逆向工程 iOS 开发 iOS Private API

iOS 如何进行逆向工程?

Android 的源码是开放的， 开发者很容易了解系统机制， 但是 iOS 的代码是封闭的， 单从文档上还是不足以深入的了解系统机制， 有没有方法能够对 iOS 进行逆向工程， 让开发者能够更深入了解系统? （当然这个是违反苹果规定的， 但是如果能从源码了解系统， 还是要比满世界找文档要好的多。）

添加评论 分享

10 个回答

默认排序

季逸超， Peak Labs、Magi 搜索、猛犸浏览器、Rasg…

360 人赞同

有幸被邀请回答，不过不知道您要了解的'系统机制'有多深入? ;-)
按照意图和深度的话，大概有这么几种途径与资源：

1. 为了学习框架，提升开发水平，可以看看私有API列表。iOS (Cocoa Touch)的各私有API都可以通过runtime查看获得，您可以自己写个method browser。如果觉得麻烦的话可以到Github看现成的，我收藏了俩: github.com/kennytm/ipho... 和 github.com/nst/iOS-Runt... ，但还是推荐自己来实时获取，因为iOS在更新，API也在更新。在App Store产品中使用私有API是违反苹果规定的，所以能不用这些API而实现一些功能是iOS工程师水平的体现。

2. 对iOS工程师而言，如果只是开发的话(1)也就差不多了。如果您十分有爱，想了解API以下的东西的话，依然可以利用Obj-C的runtime。可以在这里看到 opensource.apple.com/so... ，尤其是objc-runtime.m，这里提供了很多学习用的"工具"。比如经典的method_exchangeImplementations()，您可以用它研究很多黑箱过程的来龙去脉。值得一提的是，这种技巧(method swizzling)是合法的,可以在App Store 中使用! 苹果曾给使用了相关技巧的开发者发过邮件，表示出于安全性和稳定性最好不再使用，但没有禁止。

3. 如果是对系统本身感兴趣的话，不妨越狱看看。iOS和Mac OS X类似，基于Darwin，是一种UNIX系统。越狱后你就有了root权，可以安装个Terminal，装gcc都没问题的哈哈~ 接下来就像您研究Linux那样摆弄就好了。对于开发者来说，有了root权也就可以写一些system tweak或全局的代码，自然也可以用来深入了解系统、原生app等。这方面我很久没折腾了，所以不敢瞎说。

4. 如果您是想成为一名iOS Hacker的话，最近有本书挺火的: amazon.com/iOS-Hackers-... 我没空看不知道咋样，但作者很神。另外现在iOS越狱界也有了自己的大会，可以看看“越狱梦之队”的演讲和文档: absinthejailbreak.com/d... 。如果您还是没有满足的话，可以看看从硬件入手的逆向工程和调试，分享一个我收藏的宝贝: wenku.baidu.com/view/da...

5. 另外说iOS代码是封闭/闭源的其实不全对，苹果算是开源界的一面大旗了，比如WebKit。iOS的组成部分也一样是开源的，可以在官网 opensource.apple.com/ 看到，最新的iOS 5.1.1在这: opensource.apple.com/re... 。但是如您所见，这里并没有iOS操作系统的代码，而是一些库和编译器、调试器...其中JavaScriptCore和WebCore很有用，这两者是WebKit的基础，可以说WebKit是iOS最重要的组成之一，截止iOS 5 (6我还没下呢=___=)，所有多于一行文字的控件其实都是WebKit标准的(不可思议吧?!)。很多iOS的Hack都是从这里开始的。说到WebKit,之前Comex大神的Spirit越狱(那个"Slide to Jailbreak")就是利用Safari->WebKit->PDF Engine->TIFF字体的漏洞实现了代码注入！所以每一个系统组件都可能是iOS逆向/Hack的突破口!

水平有限，如有错误和遗漏还请各位纠正、补充 ;-)

编辑于 2013-01-30 18 条评论 感谢 分享 收藏 • 没有帮助 • 举报 • 作者保留权利

360

58 人赞同

hangcom， iOS App/越狱/逆向

58

如果只是针对ios app，应该是很容易的，class-dump，没有意外的话，直接出全部的.h文件，而且非常清晰。常见的UI结构，基础的数据对象，有了这些，即使看不到.m中的实现代码，是不是也能看个大概了?
至于ios系统本身，看private api。
再往下，那就是大牛们的事了

相关问题 换一换

香港的 iOS 开发是怎样一番景象? 5 个回答

什么水平的开发者应该参加 iOS 7 Tech Talk，会有哪方面的帮助，与会的体验如何，有什么感想? 11 个回答

如何评价 Swift 语言? 109 个回答

为什么很多 iOS 软件的启动页需要等好几秒? 而有些就不用? 20 个回答

如何评价 React Native? 62 个回答

最近对陌陌做了逆向，获取Location坐标并在陌陌里加了一个地图页，这应该就属于标准的App逆向，微博的技术贴里提到了相应的技术点，行内人看了应该就会明白，但全部的技术内容暂时应该不会整理出来，毕竟也要尊重一下app的原作者。

时隔一年半，我把这些iOS逆向知识整理出版了一本入门级书《iOS应用逆向工程》，有兴趣可以看看，顺便了解一下#iOS逆向工程七种武器#：）

编辑于 2014-02-19　10 条评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

ZY Alex， Hey buddy

104

104 人赞同

本人逆向经验丰富，对ios和mac osx底层有过深入挖掘. mac和ios有互相借鉴的地方，所以以下提到的信息可能适用于ios或者mac。

- 0x0.Background: 你必须要有很强的逆向sense，这个是逆向分析的基础，逆向的sense举个栗子：如果你发现看到一个产品之后能够大致猜出它的架构，它的关键部分，核心算法以及可能存在的bug，甚至能够猜出影响性能的是哪部分，这个需要很长时间的逆向分析和工程开发的经验。BTW:语言什么的就不说了，ARM,X86指令,Objective-C,C,C++
- 0x1.Tools: 你需要掌握以下工具:otool,lipo,ar,libtool,class-dump,mach-o-view(有空读一读它的代码，自己编译以下，加点功能什么的),hopper disassemble,ida pro,gdb,xcode开发要会的就不提了，还有一个很有用的codeunsign,最后再推荐一个我写的一个magic类:cccssw/call_at_anywhere GitHub
  这个类用途很广,发挥余地很多.
- 0x2.Frameworks: 调用私有API什么的是最简单的部分，最直接的路径是去private frameworks下面根据frameworks的名称猜测各个framework是干什么的，然后用class-dump dump出header，在项目里面引用就可以用了.如果观察力到位，发现某些官方app有某些功能能够猜测出背后可能有调用私有api，反汇编这个官方程序就能找到私有api的调用形式.
- 0x3.Kernel: ios 和macos都是bsd+mach-o的混合模型，网上有一个图很清楚，说明其架构的. mach-o 的格式学习的最佳途径就是看mach-oview这个开源项目的代码。
- 0x4.Defend technics used by Apple：用得最普遍得就是利用xpcservice将调用放在另外一个可执行程序中，然后函数调用通过进程间通信完成，核心得逻辑不会在这个xpc调用里面，该xpc远程程序会

继续调用底层frameworks,最后你就很难找到最核心的逻辑和算法到底在哪里了；关键部分用c实现，然后隐藏data structure，只留出必要的指针，hidden pointer的技术，keychain的实现就是这样做的，要逆向出它keychain的数据库格式非常难，尝试过，失败了；另外一方面可能考虑到代码的可维护性，大部分苹果的代码都有很详细的log，可能有开关控制log的打开关闭，如果能把log开起来，一个程序，framework就很容易跟踪了(静态分析),打开这些log一般要直接修改二进制文件，或者修改特殊的plist文件.

- 0x5.static analysis:静态分析一般就是先定位最关键的地方，定位的方法很多，一般先开log，然后分析完log后通过关键字来找。定位成功后，没有什么底层技术时，你想要干的事情基本上就快完成了，逻辑就在你面前，汇编配合伪代码就很简单了(除了有FSM或者jmp table的情况,这种情况还是动态分析吧). 有些涉及到底层技术，有一种内核级别调用的陷入函数(涉及到mach-o内核机制)比较麻烦，这种情况也有一些办法，做起来怎么样都是限制大，这篇文章How I cracked the security foundation of Mac OSX System 提供了一些技巧和思路。
- 0x6.dynamic analysis:动态分析要配合静态来做，主要还是用xcode的调试器或者gdb，xcode的符号断点.f6-f7-f6慢慢调.
- 0x7.reimplement logics:功能重现，逆向里面经常要做的事情，一个加密算法，解密算法要重新实现一遍，让自己可以用。途径有两条：1.照着汇编写出汇编版，c版或者objective－c的实现；2.直接调用二进制中的函数. 途径1是考你功底的，功底深做起来就是个体力活而已。途径2有两条路，一是修改二进制文件，或者patch你要的逻辑到一个有架子了的二进制文件，二是计算函数地址动态加载。 重点讲一下途径2，静态修改很麻烦，直接暴露需要解决以下问题：mach-o文件中有两个section与export function有关，其中一个Symbol table和与之相关的String table较容易修改，另外一个是Dynamic loader info比较难修改，里面有的相对位置需要做uleb128转换，另外存储信息的格式是链表格式，解决这两个就可以了. 静态修改除了这些，mach-o前面与section对应的头信息也要修改，长度，位置偏移量.; 途径2的动态加载就不多说了，看我github的项目call_it_any_where 应该是目前为止最方便的方式。
- 0x8.jail-development:越狱开发其实门槛比app store还要低.这一块与逆向有关的主要是hook class之类的，老外有一篇很详细的博文讲这块。貌似就在后面的blog list里面.
- 0x9.security issues: 建议手机别越狱，mac和ios下的maleware其实比windows下面还难发现.;keychain里面即使最严格的ACL策略也是能够绕过的;

讲一讲会逆向的好处：逆向是一门艺术也是一种研究方法，能够让你弄清楚程序运行的本来面目。看别人的实现也可以用来提升自己的架构能力。逆向能够找到一些诡异问题的root reason。最重要的是逆向能力强后，对程序，代码，算法，数据结构，计算机体系的认识会深刻很多。做项目也会从容很多，不太再会遇到什么bug搞不定。另外个人认为在漏洞挖掘上比起fuzzy逆向才是正途。

不会逆向的程序员不是好程序员。

最后再给些blog供学习:

- Reverse Engineering Mac OS X
- ChinaAlex
- Reverse Engineering
- rentzsch (Jonathan 'Wolf' Rentzsch) GitHub
- Matt Galloway

(水平一般，如有错误和遗漏还请各位纠正、补充 ;-))

发布于 2014-01-03 14 条评论 感谢 分享 收藏 • 没有帮助 • 举报 • 作者保留权利

suu， λ

7 人赞同

路过推荐几个工具

Mac端：Hopper Disassembler（我喜欢这个多过于ida），otool，ida pro

PC端：ida pro（还用说嘛→_→）其他不晓得了欢迎补充

发布于 2013-08-08 2 条评论 感谢 分享 收藏 • 没有帮助 • 举报 • 作者保留权利

哦啦啦， 呵呵

1 人赞同

跟风解答一下吧，因为最近项目整改所以研究了一下自己一直想接触的这一领域，最简单的应该是你应当使用class-dump进行dump获取头文件，接下来可以使用theos之类的编写tweak软件咯，这样基本可以获取大部分你想要的信息，再进一步配合静态分析软件IDA或者hoopper就可以获取更准确代码，以上无论是app还是framework都适用于基本的逆向分析了

发布于 2013-07-29　1 条评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

1

roysue， bbs.pediy.com/showthrea...

1 人赞同

向您推荐我们的入门教程： ［新人请看］[看雪iOS安全小组]置顶向导集合贴

发布于 2016-09-19　2 条评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

0

知乎用户， 关注细节的布道者

以上的回答者都是血虚无缥缈之辈。 不是看你掌握了多少技术，而是看你用掌握的技术做了什么？ 市场上的红包达人，3k，6k，女娲，秒杀一切，q霸，蝙蝠侠我都破解了一遍，等你们破解一遍在来回答吧

发布于 2016-11-16　添加评论　感谢　分享　收藏 • 没有帮助 • 举报 • 申请转载

0

然然， 你最想对BUG说的一句话是什么？滚远点！

这个问题竟然能邀请到季逸超同学的回答，我也就会写iOS，对逆向工程很感兴趣，准备深入了解一番。

发布于 2016-04-06　添加评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

0

沈寅， 有意换工作的同学请私信,iOS独立开发者,...

要追求底层 并一定需要逆向 ios很多底层代码还是开源的 opensource.apple.com/　如果要了解系统机制 建议先从macos入手 毕竟ios是从那儿改良的，并且他的限制也没有那么多。当然个人觉得先从文档开始 了解他的设计思路 也不是坏事。

发布于 2013-10-09　添加评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

0

胡江傲， 挨踢程序猿！

理由完全很不充分，逆向iOS上的app倒还可以理解也相对比较容易，逆向iOS本身，那就追随那些释放越狱的牛人去吧！

FYI：

theiphonewiki.com/wiki/...

编辑于 2012-06-27　添加评论　感谢　分享　收藏 • 没有帮助 • 举报 • 作者保留权利

我来回答这个问题

写回答...

我要回答