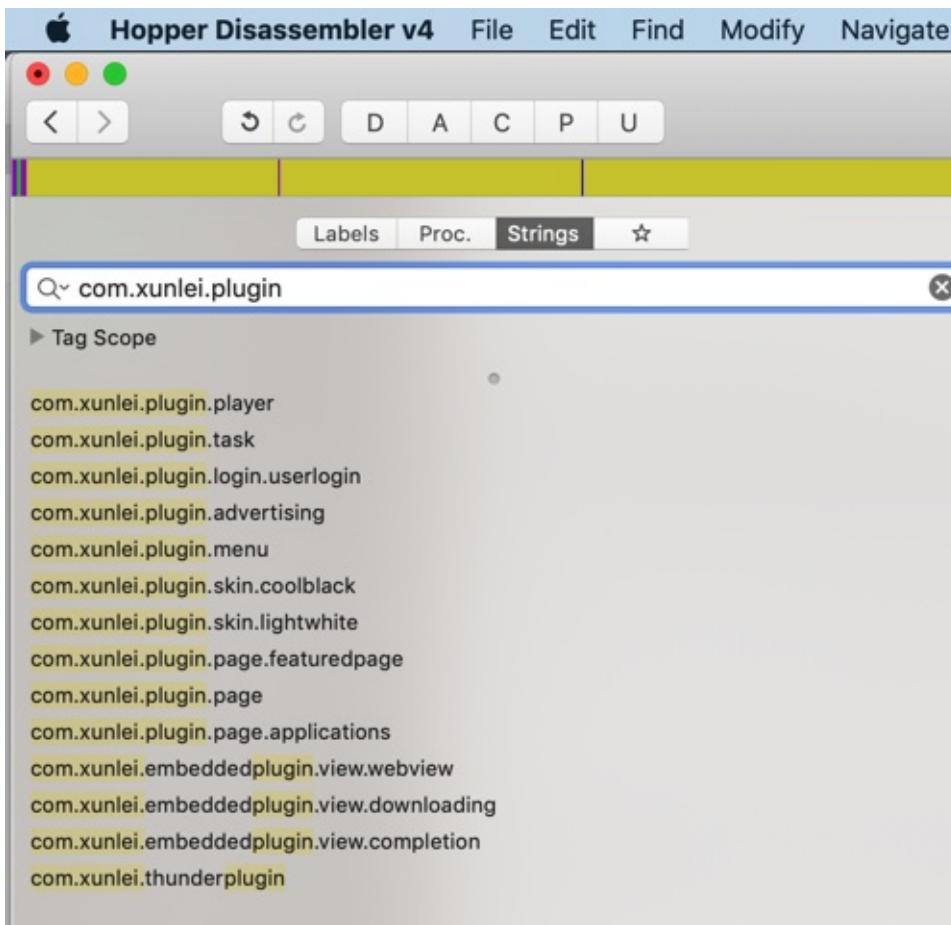
-» 或许还有其他的

com.xunlei.plugin.page.xxx

?

labels 中没找到

不过发现strings中有一些:



不过其中的：

```
00000001000ab6a4      db      "com.xunlei.embeddedplugin.view.webview", 0 ; DATA XREF cfstring_com_xunlei_embedde  
dplugin_view_webview
```

反推，倒是很可能是此处的： 精选



表示用webview 显示 精选内容

那再去搜搜：

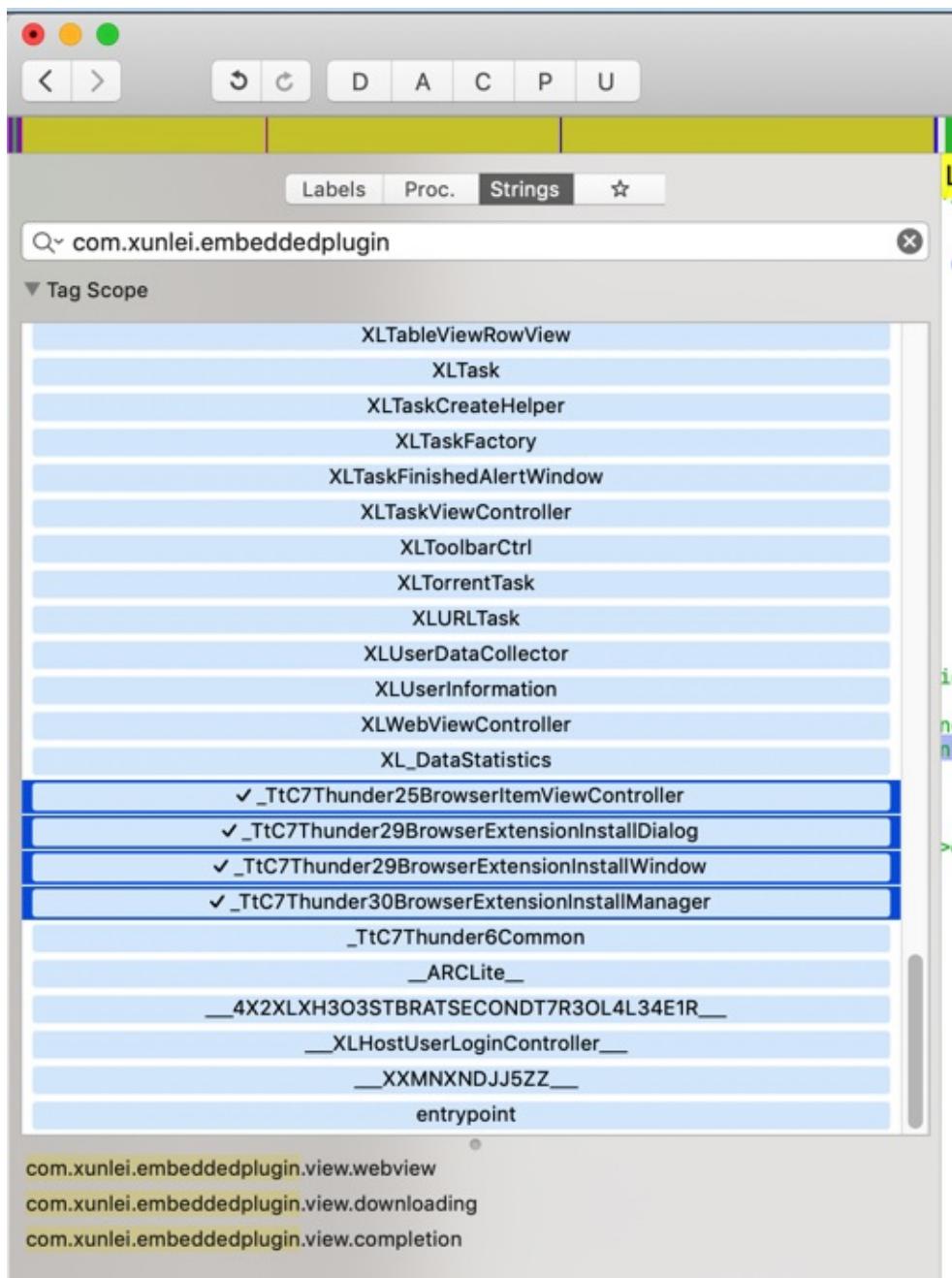
- com.xunlei.embeddedplugin.view
- com.xunlei.embeddedplugin

没有找到其他的

顺带看到一个 Tag Scope



是一个：好像是 内部的类 方法 函数 的列表



另外顺带看看：

- Proc = Procedure = 进程 = 函数

- Strings
 -

◦

去看看app截图：



有下载未完成 待观看

或许能找到这些字符串?

当然如果加密了，是找不到的。

- 搜 下载 : 搜不到。
- 搜 未完成 = `uncomplete` : 搜到一些

◦

继续:

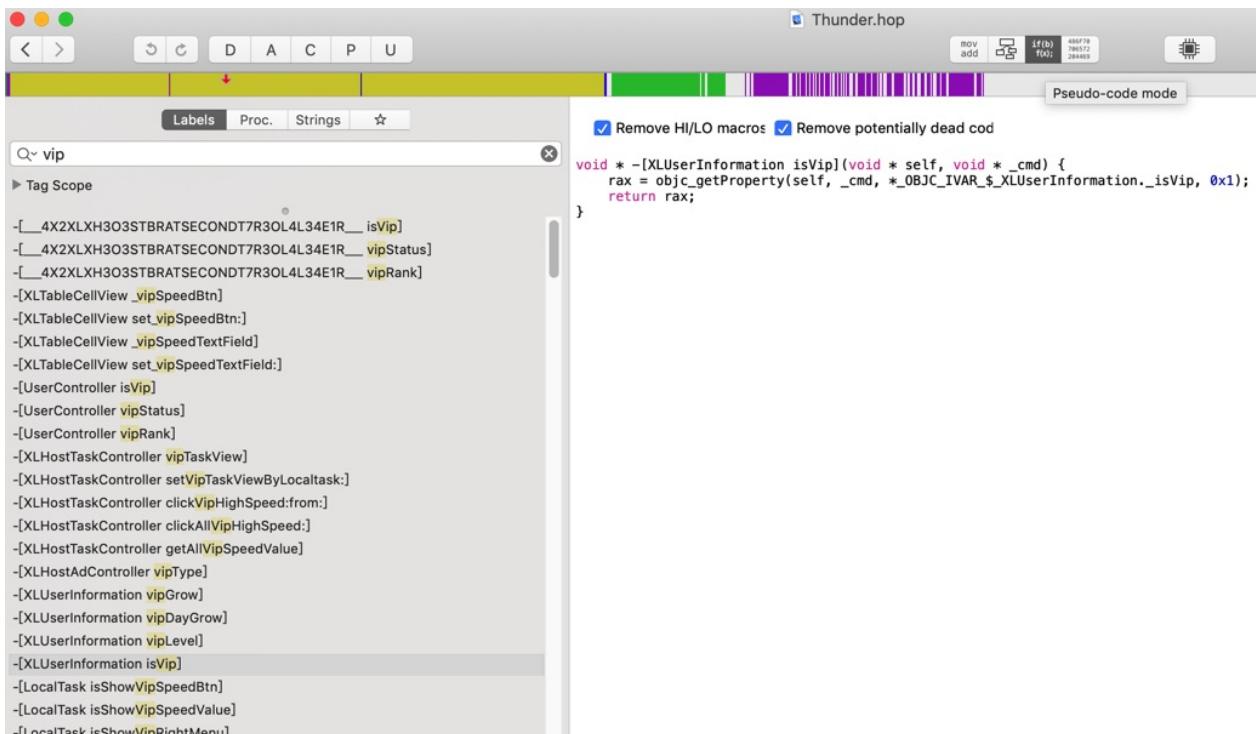
- 搜 `vip` : 还真能搜到些内容

◦

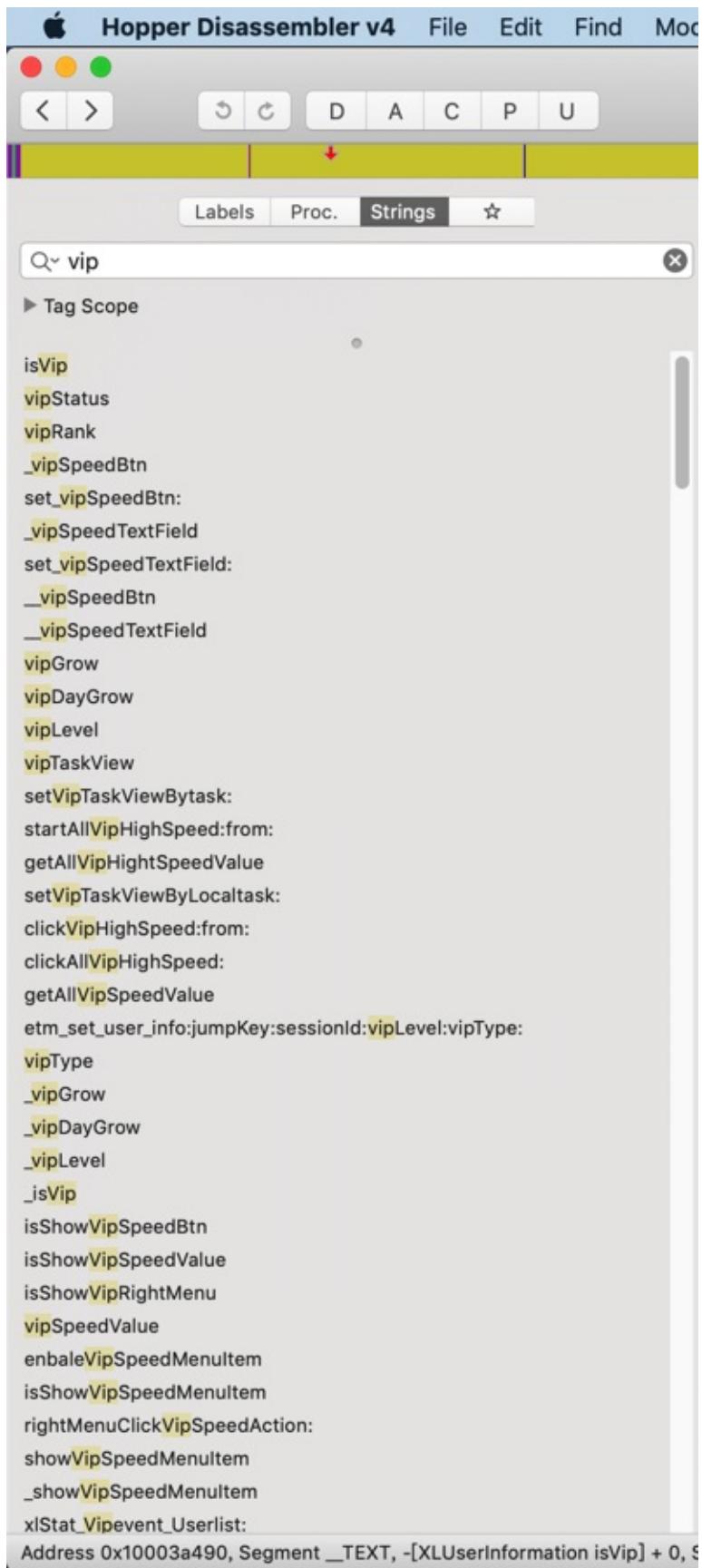
很明显有些是相关内容：

```
isVip  
vipStatus  
vipRank  
  
UserController isVip  
UserController vipStatus  
UserController vipRank
```

去看看： 伪代码 = pseudo code



再去去找找String中是否有我们要的重要内容



可以找到很多 `isVip` 相关的内容

```

000000010009e85a db "zone", 0 ; DATA XREF=0x1000cb7c8, 0x1000f3f98
000000010009e85f db "superclass", 0 ; DATA XREF=0x1000cb7e0, 0x1000f3e18
000000010009e864 db "description", 0 ; DATA XREF=0x1000cb7e8, 0x1000f3e38
000000010009e86f db "delegates", 0 ; DATA XREF=0x1000cb800, 0x1000f4000, 0x1000d5660, 0x
000000010009e87b db "isFlipped", 0 ; DATA XREF=0x1000cb820, 0x1000d5670, 0x
000000010009e88c db "mouseEntered:", 0 ; DATA XREF=0x1000cb828, 0x1000d5688, 0x
000000010009e886 db "mouseExited:", 0 ; DATA XREF=0x1000cb830, 0x1000d5690
000000010009e8a4 db "mouseDelegate", 0 ; DATA XREF=0x1000cb838, 0x1000d5688, 0x1000f5c48
000000010009e8b1 db "setRowDelegates:", 0 ; DATA XREF=0x1000cb840, 0x1000d5688
000000010009e8bd db "rowDelegates", 0 ; DATA XREF=0x1000cb848, 0x1000d5688, 0x1000f5c48
000000010009e8c1 db "rowIndex", 0 ; DATA XREF=0x1000cb850, 0x1000d5688
000000010009e8d4 db "defaultManager:", 0 ; DATA XREF=0x1000cb858, 0x1000d5688, 0x1000f5c48
000000010009e8a9 db "hostController:loadPluginsWithIdentifier:", 0 ; DATA XREF=0x1000f56d8
000000010009e913 db "appVersion", 0 ; DATA XREF=0x1000cc440, 0x1000cc668, 0x1000f56e0
000000010009e91e db "systemVersion", 0 ; DATA XREF=0x1000cc448, 0x1000cc668, 0x1000f56e0
000000010009e92c db "loadBundles", 0 ; DATA XREF=0x1000cc450, 0x1000cc668, 0x1000f56e0
000000010009e93a db "externalConfigPath", 0 ; DATA XREF=0x1000cc458, 0x1000cc668, 0x1000f56e0
000000010009e94e db "status", 0 ; DATA XREF=0x1000cc460, 0x1000cc668, 0x1000f56e0
000000010009e955 db "defaultUserController", 0 ; DATA XREF=0x1000cc468, 0x1000cc668, 0x1000f56e0
000000010009e96b db "nickName", 0 ; DATA XREF=0x1000cc470, 0x1000cc668, 0x1000f56e0
000000010009e974 db "imageUrl", 0 ; DATA XREF=0x1000cc478, 0x1000cc668, 0x1000f56e0
000000010009e97d db "isVip", 0 ; DATA XREF=0x1000cc47e, 0x1000cc668, 0x1000f56e0
000000010009e983 db "userName", 0 ; DATA XREF=0x1000cc480, 0x1000cc668, 0x1000f56e0
000000010009e98c db "loginKey", 0 ; DATA XREF=0x1000cc488, 0x1000cc668, 0x1000f56e0
000000010009e995 db "jumpKey", 0 ; DATA XREF=0x1000cc490, 0x1000cc668, 0x1000f56e0
000000010009e99d db "userNumber:", 0 ; DATA XREF=0x1000cc498, 0x1000cc668, 0x1000f56e0
000000010009e9a8 db "sessionId", 0 ; DATA XREF=0x1000cc4a0, 0x1000cc668, 0x1000f56e0
000000010009e9b2 db "userType", 0 ; DATA XREF=0x1000cc4a8, 0x1000cc668, 0x1000f56e0
000000010009e9bb db "vasType", 0 ; DATA XREF=0x1000cc4b0, 0x1000cc668, 0x1000f56e0
000000010009e9c3 db "vipStatus", 0 ; DATA XREF=0x1000cc4c8, 0x1000cc668, 0x1000f56e0
000000010009e9cd db "isYear", 0 ; DATA XREF=0x1000cc4ce, 0x1000cc668, 0x1000f56e0
000000010009e9d4 db "expiredDate", 0 ; DATA XREF=0x1000cc4d0, 0x1000cc668, 0x1000f56e0
000000010009e9df db "growValue", 0 ; DATA XREF=0x1000cc4d8, 0x1000cc668, 0x1000f56e0
000000010009e9e9 db "dayGrace", 0 ; DATA XREF=0x1000cc4e0, 0x1000cc668, 0x1000f56e0
000000010009e9f6 db "vipRank", 0 ; DATA XREF=0x1000cc4e8, 0x1000cc668, 0x1000f56e0
000000010009e9fe db "userRank", 0 ; DATA XREF=0x1000cc4f0, 0x1000cc668, 0x1000f56e0
000000010009ea07 db "getPeerId", 0 ; DATA XREF=0x1000cc4f8, 0x1000cc668, 0x1000f56e0
000000010009ea11 db "manager", 0 ; DATA XREF=0x1000cc4fa, 0x1000cc668, 0x1000f56e0
000000010009ea19 db "state", 0 ; DATA XREF=0x1000cc4fc, 0x1000cc668, 0x1000f56e0
000000010009ea1f db "dicti", 0 ; DATA XREF=0x1000cc500, 0x1000cc668, 0x1000f5700
000000010009ea2a db "setOb", 0 ; DATA XREF=0x1000cc508, 0x1000cc668, 0x1000f5700
000000010009ea3c db "xlSta", 0 ; DATA XREF=0x1000cc510, 0x1000cc668, 0x1000f5700
000000010009ea50 db "allKe", 0 ; DATA XREF=0x1000cc518, 0x1000cc668, 0x1000f5700
000000010009ea58 db "objec", 0 ; DATA XREF=0x1000cc520, 0x1000cc668, 0x1000f5700
000000010009ea72 db "setSt", 0 ; DATA XREF=0x1000cc528, 0x1000cc668, 0x1000f5700
000000010009ea85 db "fontC", 0 ; DATA XREF=0x1000cc530, 0x1000cc668, 0x1000f5700
000000010009ea96 db "backg", 0 ; DATA XREF=0x1000cc538, 0x1000cc668, 0x1000f5700
000000010009eaad db "borde", 0 ; DATA XREF=0x1000cc540, 0x1000cc668, 0x1000f5700
000000010009eac0 db "image", 0 ; DATA XREF=0x1000cc548, 0x1000cc668, 0x1000f5700
000000010009ead1 db "ident", 0 ; DATA XREF=0x1000cc550, 0x1000cc668, 0x1000f5700
000000010009eadc db "theme", 0 ; DATA XREF=0x1000cc558, 0x1000cc668, 0x1000f5700
000000010009eae9 db " XSHXAR3XEDC023NT2R04L3LER_ ", 0 ; DATA XREF=0x1000cc560, 0x1000cc668, 0x1000f5700
000000010009eae9 db "username", 0 ; DATA XREF=0x1000cc568, 0x1000cc668, 0x1000f5700
000000010009eae9 db "password", 0 ; DATA XREF=0x1000cc570, 0x1000cc668, 0x1000f5700
000000010009eae9 db "newTask:backgroundTask:", 0 ; DATA XREF=0x1000cc578, 0x1000cc668, 0x1000f5700
000000010009eae9 db "mainwindow", 0 ; DATA XREF=0x1000cc580, 0x1000cc668, 0x1000f5700
000000010009eae9 db "showLoginWindow:", 0 ; DATA XREF=0x1000cc588, 0x1000cc668, 0x1000f5700
000000010009eae9 db "selectItemWithIdentifier:", 0 ; DATA XREF=0x1000cc590, 0x1000cc668, 0x1000f5700
000000010009eae9 db "controller:loadWebPageWithURL:", 0 ; DATA XREF=0x1000cc598, 0x1000cc668, 0x1000f5700
000000010009eae9 db " XSHXAR3XEDC023NT2R04L3LER_ ", 0 ; DATA XREF=0x1000cc5a0, 0x1000cc668, 0x1000f5700

```

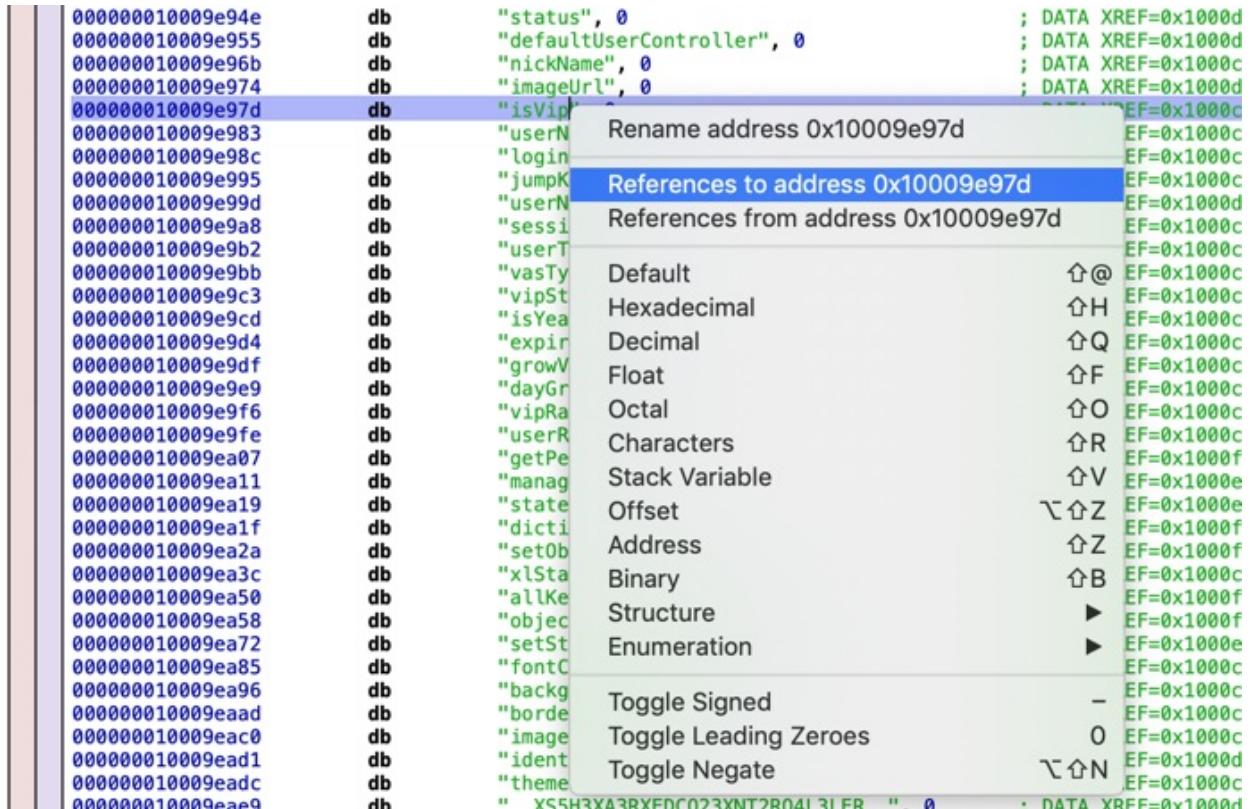
有点看起来是：和user用户相关的各种属性

```

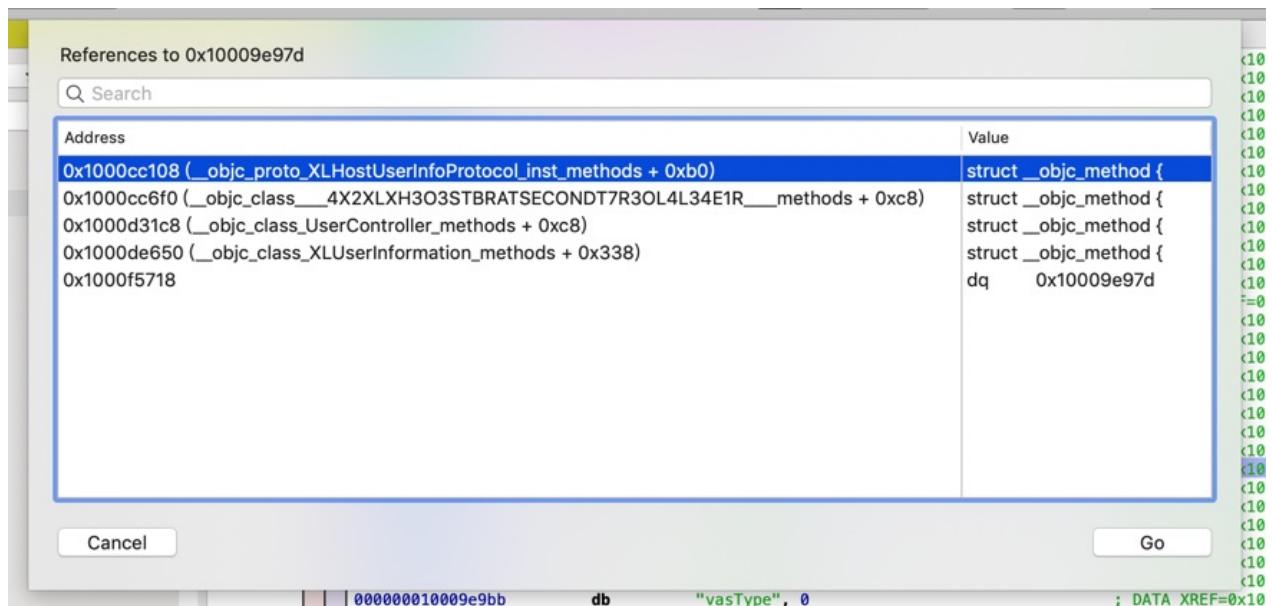
000000010009e96b db "nickName", 0 ; DATA XREF=0x1000cc0d8, 0x1000cc6c0,
0x1000d3120, 0x1000de3b0, 0x1000f5708
000000010009e974 db "imageUrl", 0 ; DATA XREF=0x1000d3198, 0x1000f5710
000000010009e983 db "userName", 0 ; DATA XREF=0x1000cc120, 0x1000cc708,
0x1000d3138, 0x1000de380, 0x1000f5720

```

继续找被调用的地方：



找到了几处：



如此，根据需求，继续深入研究，即可慢慢分析出自己要找的逻辑。

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：2023-10-08 15:08:47

Hopper心得

Hopper导出伪代码

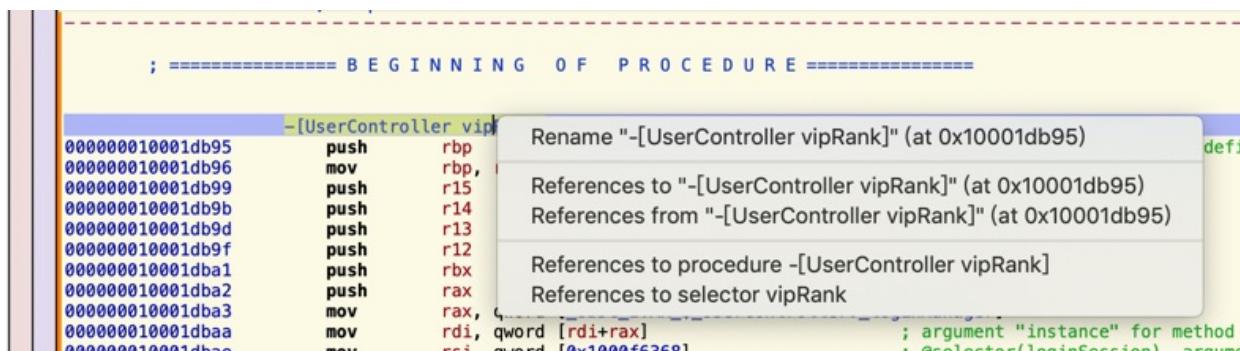
poboke/Class-Decompile: Class Decompile is a python script for Hopper Disassembler. This script can export pseudo code of the classes. ([github.com](https://github.com/poboke/Class-Decompile))

据说可以导出Hopper的全部伪代码。有空去试试。

没有右键复制，直接快捷键复制

点击了函数，竟然没有直接双击选中并复制的功能

且右键也没有复制：



-» 后来发现，其实没有双击和右键复制选择

-» 当光标处于某行，直接Control+C就是复制整行内容

常见问题

卡死

Hopper打开太大的二进制，比如：

- YouTube的 Module_Framework
- Aweme的 AwemeCore

经常直接卡死，无法正常打开

-» 导致无法正常使用。

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：2023-10-08 16:07:45

Hopper vs IDA

- Hopper vs IDA

- 概述

- IDA 算95分， Hopper 算70分

- IDA比Hopper强大很多

- 详解

- 总体对比

- 功能对比

- IDA: 更强大

- 伪代码: 逻辑更清晰

- Hopper: 功能简洁, 基本够用

- 伪代码: 代码逻辑不够清晰

- IDA和Hopper类比

- Hopper: 小工具箱



- IDA: 各种专业工具的工作室



- 分项对比

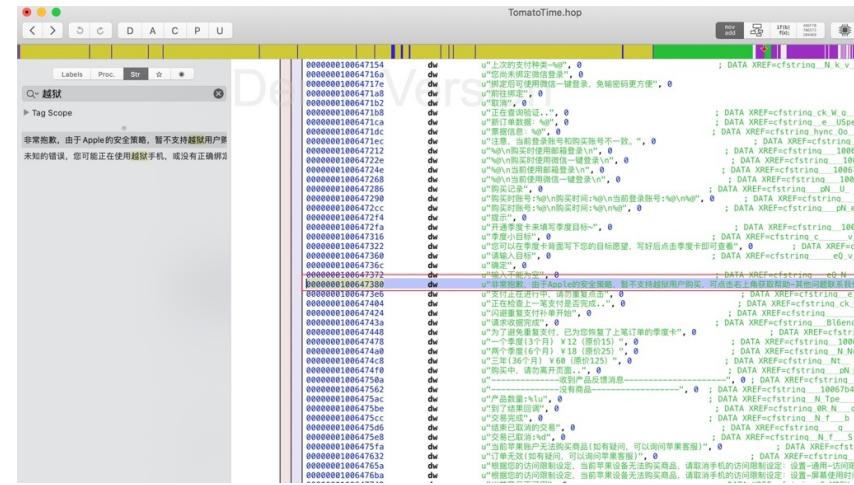
- 平台支持

- Hopper: 更倾向于 Mac

- IDA: 支持多平台: Windows、Linux、Mac

■ 功能支持

- 总体上还是IDA更强大，Hopper相对较弱
- 不过据说部分细节方面，有些Hopper支持更好？
 - 比如
 - 中文字符搜索
 - IDA 7.0+: 不支持
 - Hopper: 支持



The screenshot shows the Hopper Disassembler interface with a search result for the string "越狱". The results list numerous memory addresses (e.g., 0x000000180647154, 0x00000018064716a, etc.) where the string is found. The strings themselves are mostly in Chinese, such as "上次的支付种类-<#>", "您尚未绑定微信登录，免输入密钥更方便！", and "正在查询验证码...". The right side of the screen displays the assembly code for these strings.

■ 价格

- 正版IDA比Hopper贵很多很多
 - IDA Pro: 1000+美元
 - Hopper: ~100美元

Hopper vs IDA：反编译代码对比

某同一段代码的反汇编效果对比：

- IDA

o

- Hopper
 -

◦

◦

再用 BeyondCompare 详细对比：

hooper_loginWithLoginOption.m <-> ida_loginWithLoginOption.m - 文本比较

今天, 上午 11:18:54 3,240 字节 C/C++/CObjC 源代码 Unicode (UTF-8) UNIX

```

int -[LoginAdapter loginWithLoginOption:isForce:extraInfo:completionHandler:cancelationHandler:request:] (struct LoginAdapter *v8 // R81
{
    stack[2048] = arg4;
    r7 = $sp - 0x14 + 0xc;
    sp = sp - 0x74;
    r8 = self;
    r5 = arg3;
    stack[2051] = arg2;
    stack[2053] = [arg4 retain];
    r6 = [arg5 retain];
    r11 = [arg7 retain];
    r10 = [arg8 retain];
    if (r8->_login_service == 0x0) goto loc_2aaee0a;

loc_2aaed6c:
    stack[2054] = r6;
    if (r10 != 0x0)
    {
        [r8 acquirePendingLock];
    }
    if ((r8 accquireLoginLock) & 0xff) == 0x0) goto loc_2aaefab;

loc_2aaea0:
    if (((r5 & 0xff) != 0x0) || ((r8 isValidLogin) & 0xff) == 0x0)) goto loc_2aaee10;

loc_2aaedbe:
    if (r8->_login_service == 0x0) goto loc_2aaee0a;

loc_2aaeae0:
    if (r8->_login_service == 0x0) goto loc_2aaefab;

loc_2aaeae10:
    if (r8->_login_service == 0x0) goto loc_2aaefab;
}

```

今天, 上午 11:19:53 3,698 字节 C/C++/CObjC 源代码 Unicode (UTF-8) UNIX

```

// LoginAdapter - (int)loginWithLoginOption:(int) isForce:(char) extraInfo:(id) completionHandler:(void *)cancelationHandler:(void *)request:(struct LoginAdapter *self) -[LoginAdapter loginWithLoginOption:isForce:extraInfo:completionHandler:cancelationHandler:request:] (struct LoginAdapter *v8 // R81
{
    struct LoginAdapter *v8; // R81
    // some variable

    v8 = self;
    v4 = a4;
    v37 = a3;
    v39 = objc_retain(a5, a2);
    v11 = objc_retain(a6, v10);
    v13 = objc_retain(a7, v12);
    v15 = (void *)objc_retain(a8, v14);
    if (!v8->_login_service)
    {
        v18 = 0;
        v19 = v39;
        goto LABEL_27;
    }
    v40 = v11;
    if (v15)
        objc_msgSend(v8, "acquirePendingLock");
    if ((unsigned int)objc_msgSend(v8, "accquireLoginLock") & 0xFF)
    {
        if (v8) !!(unsigned int)objc_msgSend(v8, "isValidLogin") & 0xFF)
        {
            v36 = v13;
            if (v15)
                objc_msgSend(v8, "pendingLoginRequest:", v15);
            v20 = objc_msgSend(0x80JC_CLASS__LogAdapter, "getInstance");
            v21 = (void *)objc_retainAutoreleasedReturnValue(v20);
            v38 = v8;
            v22 = objc_msgSend(v15, "getApiName");
            v23 = objc_retainAutoreleasedReturnValue(v22);
            v24 = objc_msgSend(v15, "getApiVersion");
            v25 = objc_retainAutoreleasedReturnValue(v24);
            v26 = v25;
            v27 = objc_msgSend(
                &OBJC_CLASS__NSString,
                "stringWithFormat:",
                CFSTR("([LoginAdapter] apiName: %@, apiVersion: %@ pull login module"));
            v28 = objc_retainAutoreleasedReturnValue(v27);
            objc_msgSend(v21, "warn:", v28);
            objc_release(v28);
            objc_release(v26);
            objc_release(v23);
            objc_release(v21);
            v29 = v38->_login_service;
            v40 = &NSConcreteStackBlock;
            v49 = -1040187392;
            v50 = 0;
            v51 = sub_2AAF0B02;
            v52 = &unk_3164640;
            v30 = objc_retain(v38, sub_2AAF0B2);
            v53 = v30;
            v54 = objc_retain(v40, v31);
            v41 = &NSConcreteStackBlock;
            v42 = -1040187392;
            v43 = 0;
            v44 = sub_2AAF1AC;
            v45 = &unk_3164660;
            v12 = v36;
            v46 = objc_retain(v30, &unk_3164660);
            v19 = v39;
            v47 = objc_retain(v36, v32);
            objc_msgSend(v29, "loginWithLoginOption:extraInfo:completionHandler:cancelationHandler:request:");
            objc_release(v47);
            objc_release(v46);
            objc_release(v54);
            objc_release(v53);
            v18 = 2;
            goto LABEL_24;
        }
        objc_msgSend(v8, "releaseLoginLock");
        if (v11)
        {
            v20 = v8;
            v16 = objc_msgSend(v8->_login_service, "currentSession");
            v17 = objc_retainAutoreleasedReturnValue(v16);
            // ...
        }
    }
}

```

8 个差异部分 | 重要差异 | 插入 | 加载时间: 0 秒

hooper_loginWithLoginOption.m <-> ida_loginWithLoginOption.m - 文本比较

今天, 上午 11:18:54 3,240 字节 C/C++/CObjC 源代码 Unicode (UTF-8) UNIX

```

[r8 releaseLoginLock];
r6 = stack[2054];

if (r6 != 0x0)
{
    stack[2052] = r8;
    r5 = [[r8->_login_service currentSession] retain];
    (*r6 + 0xc)(r6, 0x1, r5, *(r6 + 0xc));
}

[r5 release];

}
else {
    stack[2052] = r8;
}

```

今天, 上午 11:19:53 3,698 字节 C/C++/CObjC 源代码 Unicode (UTF-8) UNIX

```

"stringWithFormat:",
CFSTR("([LoginAdapter] apiName: %@, apiVersion: %@ pull login module"),
v23,
v25);
v28 = objc_retainAutoreleasedReturnValue(v27);
objc_msgSend(v21, "warn:", v28);
objc_release(v28);
objc_release(v26);
objc_release(v23);
objc_release(v21);
v29 = v38->_login_service;
v40 = &NSConcreteStackBlock;
v49 = -1040187392;
v50 = 0;
v51 = sub_2AAF0B02;
v52 = &unk_3164640;
v30 = objc_retain(v38, sub_2AAF0B2);
v53 = v30;
v54 = objc_retain(v40, v31);
v41 = &NSConcreteStackBlock;
v42 = -1040187392;
v43 = 0;
v44 = sub_2AAF1AC;
v45 = &unk_3164660;
v12 = v36;
v46 = objc_retain(v30, &unk_3164660);
v19 = v39;
v47 = objc_retain(v36, v32);
objc_msgSend(v29, "loginWithLoginOption:extraInfo:completionHandler:cancelationHandler:request:");
objc_release(v47);
objc_release(v46);
objc_release(v54);
objc_release(v53);
v18 = 2;
goto LABEL_24;
}
objc_msgSend(v8, "releaseLoginLock");
if (v11)
{
    v20 = v8;
    v16 = objc_msgSend(v8->_login_service, "currentSession");
    v17 = objc_retainAutoreleasedReturnValue(v16);
    // ...
}

```

8 个差异部分 | 重要右边独有 | 插入 | 加载时间: 0 秒

The screenshot shows two windows side-by-side, both titled 'hooper_loginWithLoginOption.m <--> ida_loginWithLoginOption.m - 文本比较'. The left window is in Hopper, and the right window is in IDA.

Hopper (Left):

```

    } r5 = 0x3;
    goto loc_2aaaf044;

loc_2aaaf044:
    r4 = stack[2053];
    goto loc_2aaaf046;

loc_2aaaf046:
    if (r10 != 0x0) {
        [stack[2052] releasePendingLock];
    }
    r6 = stack[2054];
    goto loc_2aaaf060;

loc_2aaaf060:
    [r10 release];
    [r11 release];
    [r6 release];
    [r4 release];
    r0 = r5;
    return r0;

loc_2aaee18:
    stack[2050] = r11;
    if (r10 != 0x0) {
        [r8 pendingLoginRequest:r10];
    }
    r6 = [[LogAdapter getInstance] retain];
    stack[2052] = r8;
    r8 = [[r10 getApiName] retain];
    r11 = [[r10 getApiVersion] retain];
    r5 = [[NSString stringWithFormat:@"[LoginAdapter] apiName: %@, apiVersion: %@", pull
    [r6 warn:r5];
    [r5 release];
    [r11 release];
    [r8 release];
    [r6 release];
}
    v16 = objc_msgSend(v8->_login_service, "currentSession");
}

```

IDA (Right):

```

v16 = objc_msgSend(v8->_login_service, "currentSession");
v17 = objc_retainAutoreleasedReturnValue(v16);
(*(void __fastcall **)(int, signed int, int))(v11 + 12))(v11, 1, v17);
objc_release(v17);

else

{
    v38 = v8;
}

v18 = 3;

}

```

Both windows show assembly code with color-coded annotations for ObjC messages and retains. The Hopper window has a red sidebar on the left, while the IDA window has a green sidebar. The bottom status bar indicates 80:51 and 字符串.

结论：

明显能发现，反编译后的伪代码的效果：

IDA更好，Hooper不够好

再具体的说说细节：

- Hopper的ObjC函数调用的写法做了优化，比IDA更易读

- IDA

```

v22 = objc_msgSend(v15, "getApiName");
v23 = objc_retainAutoreleasedReturnValue(v22);

```

- Hopper

```
r8 = [[r10 getApiName] retain];
```

- 但是Hopper的核心代码调用逻辑，没有IDA清楚

- IDA

```
int __cdecl -[LoginAdapter loginWithLoginOption isForce extraInfo completionHandler cancelationHandler request:](struct LoginAdapter *self, SEL a2, int a3, char a4, id a5, id a6, id a7, id a8)
{
    if ( !v8 >_login_service )
    {
        v18 = 0;
        v19 = v39;
        goto LABEL_27;
    }

LABEL_27:
    objc_release(v15);
    objc_release(v13);
    objc_release(v11);
    objc_release(v19);
    return v18;
}
```

- Hopper

```
int -[LoginAdapter loginWithLoginOption isForce extraInfo completionHandler cancelationHandler request:](void
    * self, void _cmd, int arg2, char arg3, void * arg4, void * arg5, void * arg6, void * arg7) {
    ...

loc_2aaaf046
    if (r10 != 0x0) {
        [stack[2052] releasePendingLock];
    }
    r6 = stack[2054];
    goto loc_2aaaf060;

loc_2aaaf060
    [r10 release];
    [r11 release];
    [r6 release];
    [r4 release];
    r0 = r5;
    return r0;

    ...

loc_2aaee0a
    r5 = 0x0;
    r4 = stack[2053];
    goto loc_2aaaf060;
}
```



后记：

自己在给YouTube恢复符号表之后，发现：

IDA的函数调用，也已经自动为iOS的ObjC做了优化，比如：

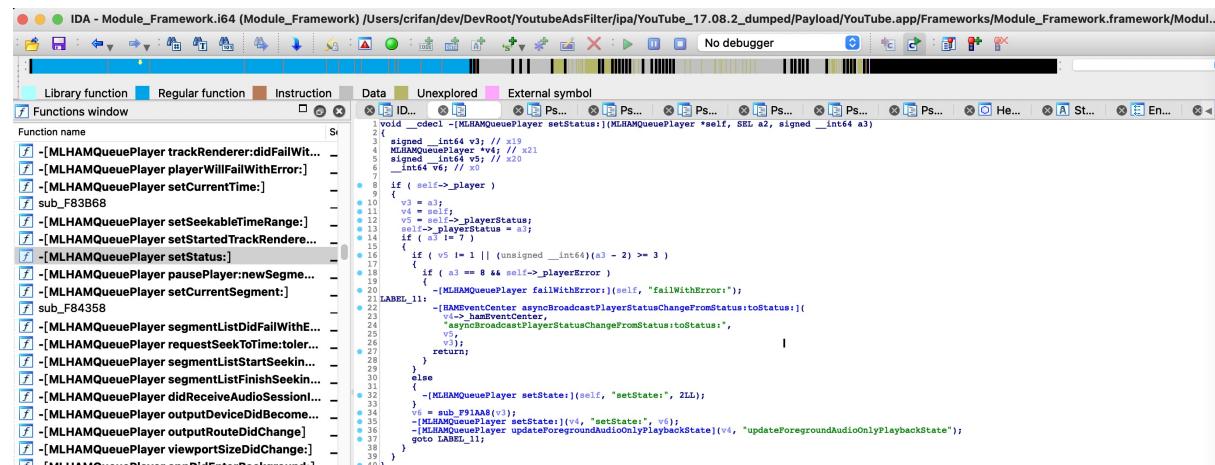
- IDA

```
void __cdecl -[MLHAMQueuePlayer setStatus:](MLHAMQueuePlayer *self, SEL a2, signed __int64 a3)
{
    signed __int64 v3; // x10
    MLHAMQueuePlayer *v4; // x21
    signed __int64 v5; // x20
    __int64 v6; // x0
```

```

if ( self->_player )
{
    v3 = a3;
    v4 = self;
    v5 = self->_playerStatus;
    self->_playerStatus = a3;
    if ( a3 != 7 )
    {
        if ( v5 != 1 || (unsigned __int64)(a3 - 2) >= 3 )
        {
            if ( a3 == 8 && self->_playerError )
            {
                [MLHAMQueuePlayer failWithError:](self, "failWithError:");
            }
        }
    }
    else
    {
        -[MLHAMQueuePlayer setState:](self, "setState:", 2LL);
    }
    v6 = sub_F91AA8(v3);
    [MLHAMQueuePlayer setState:](v4, "setState:", v6);
    [MLHAMQueuePlayer updateForegroundAudioOnlyPlaybackState:](v4, "updateForegroundAudioOnlyPlaybackState");
    goto LABEL_11;
}
}
}

```



其中的：

```

-[MLHAMQueuePlayer failWithError:](self, "failWithError");

-[HAMEventCenter asyncBroadcastPlayerStatusChangeFromStatus:toStatus:](
    v4 _hamEventCenter,
    "asyncBroadcastPlayerStatusChangeFromStatus:toStatus:",
    v5,
    v3);

```

等代码中ObjC函数调用的写法，已经优化为我们希望的效果了：

[objcClass function](para1, para2)

即：

IDA中其实也已支持ObjC函数调用的优化的写法了。

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: 2023-10-08 10:00:15

参考资料

- 【整理】iOS逆向工具：IDA和Hopper对比
- 【已解决】用Hopper去静态分析iOS版抖音
- 【未解决】Mac中用Hopper分析iOS抖音为二进制：找越狱相关内容
- 【已解决】Mac中下载和安装Hopper Disassembler
- 【未解决】Mac中安装和破解Hopper Disassembler v4.app
- 【未解决】Mac中安装Hopper Disassembler破解版
- 【已解决】Mac中用Hopper Disassembler打开迅雷Thunder
- iOS程序逆向Mac下常用工具——Reveal、HopperDisassemble、IDA - 时间已静止 - 博客园
- Hopper Alternatives and Similar Software - AlternativeTo.net
- IDA Pro反汇编工具初识及逆向工程解密实战 - 知乎 (zhihu.com)
- Hopper vs. IDA (wine) on OS X? : ReverseEngineering (reddit.com)
- Ghidra vs Cutter vs Radare2 vs IDA : LiveOverflow (reddit.com)
- A (completely unfair) comparison between radare2, IDA Pro and Hopper : ReverseEngineering (reddit.com)
- RE without IDA? : securityCTF (reddit.com)
- Hopper, an IDA-ish disassembler for OSX (it does Windows binaries!) : ReverseEngineering (reddit.com)
- Hopper VS IDA - compare differences & reviews? (saashub.com)
- tools - Is there any disassembler to rival IDA Pro? - Reverse Engineering Stack Exchange
- ARM binary dissassembly, Hopper works, Ghidra and Radare2 don't for some functions - Reverse Engineering Stack Exchange
- Reverse engineering and malware analysis tools - Infosec Resources (infosecinstitute.com)
- 今天开始学逆向：反汇编的利器 IDA 和 Hopper 的基本使用 - iOS开发 - 开发语言与工具 - 深度开源 (open-open.com)
- 今天开始学逆向：反汇编的利器 IDA 和 Hopper 的基本使用 (daimajiaoliu.com)
- 逆向工程 - Reveal、IDA、Hopper、HTTPS抓包等 - 大河_大河 - 博客园 (cnblogs.com)
- Objective-C语言的逆向(Mac OS) - 知乎 (zhihu.com)
- 利用Hopper Disassembler和IDA Pro修改函数返回值_普通网友的博客-CSDN博客_ida修改函数返回值
- IDA Pro - 如何得到比较清楚的逆向伪代码 - 码上快乐 (codeprj.com)
- hopper逆向的伪代码令人大跌眼镜 - 码上快乐 (codeprj.com)
- iOS逆向之砸壳原理 - 掘金 (juejin.cn)
- iOS逆向学习（一）基础 | BenArvin's blog (benarvintec.com)
- TikTok(抖音国际版)逆向，全球的小姐姐们，我来啦！ - 尚码园 (shangmayuan.com)
- iOS SSL Certificate Pinning: Prevent Bypassing | Guardsquare
- iOS底层原理班（上）/APP逆向实战/加壳脱壳/数据安全/编译原理-学习视频教程-腾讯课堂 (qq.com)
- iOS 崩溃分析 - 掘金 (juejin.cn)
- ios(越狱) 应用脱壳反编译hook教程 (系统ios11.3.1) - 掘金 (juejin.cn)
- 十 iOS逆向- hopper disassembler - 简书 (jianshu.com)
- 飘云阁-PYG|软件安全|破解软件|内购破解|移动安全|chinapyg.com - Powered by Discuz!
- 飘云阁安全论坛的微博_微博
- [原创]破解Hopper Disassembler v3.7.8 for mac的艰难历程-『iOS安全』-看雪安全论坛
- iOS逆向指南：静态分析 | 黑超熊猫zuik's blog
- 最简单的Hopper Disassembler玩转Mac逆向 - 简书
- 逆向破解MacOS App - 简书
- Mac OSX 之自己动手初步学习破解软件入门 - 简书
- Hopper 系列教程之入门 CrackMe 分析 | *tree_fly 's Blog
- 写给 iOS 开发者的 Hopper + lldb 简介 - OneAPM 博客
- 发现iOS SDK的Bug - Hopper使用教程向 | 小猪的博客
- iOS 13.1.3 Runtime Headers
- 动态调试及LLDB技巧集合_TuGeLe的博客-CSDN博客_lldb
- 【求问】Hopper每个方法的blocks和size怎么算的？ - 技能讨论 - 睿论坛 (iosre.com)

- [ios逆向工具Hopper Disassembler的基本使用功能整理\(持续更新\)_小手琴师的博客-CSDN博客](#)
- [iOS逆向之Reveal、Hopper、MachOView等逆向工具的安装使用 - 简书 \(jianshu.com\)](#)
- [3.8 Hopper: 另一款反汇编工具 | iOS 安全 Wiki \(gitbooks.io\)](#)
- [十 iOS逆向- hopper disassembler - 简书 \(jianshu.com\)](#)
- [iOS小技能：逆向工具hopper的使用 - 掘金 \(juejin.cn\)](#)
-

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: 2023-10-08 16:03:19