

目录

前言	1.1
iOS逆向动态调试概览	1.2
反调试和反反调试	1.3
调试代码逻辑	1.4
MonkeyDev	1.4.1
lldb+debugserver	1.4.2
Frida	1.4.3
调试界面元素	1.5
Reveal	1.5.1
Cycrypt	1.5.2
LLDBTools	1.5.3
chisel	1.5.4
FLEX	1.5.5
动态调试心得	1.6
Xcode相关	1.6.1
ObjC	1.6.2
objc_msgSend	1.6.2.1
Runtime	1.6.2.2
po	1.6.3
子教程	1.7
附录	1.8
参考资料	1.8.1

iOS逆向开发：动态调试

- 最新版本: v0.7
- 更新时间: 20221108

简介

介绍iOS逆向中的动态调试，包括动态调试的概览；以及调试代码逻辑方面，包括调试工具的MonkeyDev、lldb+debugserver、Frida等；以及相关子领域，比如反调试和反反调试等；以及查看界面元素的工具，比如Reveal、Cycrypt、LLDBTools、chisel、FLEX等；最后给出一些经验心得。

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_dynamic_debug: iOS逆向开发：动态调试](#)

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

离线下载阅读

- [iOS逆向开发：动态调试 PDF](#)
- [iOS逆向开发：动态调试 ePUB](#)
- [iOS逆向开发：动态调试 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 10:34:25

iOS逆向动态调试概览

iOS逆向，从 是否要运行代码 的角度来说，分：

- 不要运行代码的： [静态分析](#)
- 要运行代码的： [动态调试](#)

此文主要介绍 动态调试 的相关内容：

- 输入=前提：[砸壳出的ipa文件](#)（或已把ipa安装到iOS设备中）
- 主要涉及的内容=领域
 - 调试代码逻辑
 - 常见调试工具
 - 图形界面：[Xcode + MonkeyDev](#)
 - 命令行：[debugserver + lldb](#)
 - [Frida](#)
 - [IDA](#)
 - 涉及到的相关子领域
 - [反调试 和 反反调试](#)
 - [Xcode调试心得](#)
 - [【整理Book】 Xcode开发：调试心得](#)
 - [Xcode内置调试器：LLDB](#)
 - 调试界面元素
 - [Reveal](#)
 - [Cycrypt](#)
 - [\(MonkeyDev的\) LLDBTools](#)
 - [chisel](#)
 - [FLEX](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 11:46:46

反调试和反反调试

TODO:

- 【整理】iOS反越狱相关：反调试 反反调试
 - 【已解决】iOS反调试和反反调试：syscall的ptrace
 - 【未解决】iOS反调试和反反调试：svc 0x80的syscall的ptrace
 - 【已解决】Mac中lldb调试iOS的app抖音报错：Process exited with status 45
-

由于现在多数iOS的app，都做了 反调试 的防护，导致想要能顺利调试iOS的app之前，都要去解决： 反反调试 。

所以此处就分别涉及到：

- 正向的： 反调试
- 逆向的： 反反调试

反调试

反反调试

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-23 15:35:13

调试代码逻辑

TODO:

- 【未解决】如何调试iPhone中iOS的app
-

iOS逆向中的动态调试，其中主要是关于，用各种调试工具去调试代码逻辑。

常用调试工具有：

- MonkeyDev
- lldb+debugserver
- Frida
- IDA
 - 概述：其实IDA更多的是用来[静态分析](#)代码逻辑，偶尔用来 动态调试

以及相关心得：独立子教程

- Xcode调试心得
 - 【整理Book】Xcode开发：调试心得
- LLDB调试心得
 - [Xcode内置调试器：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：

2022-10-27 11:47:54

MonkeyDev

详见独立子教程：

[iOS逆向开发：MonkeyDev调试 \(crifan.org\)](#)

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：

2022-11-03 15:06:55

lldb+debugserver

TODO:

- 【已解决】把增加了权限的debugserver拷贝到越狱iPhone中
- 【已解决】debugserver带日志运行报错：Failed to open log file for writing errno 1 Operation not permitted
- 【已解决】debugserver启动iOS的app抖音报错：Segmentation fault 11
- 【已解决】用debugserver和lldb去调试iOS的app

iOS逆向调试代码逻辑的调试工具之一是：命令行的 lldb + debugserver



- `debugserver`
 - 是什么：一个终端的应用
 - 也是：`xcode` 去调试iOS设备中程序时候的进程名
 - 在哪里：iOS设备中
 - 位置：`/Developer/usr/bin/debugserver`
 - 注：iOS中默认没安装
 - iOS中安装 `debugserver`
 - 在设备连接过一次 `xcode`，并在 `Window -> Devices` 中添加此设备后
 - `debugserver` 才会被 `xcode` 安装到 iOS 的 `/Developer/usr/bin/` 下
 - 作用：作为服务端，接受来自远端的 `gdb` 或 `lldb` 的调试
 - 可以理解为：`lldb` 的 `server`
 - 为何需要
 - iOS中，由于App运行检测到越狱后会直接退出，所以需要通过 `debugserver` 来启动程序
 - 通过 `debugserver` 来启动程序
 - 举例
 - `debugserver -x backboard 0.0.0.0:1234 ./*`
 - `debugserver *:1234 -a "MoneyPlatListedVersion"`

从技术上，应属于：LLDB的远程调试，需要用到：lldb-server

- lldb-server 远程调试
 - 分2个端
 - lldb client
 - 运行在 local system

- 比如 Linux 、 Mac
- lldb server
 - 不同平台
 - Linux 和 Android : lldb-server
 - 不依赖于 lldb
 - 因为：已静态链接包含了 LLDB 的核心功能
 - 对比： lldb 是默认是动态链接 liblldb.so
 - Mac 和 iOS : debugserver
 - 运行在 remote system
 - 实现了remote-gdb的功能
 - 两者通讯
 - 用的是： gdb-remote 协议
 - 一般是在TCP/IP之上运行
- 细节详见：
 - docs/lldb-gdb-remote.txt
- 资料
 - 主页
 - Remote Debugging — The LLDB Debugger
 - <http://lldb.llvm.org/use/remote.html>

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-23 16:16:55

Frida

TODO:

- 【记录】 iOS的iPhone中安装Frida
 - 【未解决】 用Frida动态调试iOS版抖音app
 - 【已解决】 frida启动抖音app报错： Failed to attach need Gadget to attach on jailed iOS
 - 【已解决】 Frida中如何通过frida启动被测app程序iOS版抖音
 - 【未解决】 frida去hook函数_dyld_get_image_name时打印参数为字符串
 - 【未解决】 frida调试抖音app去hook函数：_dyld_get_image_name
 - 【已解决】 frida调试进程直接运行不要每次都输入%resume才运行
 - 【已解决】 用frida启动hook调试iOS抖音app
 - 【未解决】 Mac中frida-trace报错： Failed to spawn unable to find process with name
 - 【未解决】 用Frida的frida-trace去hook函数iOS版抖音
 - 【记录】 frida的frida-ps用法
 - 【记录】 iOS的iPhone中安装Frida
-

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：
2022-11-07 14:14:21

调试界面元素

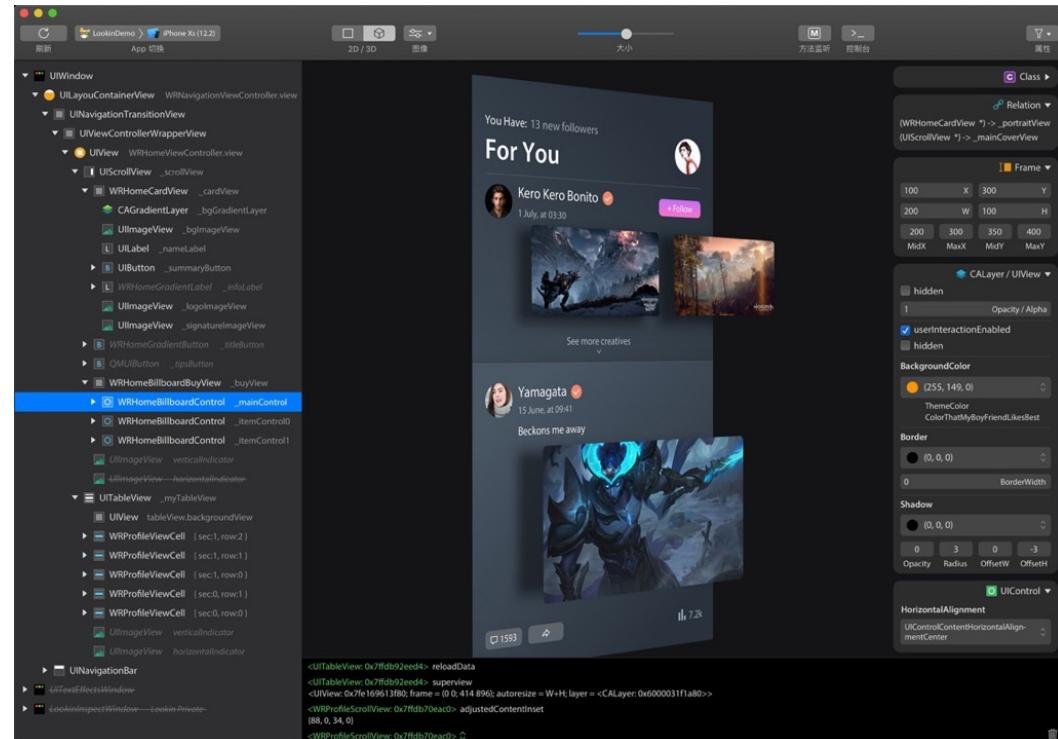
TODO:

- 【整理】页面元素调试结果对比：Reveal、Cycrypt、LLDBTools、chisel
- 【记录】XCode+MonkeyDev动态调试抖音：从点赞关注UI界面入手找底层代码逻辑

iOS逆向的动态调试，也常会，从app的界面入手找对应的按钮等元素，此时就会涉及到：调试界面元素

常用的iOS的app的界面调试工具：

- Reveal
- Cycrypt
- (MonkeyDev的) LLDBTools
- chisel
- FLEX
- 其他
 - LookinLoader
 - <https://github.com/creantan/LookinLoader>



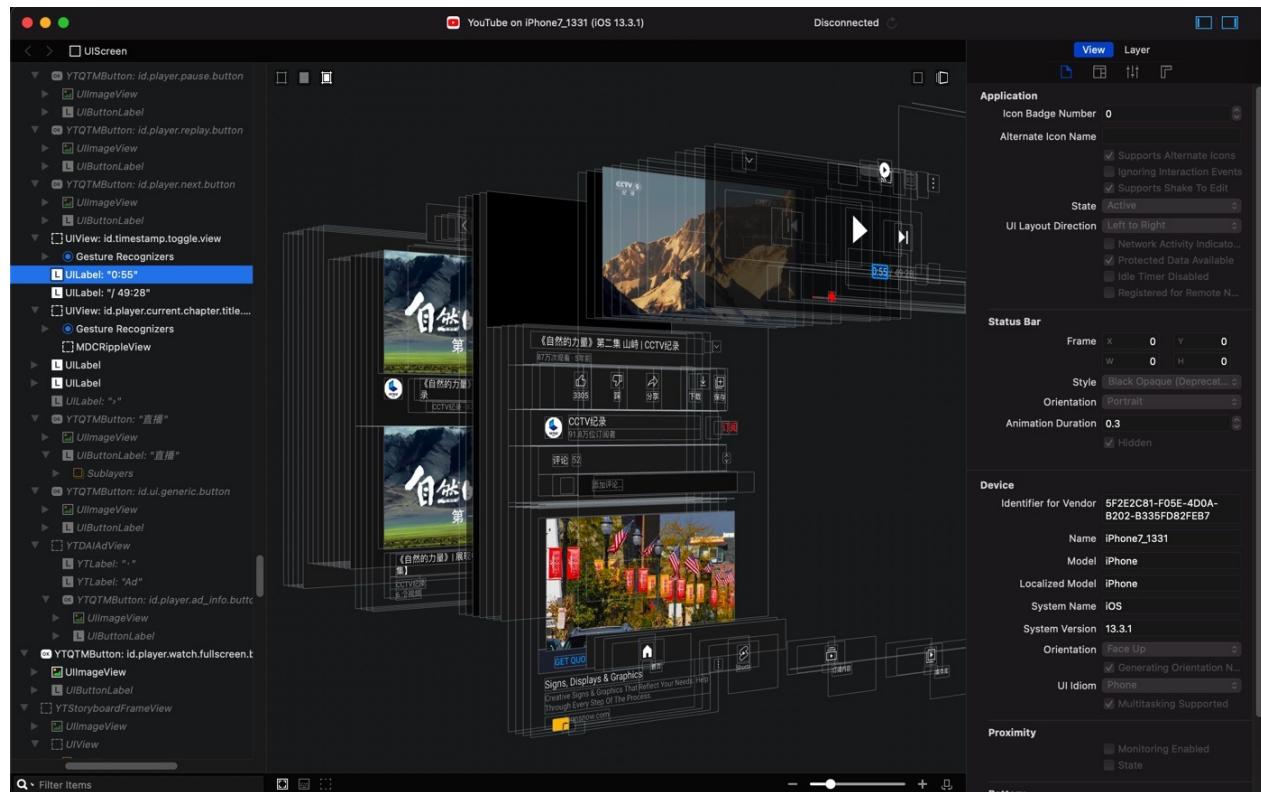
crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：
2022-10-23 16:20:14

Reveal

TODO:

- 【已解决】用Reveal查看抖音UI界面中点赞关注按钮相关的类和实现
- 【记录】找抖音关注按钮响应事件: pactions
- 【记录】通过Reveal查看页面元素找YouTube广告相关类
- 【记录】通过Reveal查看YouTube广告页面元素
- 【已解决】MonkeyDev中如何使用Reveal调试YouTube广告页面元素
- 【已解决】Mac中下载和安装Reveal

iOS逆向中，用来调试界面元素，比较好用的工具之一就是： Reveal



crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2022-10-27 11:51:19

Cycript

TODO:

- 【部分解决】用Cycript查看抖音UI界面元素以寻找关注按钮所属元素
 - 【已解决】用MonkeyDev中Cycript去调试YouTube的UI页面的元素
-

iOS逆向的调试界面元素的工具，也有命令行的：[Cycript](#)

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

2022-10-23 16:04:49

LLDBTools

TODO:

- 【已解决】用MonkeyDev的LLDBTools去打印UI界面元素
-

iOS逆向调试界面元素时，也可以用：MonkeyDev的 `LLDBTools` 的相关命令，输出界面元素信息。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-23 16:05:41

chisel

chisel 本身是 Xcode 内置调试器：LLDB 的插件，其中有部分命令，也可以用来，调试打印界面元素。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook 最后更新：
2022-10-27 11:52:25

FLEX

TODO:

【整理】iOS的iPhone越狱和改机相关知识

iOS越狱插件 FLEX，可以用来辅助调试iOS的app的界面元素。

- FLEX
 - 效果
 - 当它加载时，会向目标程序上方添加一个悬浮的工具栏，通过这个工具栏可以查看和修改视图的层级结构、动态修改类的属性、动态调用实例和方法、动态查看类和框架以及动态修改UI等。
 - 截图
 -

教程



Word 文档

Excel 表格

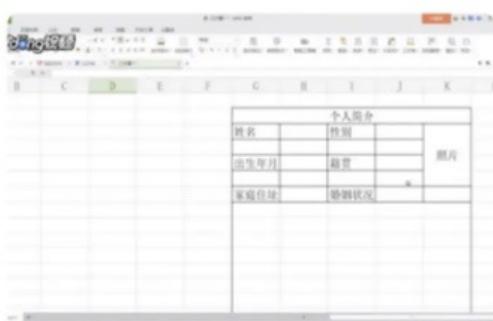
PPT 幻灯片

热门教程



将文档中的表格转化为...
存档方便快捷

如何让你的图表动起来
让你的工作报表更新颖



用表格也能制作简历
快速晋升商务办公达人



震惊！PPT究竟有多么...
PPT 进阶





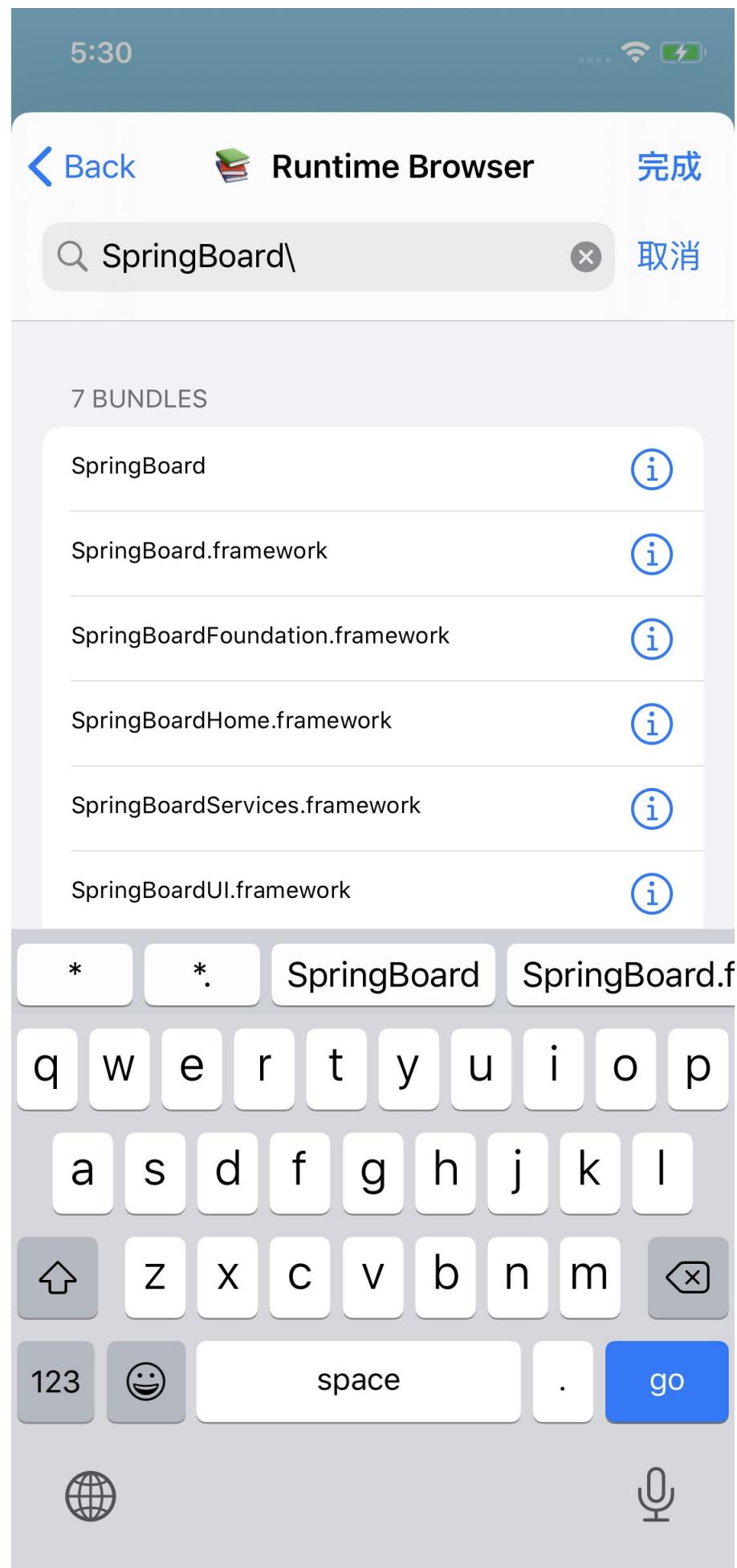
View Hierarchy Tree

完成

Filter

- UIWindow
frame {(0, 0), (414, 736)} (i)
- UITransitionView
frame {(0, 0), (414, 736)} (i)
- UIDropShadowView
frame {(0, 0), (414, 736)} (i)
- UILayoutContainerView (wordios.BaseTab)
frame {(0, 0), (414, 736)} (i)
- UITransitionView
frame {(0, 0), (414, 736)} (i)
- UIViewControllerWrapperView
frame {(0, 0), (414, 736)} (i)
- UILayoutContainerView (wordios.BaseNav)
frame {(0, 0), (414, 736)} (i)
- UINavigationTransitionView
frame {(0, 0), (414, 736)} (i)
- UIViewControllerWrapperView
frame {(0, 0), (414, 736)} (i)
- UIView (wordios.CourseVC)
frame {(0, 0), (414, 736)} (i)
- UIView
frame {(0, 90), (414, 92)} (i)
- UIView
frame {(0, 0), (138, 92)} (i)

- 好像还可以擦好看类的定义





5:30

... WiFi 🔋

返回 NSBundle 完成

NSBundle NSObject

DESCRIPTION

NSBundle </System/Library/PrivateFrameworks/
SpringBoard.framework> (loaded)

SHORTCUTS

Browse Bundle Directory >

Browse Bundle as Database... > •••

@property NSString *bundleIdentifier
com.apple.SpringBoardFramework > •••

@property Class principalClass
nil > •••

@property NSDictionary *infoDictionary
{ BuildMachineOSBuild = 1... arm64); } >

@property NSString *bundlePath
/System/Library/PrivateFrameworks/SpringBoard.framework >

@property NSString *executablePath
/System/Library/PrivateFrameworks/SpringBoard.framework/SpringBoard >

@property BOOL loaded
1 >

PROPERTIES (32)

CCUILayoutSize ccui_prototypeModuleSize

...

↑

Bookmark

Share

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-23 16:39:36

动态调试心得

TODO:

- 计算类的属性的偏移量
 - 【已解决】 调试寻找HAMPlayerInternal的_currentTime中字段的偏移量
 - 【整理】 iOS逆向心得：通过查看类的地址保存的值找到值和属性字段的偏移量和对应关系
 - 【已解决】 iOS逆向：写hook代码时打印出类的私有属性变量值的类型
 - 【整理】 iOS逆向心得：类的属性字段偏移量计算要加上isa的父类
 - C++
 - 【整理】 iOS逆向心得：Cronet相关的C++的struct结构体类的属性和函数的偏移量计算逻辑
 - 【已解决】 iOS逆向：IDA中如何逆向分析C++的vtable
 - 【整理】 iOS逆向涉及内容：C++中的vtable
 - 其他
 - 【已解决】 iOS逆向：查看NSMutableURLRequest的HTTPBody的数据
-

iOS逆向的动态调试，有很多心得，整理如下。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 12:03:43

Xcode相关

TODO:

- Xcode相关
 - 问题
 - 【已解决】XCode调试YouTube报错：Unable to install There was an internal API error
 - 心得
 - 【已解决】XCode+MonkeyDev调试YouTube：如何在广告页面停止供调试
-

lldb

- 【整理】iOS逆向心得：lldb中打印d的d0寄存器不是double而是data
- 【整理】如何找到Xcode中lldb调试出的无名的函数对应的IDA的伪代码中是哪个函数

EXC_BREAKPOINT

- EXC_BREAKPOINT
 - 【已解决】iOS逆向调试报错EXC_BREAKPOINT: HAMNetworkRequestResponseEvent的initWithRequest
 - 【已解决】Xcode中调试iOS的app再次报错：Thread EXC_BREAKPOINT code 1 subcode 0x1bf09c598
 - 【未解决】Xcode调试iOS的YouTube时objc_msgSend崩溃：Thread EXC_BREAKPOINT code 1 subcode 0x1bf09c598

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 11:59:18

ObjC

TODO:

- ObjC
 - 【整理】iOS逆向调试心得：ObjC或ARM中从偏移量中取值的不同写法
 - 【未解决】Xcode调试iOS的Objc时获取self的父类的实例
 - 【整理】iOS逆向心得：ObjC函数调用时参数顺序和汇编代码中寄存器传递的参数顺序不一致
 - 【整理】iOS逆向和IDA使用心得：调用objc_msgSend时传递给MLPlayerItemQOEErrorEvent的initWithError:fatal:absoluteTime:的参数不够
 - 【整理】iOS逆向心得：打印ObjC类的属性

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-27 12:03:20

objc_msgSend

iOS的ObjC的函数调用，比如A函数调用B函数，底层都是通过 `objc_msgSend` 实现的。

所以iOS逆向期间，涉及到最多的，应该就属 `objc_msgSend` 了。

所以关于 `objc_msgSend` 也有很多心得，整理如下。

不带`lldb_unnamed_symbol`的无名的bl，往往是更重要的，我们所关注的`objc_msgSend`

折腾：

【未解决】研究抖音关注逻辑：`__lldb_unnamed_symbol1588524$$AwemeCore`

期间，调试到目前的心得：

如果是带 `__lldb_unnamed_symbol` 的写法，往往不是主要的，我们所关心的 `objc_msgSend` 函数而无名的 `b1`，往往是重要的，我们所关注的：`objc_msgSend` 的相关调用

举例：

```
0x11427c2c8 <+336>: b1      0x115ce58fc
```

其实就是：`objc_msgSend`

而其他很多其他的bl：

```
0x11427c2b0 <+312>: b1      0x11427e920           ; __lldb_unnamed_symbol1588573$$AwemeC
```

ore

只是个 `jmp_objc_retain`，不是我们关注的重点。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-26 16:42:30

Runtime运行时

TODO:

- ObjC运行时
 - 【记录】iOS中的ObjC的函数: dispatch_async
 - 【已解决】iOS逆向心得: OS_dispatch_data
 - 【已解决】iOS底层函数: objc_enumerationMutation

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-27 12:03:11

po

TODO:

- po失效
 - 【无需解决】Xcode中lldb调试iOS的ObjC的汇编代码时：偶尔po失效打印不出变量类型
 - 【未解决】Xcode中lldb的po再次失效尝试搞懂内部原因

[LLDB中的调试命令po](#)，也是iOS逆向期间，用的最多的命令：用于查看某个地址，具体是什么（iOS的ObjC的）类。

对于 `po`，也有很多经验和心得，整理如下。

类Class 对比 实例Instance

折腾：

【记录】XCode+MonkeyDev动态调试YouTube类：YTWatchMiniBarViewController

期间，可以通过hook代码：

```
hook YTWatchController

// - (void)playbackControllerDidLoadPlayerWithPlaybackData:(id)arg1;
- (void)playbackControllerDidLoadPlayerWithPlaybackData:(id)arg1{
    iosLogInfo("arg1=%@", arg1);
    orig;
}

end
```

而输出log：

```
2022 03 27 17 16 50.049397 0800 YouTube[25245 2617390] hook_ youtubeDylib.xm YTWatchController$ 
playbackControllerDidLoadPlayerWithPlaybackData$ arg1= YTPlaybackData 0x282153e70
```

而另外可以通过 `po` 查看到类 `YTPlaybackData` 的信息

```
(lldb) po [objc_getClass("YTPlaybackData") _shortMethodDescription]
YTPlaybackData 0x1085737c0
in YTPlaybackData
  Class Methods
    + (id) playbackDataWithPlayerResponse:(id)arg1 CPN:(id)arg2; (0x1032d8060)
    + (id) playbackDataWithAd:(id)arg1; (0x102b2a228)
    + (id) playbackDataWithPlayerResponse:(id)arg1; (0x1032d8050)
  Properties
    @property (readonly, nonatomic) YTPlayerResponse* playerResponse; (@synthesize playerResponse = _playerResponse;)
```

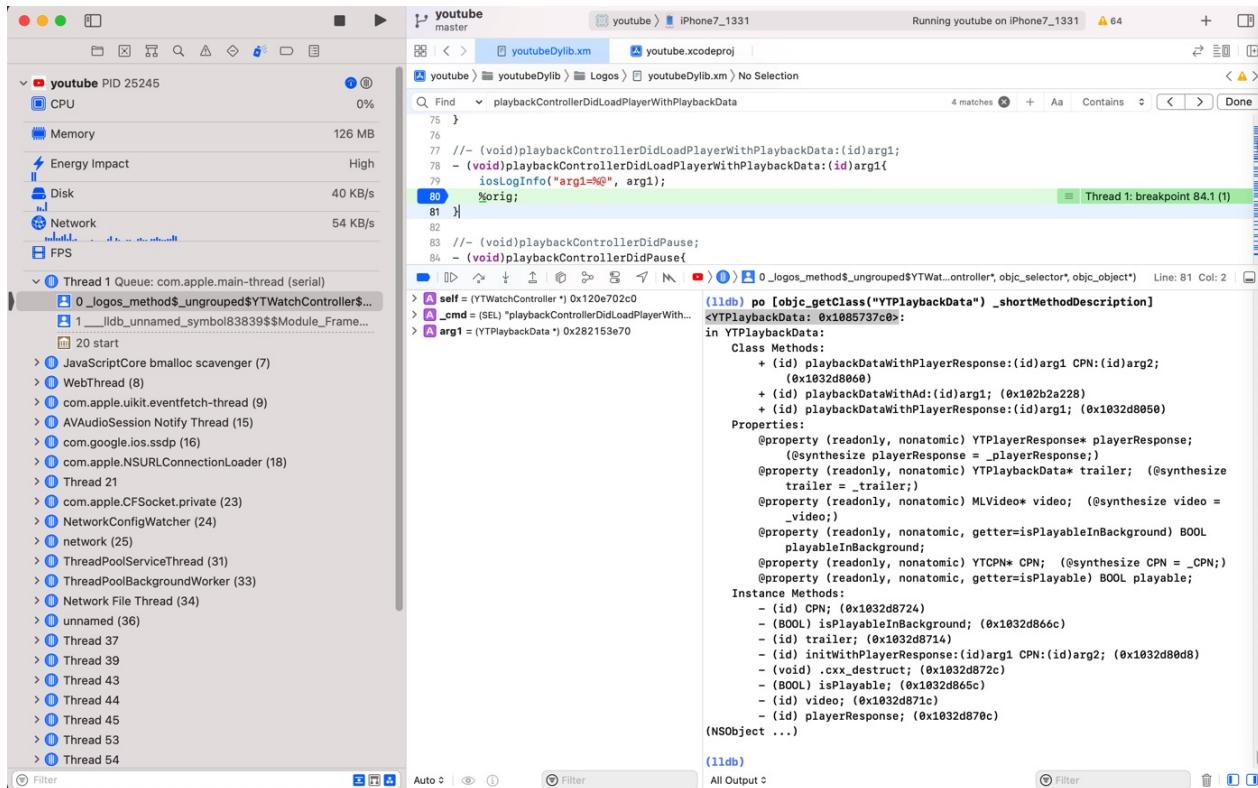
```

@property (readonly, nonatomic) YTPlaybackData* trailer; (@synthesize trailer = _trailer)
@property (readonly, nonatomic) MLVideo* video; (@synthesize video = _video)
@property (readonly, nonatomic, getter=isPlayableInBackground) BOOL playableInBackground;
@property (readonly, nonatomic) YTCPN* CPN; (@synthesize CPN = _CPN)
@property (readonly, nonatomic, getter=isPlayable) BOOL playable;

Instance Methods:
- (id) CPN; (0x1032d8724)
- (BOOL) playableInBackground; (0x1032d866c)
- (id) trailer; (0x1032d8714)
- (id) initWithPlayerResponse (id)arg1 CPN (id)arg2; (0x1032d80d8)
- (void) .cxx_destruct; (0x1032d872c)
- (BOOL) playable; (0x1032d865c)
- (id) video; (0x1032d871c)
- (id) playerResponse; (0x1032d870c)

(NSObject ...)

```



两者对比：

- **Class = 类**

- 无需遇到对应类的变量，任何时候，只要代码加载到内存了，即可查看具体的内容
- 查看类的信息的方式：

```
po [objc_getClass("YTPlaybackData") _shortMethodDescription]
```

- 场景举例

- 比如给 YouTube 加了断点 `UIApplicationMain`，断点生效时，即可查看类 `YTPlaybackData` 的信息，而无需实际调试找到 `YTPlaybackData` 的实例变量

- **Instance = 实例**

- 只有遇到对应的变量类型了，才能看到具体的值

- 比如，此处是运行到函数playbackControllerDidLoadPlayerWithPlaybackData的内部，`YTPlaybackData`作为参数，所以才能看到具体的Instance实例的值
 - 查看示例变量值的方式

- 举例

```
(lldb) po [(YTPlaybackData*) 0x282153e70 isPlayable]
true

(lldb) po [(YTPlaybackData*) 0x282153e70 isPlayableInBackground]
false

(lldb) po [(YTPlaybackData*) 0x282153e70 video]
MLVideo 0x282149da0

(lldb) po [(YTPlaybackData*) 0x282153e70 trailer]
nil
```

- 即可看到，当前类 `YTPlaybackData` 的实例：`<0x282153e70>` 的各种属性值

用po打印Class类的属性Property或函数Method

- 打印Class的属性或函数值

```
po [ClassName classMethodOrProperty]
```

- 打印Class的Instance的属性或函数之

```
po [objc_getClass("ClassName") instanceMethodOrProperty]
```

举例：

```
(lldb) po [objc_getClass("TTMacroManager") _shortMethodDescription]
<TTMacroManager 0x103b56e48>
in TTMacroManager
  Class Methods:
    + (BOOL) isBDWEBIMAGE_APP_EXTENSIONS; (0x117cf1d50)
    + (BOOL) isDebug; (0x117cf1e28)
  (NSObject ...)
```

进一步的，对应着（导出抖音的）头文件：

```
#import <objc/Object.h>

@interface TTMacroManager : NSObject
{
}

+ (_Bool)isBDWEBIMAGE_APP_EXTENSIONS;
+ (_Bool)isDebug;
```

@end

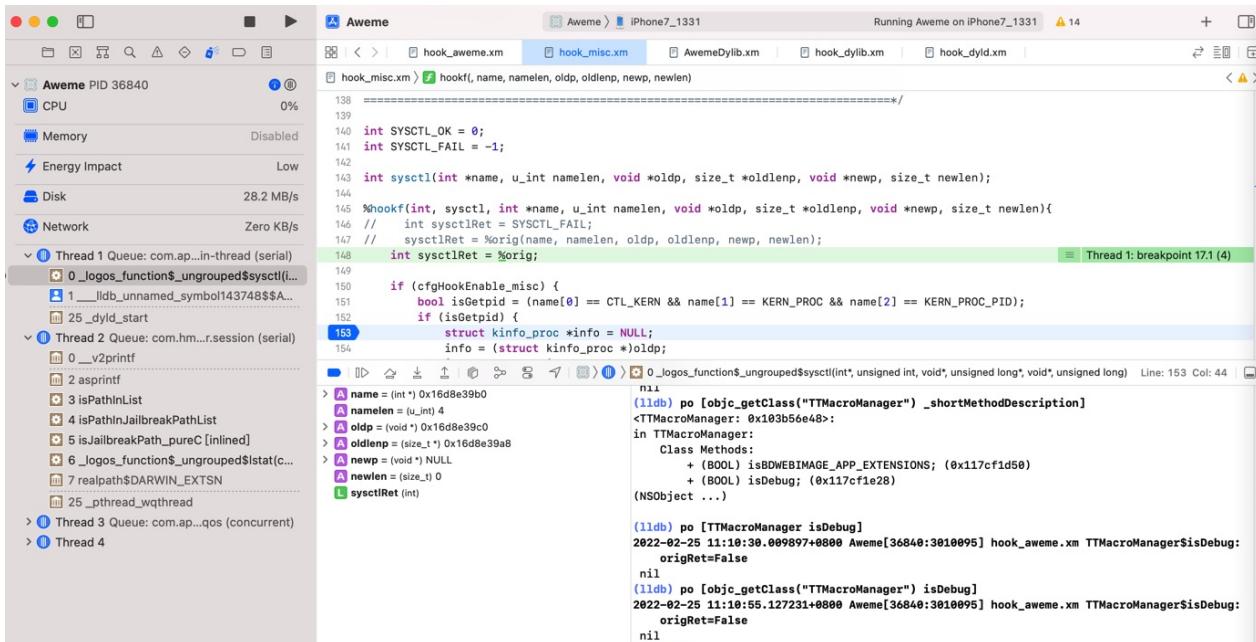
想要查看当前的Class的值，则是：

```
(lldb) po [TTMacroManager isDebug]
2022-02-25 11:10:30.009897+0800 Aweme[36840:3010095] hook_aweme.xm TTMacroManager$isDebug orig
Ret False
nil
```

注：此处输出的是被我加了hook了的代码的log

如果想要查看实例instance的值，则是：

```
(lldb) po [objc_getClass("TTMacroManager") isDebug]
2022-02-25 11:10:55.127231+0800 Aweme[36840:3010095] hook_aweme.xm TTMacroManager$isDebug orig
Ret False
nil
```



po 失效时换用 object_getClassName 查看是什么类

iOS逆向期间，正常的话，po 是可以打印出某个地址，具体是什么（ObjC的）类

比如：

```
(lldb) po 0x0000000137419800
AWEInviteSearchTableViewCell: 0x13741980; baseClass = UITableViewCell; frame = (0 152 375 76)
; autoresizingMask = W; layer = <CALayer: 0x28f694b00>
```

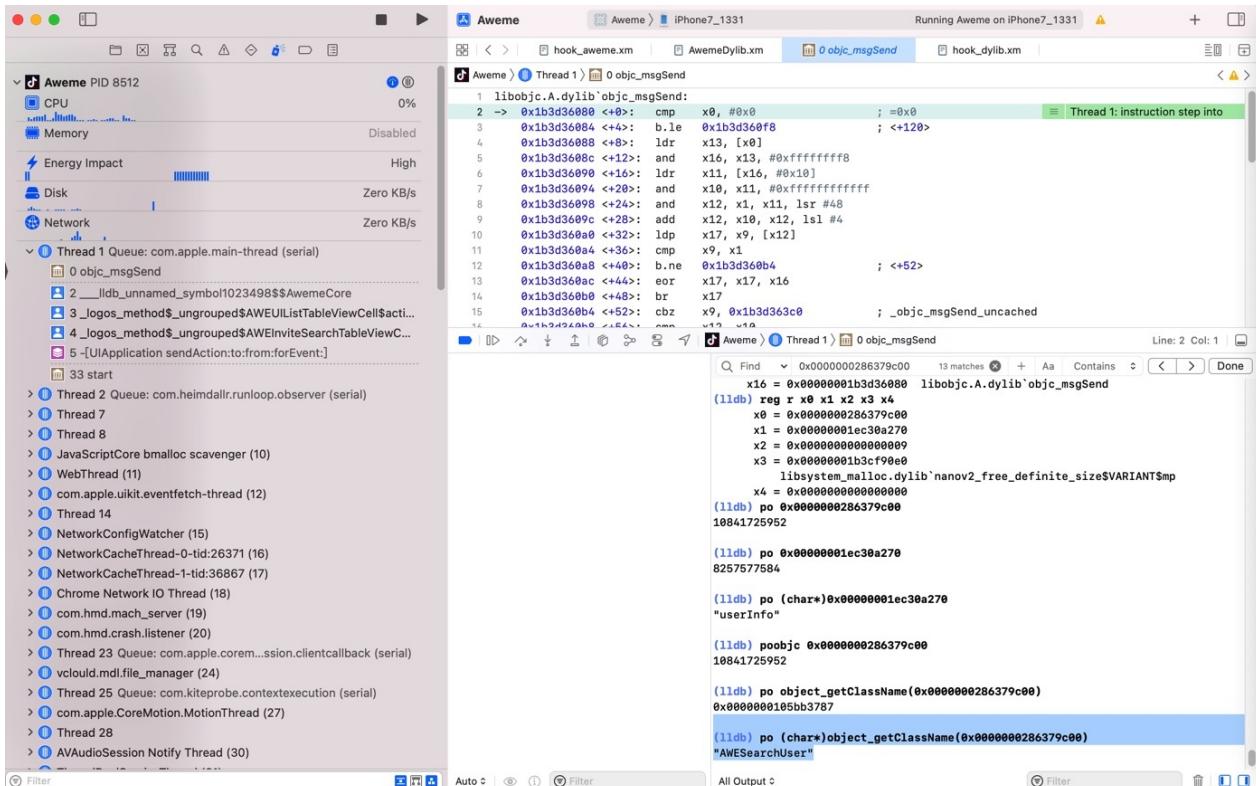
-> 从而通过调试搞懂代码的具体逻辑：调用了什么(ObjC的)类的什么函数。

而有时候，不知何故，`po` 失效，则打印不出来是什么类：

```
(lldb) po 0x0000000286379c00
10841725952
```

此时，可以换用：`object_getClassName`

```
(lldb) po (char*)object_getClassName(0x0000000286379c00)
"AWEUserSearch"
```



po查看类的描述的同时可以看到父类的相关定义

比如抖音的：

`Aweme_classDump/Aweme_17.8.0_header/Aweme/AWEPlayInteractionFollowSuccessElement.h`

```
#import "AWEPlayInteractionBottomElement.h"

#import "AWEUserMessage-Protocol.h"

@class AWEAntiAddictedNoticeBarView, AWEHistoryPublicDataController, NSString;

@interface AWEPlayInteractionFollowSuccessElement : AWEPlayInteractionBottomElement <AWEUserMessage>
{
    AWEAntiAddictedNoticeBarView *_antiAddictedNoticeBarView;
    AWEHistoryPublicDataController *_listDataController;
    long long _actionType;
}
```

```

- (void).cxx_destruct;
@property(nonatomic) long long actionType; // @synthesize actionType=_actionType;
@property(retain, nonatomic) AWEHistoryPublicDataController *listDataController; // @synthesize
listDataController=_listDataController;
@property(retain, nonatomic) AWEAntiAddictedNoticeBarView *antiAddictedNoticeBarView; // @synth
esize antiAddictedNoticeBarView=_antiAddictedNoticeBarView;
- (void)didFinishUnFollowUser:(id)arg1 status:(long long)arg2 error:(id)arg3;
- (void)p_hideAntiAddictedNoticeBarView:(long long)arg1 animation:(CDUnknownBlockType)arg2;
- (void)p_showAntiAddictedNoticeBarViewWithCompletion:(CDUnknownBlockType)arg1;
- (void)noticeTapped;
- (void)showFollowSuccessNoticeBar:(id)arg1;
- (void)hideMutexTempElement:(CDUnknownBlockType)arg1;
- (void)dealloc;
- (void)viewDidDisposed;
- (void)reset;
- (void)viewDidLoad;
- (void)initializeElement;

// Remaining properties
@property(nonatomic, copy) NSString *debugDescription;
@property(nonatomic, copy) NSString *description;
@property(nonatomic) unsigned long long hash;
@property(nonatomic) Class superclass;

@end

```

去Xcode的lldb中动态调试抖音期间，通过：

```
po [objc_getClass("AWEPlayInteractionFollowSuccessElement") _shortMethodDescription]
```

- 不仅能看到：类 AWEPlayInteractionFollowSuccessElement 本身的信息
- 还能看到：父类 AWEPlayInteractionBottomElement
 - 父类的父类： AWEPlayInteractionBottomElement
 - 父类的父类的父类： AWEPlayInteractionBaseElement
 - 父类的父类的父类的父类： AWEBaseElement
 - 直到最后的根对象： NSObject

具体输出内容是：

```
(lldb) po [objc_getClass("AWEPlayInteractionFollowSuccessElement") _shortMethodDescription]
2022 04 02 13:36:53.128548 0800 Aweme[45939:3543378] hook_msc.xm NSBundle$bundlePath: origBund
lePath /usr/lib
AWEPlayInteractionFollowSuccessElement 0x10629ab90>;
in AWEPlayInteractionFollowSuccessElement
Properties
@property (retain, nonatomic) AWEAntiAddictedNoticeBarView *antiAddictedNoticeBarView;
@synthesize antiAddictedNoticeBarView = _antiAddictedNoticeBarView;
@property (retain, nonatomic) AWEHistoryPublicDataController *listDataController; (@sy
nthesize listDataController = _listDataController;
@property (nonatomic) long actionType; (@synthesize actionType = _actionType;)
@property (readonly) unsigned long hash;
@property (readonly) Class superclass;
```

```

@property (readonly, copy) NSString* description;
@property (readonly, copy) NSString* debugDescription;

Instance Methods:
- (void) didFinishUnFollowUser (id)arg1 status:(long)arg2 error (id)arg3; (0x1158b84b4)
- (void) viewDidDisposed; (0x1158b6a8c)
- (void) initializeElement; (0x1158b672c)
- (id) listDataController; (0x1158b86f0)
- (void) setListDataController (id)arg1; (0x1158b8758)
- (void) noticeTapped; (0x1158b731c)
- (void) setAntiAddictedNoticeBarView (id)arg1; (0x1158b8748)
- (id) antiAddictedNoticeBarView; (0x1158b859c)
- (void) hideMutexTempElement (^block)arg1; (0x1158b6b4c)
- (void) p_hideAntiAddictedNoticeBarView (long)arg1 animation (^block)arg2; (0x1158b7f74)
)

- (void) p_showAntiAddictedNoticeBarViewWithCompletion:(block)arg1; (0x1158b78a4)
- (void) showFollowSuccessNoticeBar:(id)arg1; (0x1158b6e7c)
- (void) dealloc; (0x1158b6ad8)
- (void) .cxx_destruct; (0x1158b8788)
- (void) reset; (0x1158b69fc)
- (void) viewDidLoad; (0x1158b67b4)
- (long) actionType; (0x1158b8768)
- (void) setActionType:(long)arg1; (0x1158b8778)

in AWEPlayInteractionBottomElement:
Instance Methods:
- (void) configWithParamDict (id)arg1; (0x10b014ce0)
- (id) bottomElementContainer; (0x10b014df8)
- (BOOL) elementAppearLowPriorityNeedAvoid; (0x115851c34)
- (void) updateNextElementAppearStatus; (0x1158519a0)
- (void) reset; (0x115851b1c)

in AWEPlayInteractionBaseElement:
Properties
@property (retain, nonatomic) AWEAwemeModel* model; (@synthesize model = _model;)
@property (nonatomic) unsigned long playerStatus; (@synthesize playerStatus = _playerStatus;)
@property (weak, nonatomic) NSPointerArray allElements; (@synthesize allElements = _allElements;)
@property (readonly) unsigned long hash;
@property (readonly) Class superclass;
@property (readonly, copy) NSString* description;
@property (readonly, copy) NSString* debugDescription;

Instance Methods:
- (struct CGRect) viewFrame; (0x11584af68)
- (void) videoDidActivity; (0x10b0937f8)
- (BOOL) alertIfNotValidForAction:(long)arg1; (0x11584aae8)
- (id) elementFromAll:(id)arg1; (0x10b043ad8)
- (void) viewController_viewWillDisappear; (0x11584abc4)
- (void) viewController_viewDidDisappear; (0x11584abc8)
- (void) viewController_didEndDisplaying; (0x11584abcc)
- (void) viewController_willDisplay; (0x11584abb4)
- (void) viewController_viewWillAppear; (0x11584abbc)
- (void) viewController_viewDidAppear; (0x11584abc0)
- (void) hideAllElementExcepts (id)arg1; (0x11584a90c)
- (void) updateAllElement; (0x11584ab74)
- (void) setAllElements:(id)arg1; (0x10b019980)
- (id) currentInfoForUnitWithIdentifier (id)arg1; (0x11584aed8)
- (void) hideProgressSliderPopView; (0x11584ac74)

```

```

- (id) currentInfoForSubUnits; (0x11584adcc)
- (id) currentInfoForUnitWithClassName (id)arg1; (0x11584ae5c)
- (void) dealloc; (0x11584abe4)
- (void) .cxx_destruct; (0x11584b090)
- (void) pause; (0x11584ac54)
- (void) resume; (0x11584ac64)
- (void) setData (id)arg1; (0x10b04aecc)
- (id) context; (0x10b00ef84)
- (void) reset; (0x11584abd0)
- (id) model; (0x10b016d44)
- (void) setModel (id)arg1; (0x10b01cf58)
- (void) play; (0x11584abe0)
- (void) prepareForDisplay; (0x10b06b068)
- (BOOL) isShowing; (0x11584b074)
- (void) didEndDisplaying; (0x11584abb8)
- (id) currentInfo; (0x11584acc0)
- (unsigned long) playerStatus; (0x11584b080)
- (void) setHide (BOOL)arg1; (0x10b0899c0)
- (void) setPlayerStatus:(unsigned long)arg1; (0x10b0431dc)
- (id) allElements; (0x10b043c60)

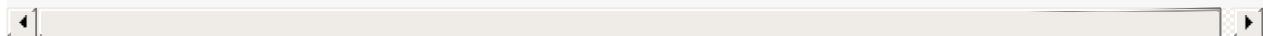
in AWEBaseElement
Properties
@property (weak, nonatomic) AWEElementContainer* elementContainer; (@dynamic elementContainer;
)
@property (weak, nonatomic) UIView* boxView; (@synthesize boxView = _boxView;
)
@property (weak, nonatomic) UIView* elementView; (@synthesize elementView = _elementView;
)
@property (nonatomic) BOOL hascreateView; (@synthesize hascreateView = _hascreateView;
)
@property (retain, nonatomic) AWEPageContext* context; (@synthesize context = _context;
)
@property (weak, nonatomic) AWEElementContainer* elementContainer; (@synthesize elementContainer = _elementContainer;
)
@property (retain, nonatomic) UIView* view; (@synthesize view = _view;
)
@property (retain, nonatomic) id data; (@synthesize data = _data;
)
@property (readonly, nonatomic, getter = isViewLoaded) BOOL viewLoaded;
@property (copy, nonatomic) NSString* identity; (@synthesize identity = _identity;
)
@property (nonatomic) BOOL appear; (@synthesize appear = _appear;
)
@property (readonly, weak, nonatomic) UIViewController* viewController; (@synthesize viewController = _viewController;
)
@property (readonly) unsigned long hash;
@property (readonly) Class superclass;
@property (readonly, copy) NSString* description;
@property (readonly, copy) NSString* debugDescription;

Instance Methods
- (void) configWithParamDict (id)arg1; (0x11232b734)
- (void) viewDidDisposed; (0x11232b730)
- (void) processAppear:(BOOL)arg1; (0x10b010138)
- (id) elementContainer; (0x10b01036c)
- (void) setAppear (BOOL)arg1; (0x10b01d350)
- (void) initializeElement; (0x10b00f048)
- (void) setElementContainer (id)arg1; (0x10b00ed6c)
- (id) boxView; (0x11232b8fc)
- (void) setBoxView (id)arg1; (0x11232b914)
- (id) elementView; (0x11232b920)
- (void) setElementView:(id)arg1; (0x11232b938)

```

```
- (BOOL) hasCreateView; (0x11232b944)
- (void) setHasCreateView:(BOOL)arg1; (0x11232b94c)
- (void) addSubviewWithLayout (id)arg1 withEdgeInsets (struct UIEdgeInsets)arg2; (0x11232b9ec)
- (void) addSubviewWithLayout (id)arg1 withEdgeInsets (struct UIEdgeInsets)arg2 withHeight (double)arg3; (0x11232b9fc)
- (void) hide:(BOOL)arg1 duration:(double)arg2 animations:(block)arg3; (0x11232bcc4)
- (void) hide:(BOOL)arg1 duration:(double)arg2 withTransform (struct CGAffineTransform) arg3 animations (^block)arg4; (0x11232bd48)
- (void) addSubviewWithLayout (id)arg1; (0x11232b9d0)
- (void) hide:(BOOL)arg1 duration:(double)arg2; (0x11232bcb4)
- (void) .cxx_destruct; (0x11232b968)
- (id) data; (0x11232b8e8)
- (void) setData (id)arg1; (0x11232b6cc)
- (id) context; (0x11232b954)
- (id) identity; (0x11232b8f0)
- (void) setContext (id)arg1; (0x11232b95c)
- (void) setIdentity (id)arg1; (0x10b014df0)
- (id) view; (0x10b0101fc)
- (void) setView (id)arg1; (0x11232b8dc)
- (void) loadView; (0x10b0102e0)
- (void) viewDidLoad; (0x11232b72c)
- (BOOL) isViewLoaded; (0x10b0101ec)
- (id) viewController; (0x10b012c08)
- (BOOL) appear; (0x10b00f010)

(NSObject ...)
```



Aweme > Thread 161 > 0 -[NSString stringByAppendingString:] Line: 2 Col: 1

```
(lldb) po [objc_getClass("AWEPlayInteractionFollowSuccessElement")
shortMethodDescription]
2022-04-02 13:36:53.128548+0800 Aweme[45939:3543378] hook_misc.xm NSBundle$bundlePath:
origBundlePath=/usr/lib
<AWEPlayInteractionFollowSuccessElement: 0x10629ab90>:
in AWEPlayInteractionFollowSuccessElement:
Properties:
@property (retain, nonatomic) AWEAntiAddictedNoticeBarView*
antiAddictedNoticeBarView; (@synthesize antiAddictedNoticeBarView =
_antiAddictedNoticeBarView;
@property (retain, nonatomic) AWEHistoryPublicDataController* listDataController;
(@synthesize listDataController = _listDataController;
@property (nonatomic) long actionType; (@synthesize actionType = _actionType;
@property (readonly) unsigned long hash;
@property (readonly) Class superclass;
@property (readonly, copy) NSString* description;
@property (readonly, copy) NSString* debugDescription;
Instance Methods:
- (void) didFinishUnFollowUser:(id)arg1 status:(long)arg2 error:(id)arg3;
(0x1158b84b4)
- (void) viewDidDisposed; (0x1158b6a8c)
- (void) initializeElement; (0x1158b672c)
- (id) listDataController; (0x1158b86f0)
- (void) setListDataController:(id)arg1; (0x1158b8758)
- (void) noticeTapped; (0x1158b731c)
- (void) setAntiAddictedNoticeBarView:(id)arg1; (0x1158b8748)
- (id) antiAddictedNoticeBarView; (0x1158b859c)
- (void) hideMutexTempElement:(^block)arg1; (0x1158b6b4c)
- (void) p_hideAntiAddictedNoticeBarView:(long)arg1 animation:(^block)arg2;
(0x1158b7f74)
- (void) p_showAntiAddictedNoticeBarViewWithCompletion:(^block)arg1; (0x1158b78a4)
- (void) showFollowSuccessNoticeBar:(id)arg1; (0x1158b6e7c)
- (void) dealloc; (0x1158b6ad8)
- (void) .cxx_destruct; (0x1158b8788)
- (void) reset; (0x1158b69fc)
- (void) viewDidLoad; (0x1158b67b4)
```

All Output

Filter



```

Aweme > Thread 161 > 0 -[NSString stringByAppendingString:] Line: 2 Col: 1
  instance methods:
    - (void) didFinishUnFollowUser:(id)arg1 status:(long)arg2 error:(id)arg3;
      (0x1158b84b4)
    - (void) viewDidDisposed; (0x1158b6a8c)
    - (void) initializeElement; (0x1158b672c)
    - (id) listDataController; (0x1158b86f0)
    - (void) setListDataController:(id)arg1; (0x1158b8758)
    - (void) noticeTapped; (0x1158b731c)
    - (void) setAntiAddictedNoticeBarView:(id)arg1; (0x1158b8748)
    - (id) antiAddictedNoticeBarView; (0x1158b859c)
    - (void) hideMutexTempElement:(^block)arg1; (0x1158b6b4c)
    - (void) p_hideAntiAddictedNoticeBarView:(long)arg1 animation:(^block)arg2;
      (0x1158b7f74)
    - (void) p_showAntiAddictedNoticeBarViewWithCompletion:(^block)arg1; (0x1158b78a4)
    - (void) showFollowSuccessNoticeBar:(id)arg1; (0x1158b6e7c)
    - (void) dealloc; (0x1158b6ad8)
    - (void) .cxx_destruct; (0x1158b8788)
    - (void) reset; (0x1158b69fc)
    - (void) viewDidLoad; (0x1158b67b4)
    - (long) actionType; (0x1158b8768)
    - (void) setActionType:(long)arg1; (0x1158b8778)

in AWEPlayInteractionBottomElement:
  Instance Methods:
    - (void) configWithParamDict:(id)arg1; (0x10b014ce0)
    - (id) bottomElementContainer; (0x10b014df8)
    - (BOOL) elementAppearLowPriorityNeedAvoid; (0x115851c34)
    - (void) updateNextElementAppearStatus; (0x1158519a0)
    - (void) reset; (0x115851b1c)

in AWEPlayInteractionBaseElement:
  Properties:
    @property (retain, nonatomic) AWEAwemeModel* model; (@synthesize model = _model;)
    @property (nonatomic) unsigned long playerStatus; (@synthesize playerStatus =
      _playerStatus;)
    @property (weak, nonatomic) NSPointerArray* allElements; (@synthesize
      allElements = _allElements;)

All Output ◊ Filter

```

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2022-10-27 11:54:30

子教程

iOS逆向的动态调试中，已把部分独立内容整理到子教程：

- MonkeyDev
 - [iOS逆向开发：MonkeyDev调试 \(crifan.org\)](#)

相关代码：

- iOSYouTubeAdsFilter
 - [crifan/iOSYouTubeAdsFilter: MonkeyDev+Xcode项目，iOS逆向YouTube，尝试实现广告过滤功能](#)

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

2022-11-08 09:35:35

附录

下面列出相关参考资料。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-03-17 20:39:28

参考资料

- 【记录】XCode+MonkeyDev动态调试YouTube类：YTWatchMiniBarViewController
-
- [iOS逆向攻防实战 - 掘金 \(juejin.cn\)](#)
- [SpringBoard tweak 双击图标启动debugserver - 干货分享 - 睿论坛](#)
-

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：
2022-10-26 16:49:43