

# 目录

前言	1.1
恢复符号表概述	1.2
什么是符号表	1.3
恢复符号表前后对比	1.4
如何恢复符号表	1.5
原版restore-symbol	1.5.1
crifan版restore-symbol	1.5.2
常见问题	1.6
Address not found in the image	1.6.1
附录	1.7
参考资料	1.7.1

# iOS逆向分析：恢复符号表

- 最新版本: v0.4
- 更新时间: 20240304

## 简介

介绍关于iOS逆向期间的，恢复符号表相关的各种内容。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### HonKit源码

- [crifan/ios\\_re\\_restore\\_symbol: iOS逆向分析：恢复符号表](#)

### 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逆向分析：恢复符号表 book.crifan.org](#)
- [iOS逆向分析：恢复符号表 crifan.github.io](#)

### 离线下载阅读

- [iOS逆向分析：恢复符号表 PDF](#)
- [iOS逆向分析：恢复符号表 ePub](#)
- [iOS逆向分析：恢复符号表 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：2024-03-04 23:05:48

# 恢复符号表概述

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2024-03-03 17:53:45

# 什么是符号表

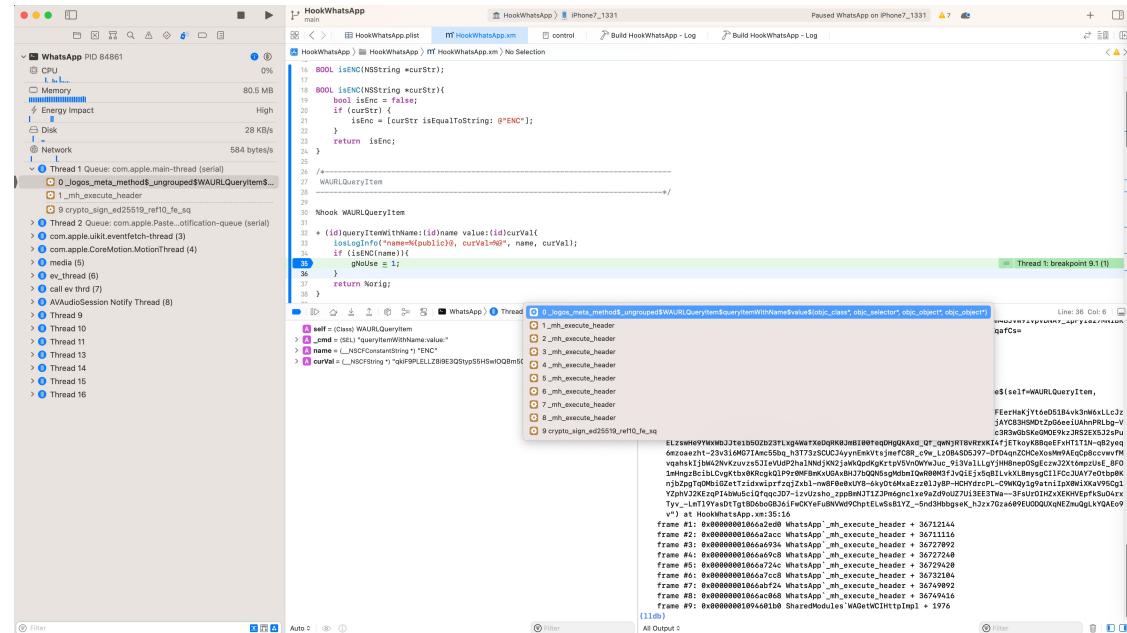
crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2024-03-03 17:53:45

# 恢复符号表前后对比

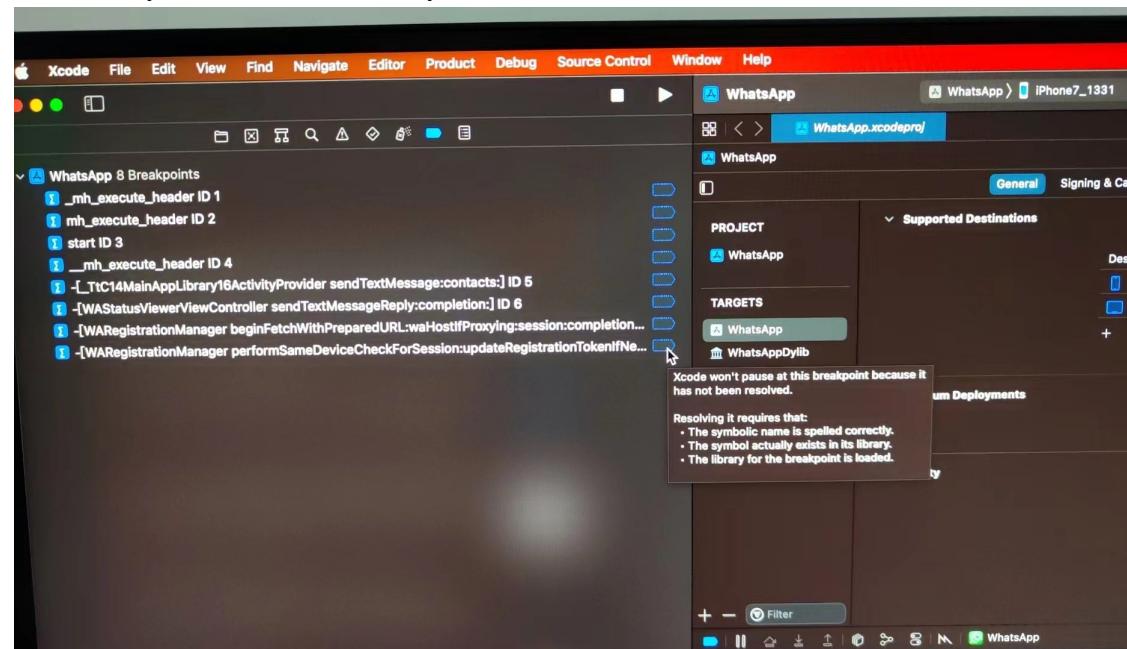
## 恢复符号表之前

- 效果

- Xcode调试iOS程序 -> 查看函数调用堆栈 -> 只能看到无名函数或错误的函数名 -> 无法看到期望的（ObjC等）函数名



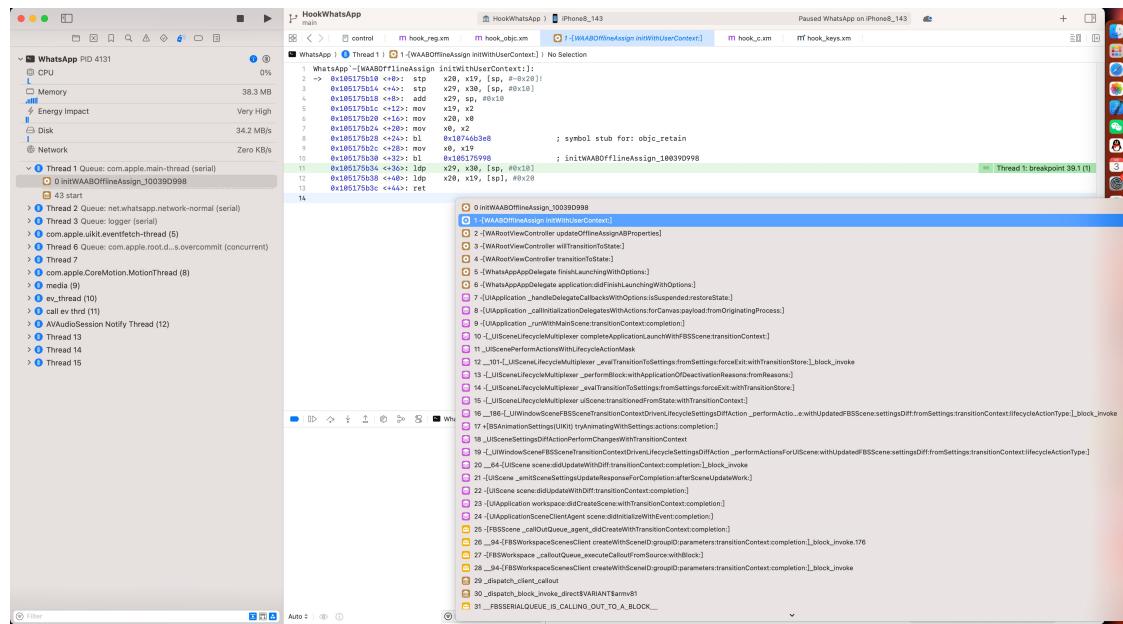
- Xcode给iOS的ObjC函数加断点 -> 通过（ObjC的）函数名加断点，加不上



## 恢复符号表之后

- 效果

- Xcode调试iOS程序 -> 查看函数调用堆栈 -> 就能看到函数名了

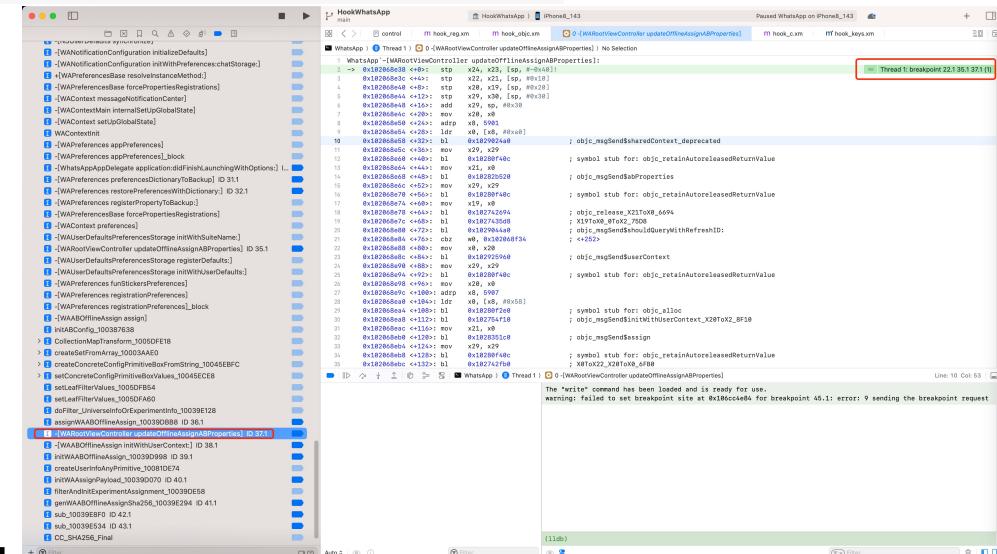


- Xcode给iOS的ObjC函数加断点 -> 断点就能加上了

- 举例

- WhatsApp

- -[WARootViewController updateOfflineAssignABProperties]



## 用工具辅助验证

且可以用其他工具辅助验证：的确加上了函数名=符号表了：

- MachOView

- Dynamic Symbol Table -> Indirect Symbols

- 之前=没有恢复符号表： AwemeCore\_noSymbol 、 AwemeCore\_restoredSymbol

## 恢复符号表前后对比

MachOView fGI's branch

**AwemeCore\_restoredSymbol**

RAW RVA

Search

Offset Data Description Value

Offset	Data	Description	Value
#0	0DDEA900 000000A	Symbol Table Index ,#10 _\$s10Foundation25NSFastEnumerationIteratorVMA	
		Section (_DATA,_got)	
		Indirect Address 0x304000 (\$+0)	
#1	0DDEA904 000000B	Symbol Table Index ,#11 _\$s10Foundation25NSFastEnumerationIteratorVStAMc	
		Section (_DATA,_got)	
		Indirect Address 0x304008 (\$+8)	
#2	0DDEA908 00000012	Symbol Table Index ,#18 _\$s10Foundation3URLVMn	
		Section (_DATA,_got)	
		Indirect Address 0x304010 (\$+16)	
#3	0DDEA90C 00000017	Symbol Table Index ,#23 _\$s10Foundation4DataVN	
		Section (_DATA,_got)	
		Indirect Address 0x304018 (\$+24)	
#4	0DDEA910 0000001F	Symbol Table Index ,#31 _\$s10Foundation4DataVMn	
		Section (_DATA,_got)	
		Indirect Address 0x304020 (\$+32)	
#5	0DDEA914 00000029	Symbol Table Index ,#41 _\$s10Foundation9IndexPathVMA	
		Section (_DATA,_got)	
		Indirect Address 0x304028 (\$+40)	
#6	0DDEA918 0000002A	Symbol Table Index ,#42 _\$s10Foundation9IndexPathVMn	
		Section (_DATA,_got)	
		Indirect Address 0x304030 (\$+48)	
#7	0DDEA91C 0000002B	Symbol Table Index ,#43 _\$s10Foundation9IndexPathVSEAMc	
		Section (_DATA,_got)	
		Indirect Address 0x304038 (\$+56)	
#8	0DDEA920 0000002C	Symbol Table Index ,#44 _\$s10Foundation9IndexPathVSQAMc	
		Section (_DATA,_got)	
		Indirect Address 0x304040 (\$+64)	
#9	0DDEA924 0000002D	Symbol Table Index ,#45 _\$s10Foundation9IndexPathVSeAMc	
		Section (/ DATA cont)	

LinkEdit Parsing ...

### ■ 之后=已恢复符号表: AwemeCore\_restoredSymbol\_HeiTanBc

MachOView fGI's branch

**AwemeCore\_restoredSymbol\_HeiTanBc**

RAW RVA

Search

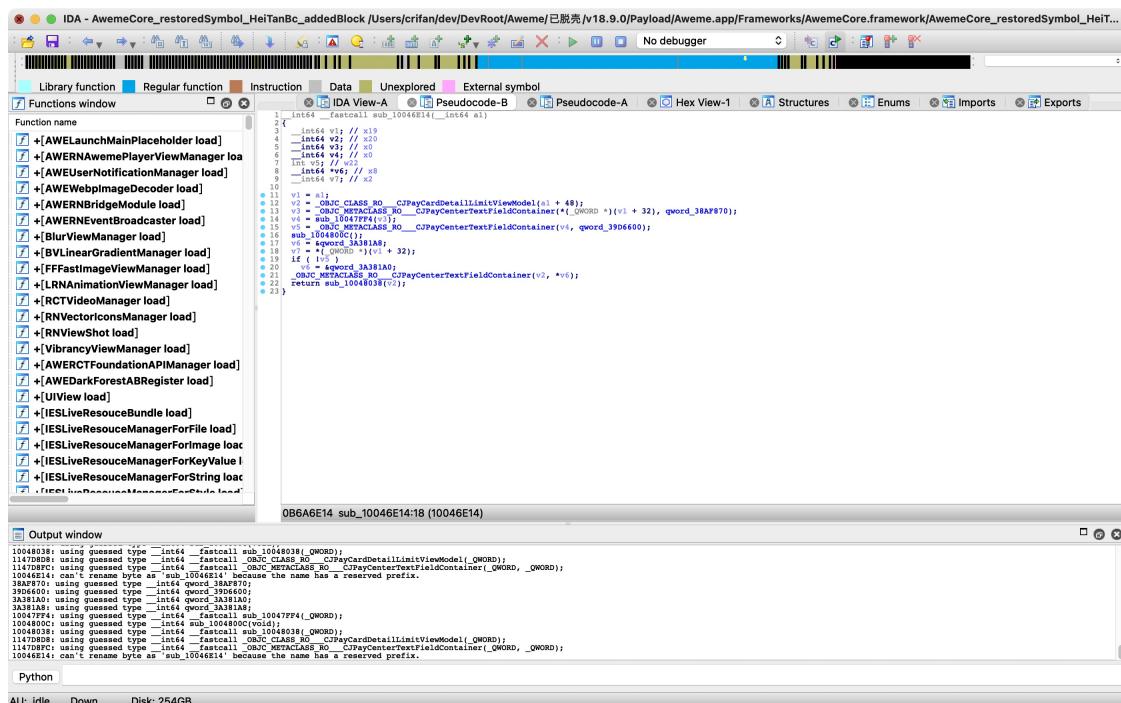
Offset Data Description Value

Offset	Data	Description	Value
#0	0E4ADC60 000000A	Symbol Table Index ,#10 -[BDLDecompressor_AwemeCore setFoo]	
		Section (_DATA,_got)	
		Indirect Address 0x304000 (\$+0)	
#1	0E4ADC64 000000B	Symbol Table Index ,#11 -[BDLDecompressor_AwemeCore foo]	
		Section (_DATA,_got)	
		Indirect Address 0x304008 (\$+8)	
#2	0E4ADC68 00000012	Symbol Table Index ,#18 __OBJC_METACLASS_RO_\$_AWEALogABTestSettings	
		Section (_DATA,_got)	
		Indirect Address 0x304010 (\$+16)	
#3	0E4ADC6C 00000017	Symbol Table Index ,#23 __OBJC_CLASS_RO_\$_AWELaunchSpanManager	
		Section (_DATA,_got)	
		Indirect Address 0x304018 (\$+24)	
#4	0E4ADC70 0000001F	Symbol Table Index ,#31 __OBJC_CLASS_RO_S_AWELaunchHolderTask	
		Section (_DATA,_got)	
		Indirect Address 0x304020 (\$+32)	
#5	0E4ADC74 00000029	Symbol Table Index ,#41 __OBJC_S_PROP_LIST_BaseCell	
		Section (_DATA,_got)	
		Indirect Address 0x304028 (\$+40)	
#6	0E4ADC78 0000002A	Symbol Table Index ,#42 -[BaseCell .cxx_destruct]	
		Section (_DATA,_got)	
		Indirect Address 0x304030 (\$+48)	
#7	0E4ADC7C 0000002B	Symbol Table Index ,#43 -[BaseCell showTagDetailH5]	
		Section (_DATA,_got)	
		Indirect Address 0x304038 (\$+56)	
#8	0E4ADC80 0000002C	Symbol Table Index ,#44 -[BaseCell relationTagLabelTapped]	
		Section (_DATA,_got)	
		Indirect Address 0x304040 (\$+64)	
#9	0E4ADC84 0000002D	Symbol Table Index ,#45 -[BaseCell replyButtonTapped]	

LinkEdit Parsing ...

### ● IDA

#### ○ 之后=已恢复符号表: AwemeCore\_restoredSymbol\_HeiTanBc



- (Xcode中)lldb

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: 2024-03-04 09:59:08

# 如何恢复符号表

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: 2024-03-03 17:53:45

## 原版**restore-symbol**

用**restore-symbol**恢复ObjC符号表

用**restore-symbol**恢复ObjC+block符号表

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: 2024-03-03 17:57:15

# crifan版restore-symbol

用crifan版restore-symbol恢复符号表：

- 核心步骤
  - 用IDA脚本（[exportIDASymbol.py](#)）从IDA中导出符号表
    - 导出之前
      - 优化变量名=符号名称
        - 自动
          - 用crifan的IDA脚本AutoRename，自动优化函数名=符号名
        - 手动
          - 经过（静态或动态）分析代码逻辑后，给函数名等重新命名，优化函数名=符号名称
    - 用[crifan版restore-symbol](#)去恢复符号表（导入符号表）

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：2024-03-04 10:01:39

## 常见问题

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: 2024-03-03 17:53:45

# Address not found in the image

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: 2024-03-03 17:53:45

## 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: 2024-03-03 17:51:12

## 参考资料

- 【未解决】iOS逆向WhatsApp: +[WAURLQueryItem queryItemWithName:value:]
- 【未解决】iOS逆向WhatsApp: SharedModules中的函数加不上如何加上断点且确保能触发
- 【整理】抖音AwemeCore恢复符号表的效果举例
- 【记录】用MachOView对比AwemeCore恢复符号表前后的Symbol变化
- 【记录】用IDA分析加了符号表的抖音AwemeCore二进制
- 【整理】抖音AwemeCore恢复符号表的效果举例
- 【已解决】抖音AwemeCore恢复符号表后导致部分函数显示错乱
- 【已解决】Xcode的lldb调试恢复符号表后的抖音AwemeCore
- 【已解决】restore-symbol给抖音AwemeCore恢复符号表无效
- 

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: 2024-03-04 09:56:39