
目录

前言	1.1
Frida调试iOS心得概述	1.2
经验心得	1.3
常用iOS函数	1.3.1
onLeave获取self实例	1.3.2
常见问题	1.4
用ApiResolver触发函数但没日志	1.4.1
XinaA15中的frida	1.4.2
need Gadget to attach on jailed iOS	1.4.3
导致iPhone重启	1.4.4
Waiting for USB device to appear	1.4.5
unable to intercept function at	1.4.6
TypeError: cannot read property 'implementation' of undefined	1.4.7
附录	1.5
参考资料	1.5.1

Frida调试iOS经验总结

- 最新版本： v1.0.0
- 更新时间： 20251213

简介

把之前用iOS逆向期间，用Frida调试iOS的程序的各種經驗心得和常見問題整理過來供參考。

源码+浏览+下载

本書的各種源碼、在線瀏覽地址、多種格式文件下載如下：

HonKit源碼

- [crifan/frida_debug_ios_summary: Frida调试iOS经验总结](#)

如何使用此HonKit源碼去生成發布為電子書

詳見：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [Frida调试iOS经验总结 book.crifan.org](#)
- [Frida调试iOS经验总结 crifan.github.io](#)

离线下载阅读

- [Frida调试iOS经验总结 PDF](#)
- [Frida调试iOS经验总结 ePub](#)
- [Frida调试iOS经验总结 Mobi](#)

版权和用途说明

此電子書教程的全部內容，如無特別說明，均為本人原創。其中部分內容參考自網絡，均已備注了出處。如發現有侵權，請通過郵箱联系我 [admin 艾特 crifan.com](#)，我會盡快刪除。謝謝合作。

各種技術類教程，僅作為學習和研究使用。請勿用於任何非法用途。如有非法用途，均與本人無關。

鸣谢

感謝我的老婆陳雪的包容理解和悉心照料，才使得我 [crifan](#) 有更多精力去專注技術專研和整理歸納出這些電子書和技術教程，特此鳴謝。

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：
2025-12-13 23:06:59

Frida调试iOS心得概述

之前用Frida调试iOS的各种程序，有各种经验心得和常见问题，整理过来，供参考。

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：
2025-12-13 22:58:37

Frida调试iOS经验心得

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:
2025-12-13 22:59:45

常用iOS函数

Frida中逆向iOS的app期间，常会涉及到，一些相对通用的，常用的函数，整理如下：

- 网络相关

- 请求=request

- NSURLRequest

```
+ [NSURLRequest requestWithURL:]
+ [NSURLRequest requestWithURL:cachePolicy:timeoutInterval:]
- [NSURLRequest requestWithURL:cachePolicy:timeoutInterval:]
- [NSURLRequest initWithURL:]
- [NSURLRequest initWithURL:cachePolicy:timeoutInterval:]
```

- NSMutableURLRequest

```
- [NSMutableURLRequest initWithURL:]
- [NSMutableURLRequest setHTTPBody:]
```

- NSURL

```
+ [NSURL URLWithString:]
+ [NSURL URLWithString:relativeToURL:]
- [NSURL initWithScheme:host:path:]
- [NSURL initWithString:]
- [NSURL initWithString:relativeToURL:]
```

- NSURLConnection

```
+ [NSURLConnection sendSynchronousRequest:returningResponse:error:]
+ [NSURLConnection sendAsynchronousRequest:queue:completionHandler:]
+ [NSURLConnection connectionWithRequest:delegate:]
- [NSURLConnection initWithRequest:delegate:]
- [NSURLConnection start]
- [NSURLConnection cancel]
- [NSURLConnection connection:didReceiveResponse:]
- [NSURLConnection connection:didReceiveData:]
- [NSURLConnection connectionDidFinishLoading:]
- [NSURLConnection connection:didFailWithError:]
```

- 响应=response

- NSHTTPURLResponse

```
- [NSHTTPURLResponse initWithURL:statusCode:HTTPVersion:headerFields:]
- [NSHTTPURLResponse allHeaderFields]
- [NSHTTPURLResponse statusCode]
```

- NSURLResponse

```
- [NSURLResponse initWithURL:MIMETYPE:expectedContentLength:textEncodingName:]
```

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:
2023-10-24 17:48:36

onLeave中如何获取到self类本身的实例变量

- 思路： onEnter 时，把self变量赋值保存给 this.xxx ， onLeave 时去读取即可
- 代码

```

Interceptor.attach(curMethod.implementation, {
  onEnter: function(args) {
    this.argSelfObj = argSelfObj
    ...
  },

  onLeave: function(retval) {
    var retValObj = ObjC.Object(retval);
    var retValObjStr = retValObj.toString();
    console.log("onLeave: retValObj=" + retValObj + ",retValObjStr=" + retValObjStr)

    const curSelfObj = this.argSelfObj
    console.log("curSelfObj=" + curSelfObj)
  }
})

```

- 输出

```

===== -[NSXPCConnection setExportedObject:] =====
argSelfObj:  ATrackedXPCConnection: 0x154907c50> connection to service named com.
apple.accounts.accountmanager
onLeave: retValObj=nil,retValObjStr nil
curSelfObj= ATrackedXPCConnection: 0x154907c50 connection to service named com.ap
ple.accounts.accountmanager

```

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:
2025-05-25 10:34:34

Frida调试iOS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:
2025-12-13 22:59:59

用ApiResolver去hook已触发函数但没日志

- 问题：之前用了 `ApiResolver` 去写了hook代码，后续调试的函数确保被触发了，但是却没输出我们希望的日志
 - 注：`ApiResolver` 的完整代码详见
 - [ApiResolver · Frida逆向实例和工具函数](#)

```
var resolver = new ApiResolver('objc');
resolver.enumerateMatches('*[AAUISignInViewController *]', {
  onMatch: function(match) {
    console.log(match['name'] + ":" + match['address']);
  },

  onComplete: function() {}
});
```

- 原因：此处 `ApiResolver` 只是search查找，函数是否存在，只运行一次，后续的确，本身就不会再次触发，所以后续代码被触发时，就不会打印日志
- 解决办法：真正的要实现，hook函数，待函数被运行后触发日志打印，应该改用：`Interceptor`
- 具体写法
 - 注：
 - `Interceptor` 的完整示例代码详见：[Interceptor=hook函数 · Frida逆向实例和工具函数](#)

```
// https://github.com/noobpk/frida-ios-hook/blob/master/frida-ios-hook/frida-scripts/hook-specific-method-of-class.js
function hook_specific_method_of_class(className, funcName)
{
  // console.log("className=" + className + ", funcName=" + funcName)
  var curClass = ObjC.classes[className];
  if (typeof(curClass) !== 'undefined') {
    var curMethod = curClass[funcName];
    if (typeof(curMethod) !== 'undefined') {
      Interceptor.attach(curMethod.implementation, {
        onEnter: function(args) {
          console.log("argSelf=" + args[0] + ", argSel=" + args[1] + ", realArg1=" + args[2]);
        }
      });
    }
  }
  else{
    console.log("Can't find method", funcName);
  }
  else{
    console.log("Can't find class: ", className);
  }
}

//Your class name and function name here
hook_specific_method_of_class("NSXPCCConnection", "- setExportedObject:")
```

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:
2025-05-25 10:42:12

XinaA15中的frida

- XinaA15
 - 截至 XinaA15 v1.1.8 + Sileo Nightly v2.4 + Frida v16.0.11 : 在rootless越狱的 xinaA15 中, 无法通过 Sileo Nightly 正常安装和使用 Frida
 - 只要一使用frida工具 (比如 `frida-ps -U` 等) 就会导致iPhone重启
 - 且安装和卸载都会出现一些异常报错
 - 安装Frida
 - `/var/jb/var/ib/Library/LaunchDaemons/re.frida.server.plist service is disabled`
 - 且此处 Sileo Nightly v2.4 中看到的最新版 Frida v16.0.13 , 竟然还会出现无法安装: 404错误



- 卸载Frida
 - `/var/jb/var/jb/Library/LaunchDaemons/re.frida.server.plist: Could not find specified service`

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:
2025-12-13 23:03:31

Failed to attach: need Gadget to attach on jailed iOS

- 问题：Mac中用frida启动iOS版抖音

```
frida -U -f com.ss.iphone.ugc.Aweme
```

- 报错

```
Failed to attach: need Gadget to attach on jailed iOS; its default location is:  
/Users/crifan/.cache/frida/gadget-ios.dylib
```

- 原因：缺少对应的Frida的 gadget 库文件
- 解决办法
 - 概述：下载对应版本的 gadget 库文件，放到对应位置（此处提示的 /Users/crifan/.cache/frida/gadget-ios.dylib ）即可
 - 详见：[Mac · 逆向调试利器：Frida](#)中的 安装Frida的gadget

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:
2025-05-22 22:30:24

导致iPhone重启

之前遇到过：

XinaA15的rootless越狱的iPhone11中，用最新版 v16.0.10 的 Frida ，但是却会导致iPhone重启（从而丢失XinaA15的越狱，需要再去恢复越狱）

但是：没有解决方案。

目前的结论是：

（XinaA15等）rootless越狱后，Frida 的使用，基本上是个大问题，有时候（某个旧版本）能用，有时候（此处新版 v16.0.10 ）却又无法正常使用。

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：
2024-07-21 15:49:46

Waiting for USB device to appear

- 问题：Mac中运行frida去调试app

```
frida -U -f com.apple.store.Jolly
```

- 但是报错

```
Waiting for USB device to appear...
```

- 原因：（连接iPhone到Mac的）USB数据线没插好
- 解决办法：重新拔插USB数据线，确保USB连接正常
 - 注：
 - 如何确认iPhone是否已插好
 - 方式1：通过爱思助手可以确认
 - 已插好：能看到iPhone详情
 - 没插好：看不到iPhone设备
 - 方式2：用frida的工具：[frida-ls-devices](#)

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：
2025-05-22 22:35:27

Error unable to intercept function at please file a bug

- 现象：Frida的hook的js代码中，函数地址写的是：

```
var akdSymbol12575_functionAddress = 0x1000a0460;
```

- 导致报错：

```
moduleName= akd moduleBaseAddress= 0x102b40000  
functionRealAddress)= 0x202be0460  
Error: unable to intercept function at 0x202be0460; please file a bug  
at value (frida/runtime/core.js:367)
```

- 原因：函数地址写错了 -》找不到函数地址 -》 所以报错
- 解决办法：确保函数地址是正确的
- 具体做法：代码改为：

```
var akdSymbol12575_functionAddress = 0xa0460;
```

- 说明

- 0x1000a0460 是加了 VM 虚拟地址后的 akd二进制内函数的地址
 - 0xa0460 ：是不带VM的，真正的函数内的偏移地址

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：
2024-07-21 15:54:49

TypeError: cannot read property 'implementation' of undefined

- 问题:

写frida的hook代码调试ObjC的函数

```
function hook_specific_method_of_class(className, funcName)
{
    var hook = ObjC.classes[className][funcName];
    Interceptor.attach(hook.implementation, {
        onEnter: function(args) {
            console.log("[*] Detected call to: " + className + " -> " + funcName);
        }
    });
}

//Your class name and function name here
hook_specific_method_of_class("AAUISignInViewController", "_nextButtonSelected:")
```

始终报错:

TypeError: cannot read property 'implementation' of undefined

- 原因: 此处iOS的ObjC的函数名 `_nextButtonSelected:` 写法有误
- 解决办法: 改为正确的写法: `- _nextButtonSelected:`
 - 注: 此处的ObjC的类的函数名的写法:
 - 加上类型是 `class` 的还是 `instance`, 所对应的 加号 `= +` 还是 减号 `= -`, 以及中间带上 空格 `=`
 - 相关代码改动:

```
hook_specific_method_of_class("AAUISignInViewController", "- _nextButtonSelected:")
```

- 完整代码详见
 - [单个类的单个函数 · Frida逆向实例和工具函数](#)

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:
2025-05-25 10:36:21

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

- 【已解决】frida-ios-dump砸壳ipa报错：need Gadget to attach on jailed iOS
- 【已解决】frida启动抖音app报错：Failed to attach need Gadget to attach on jailed iOS
- 【未解决】用Frida动态调试iOS版抖音app
- 【未解决】Mac中用Frida调试iOS版抖音
- 【已解决】用frida启动hook调试iOS抖音app
- 【记录】iOS逆向Apple账号：用frida和frida-trace去hook打印更多账号相关函数调用
-

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：
2025-12-13 22:56:28