稀土掘金 (十七) : 教你反编译别人的app

虽然iOS系统相比于其他手机操作系统相对安全，但是这个安全并不是绝对的，我一直相信，道高一尺魔高一丈。此文想以实际例子出发，告诉大家，如何去反编译一个app，并且从某个角度来说，iOS没有传说中的“安全”。

**这里只最简单的结论给大家，具体原理以及工具的详细使用，请各位自行Google，为什么这么做呢，因为我想用最简单的语言，让没有汇编基础的同学也能明白。**

**将用以下几个方面来阐述我的观点：**

* iOS app可以“有条件”地被反编译
* 脱掉App Store给“二级制文件”穿上的外衣
* 如何利用工具去反编译
* 在感兴趣的地方生成伪代码
* 如何预防反编译

**iOS app可以“有条件”地被反编译**

首先，告诉大家一个不幸的消息，上架至App Store的app，被apple加密了。所以标题才是“有条件”才能进行反编译，令人欣喜的是，在debug下产生的ipa或者是release下的ipa包或者是渠道包（各种应用市场能下的ipa包）都能直接反编译。

**脱掉App Store给“二级制文件”穿上的外衣**

当然，引言已经说了，道高一尺魔高一丈，就算是被apple加密的ipa，其实也是可以反编译的，只不过相对麻烦罢了。 来来来，给你们几个工具，就可以解掉apple的加密了。

* clutch
* dumpdecrypted
* gdb
* AppCrackr

由于AppCrackr被很多开发者吐槽，这个傻瓜式的解密会严重导致盗版泛滥，所以这个工具已经很不好用了。 这里就不展示demo了，感觉如果把App Store上的东西反编译出来发到网上去，我感觉我的程序员生涯可能就要到头了。

**如何利用工具去反编译**

这里会介绍两个工具**class-dump**和**Hopper Disassembler**。 首先，先撸一个最简单的app来做小白鼠。[下载地址（包含二进制文件&dump结果）](https://github.com/kuailejim/demo) 看图，就是这样，我只改了ViewController这个类。

由代码可以看出，我就写了两个方法testClassDump和testHideClassDump，后者没有声明在.h中（ps：我想试验这样能不能被反编译到）。

**ok，前戏都做完了，可以开始干活了。**

我们run一下工程，然后打开Products文件夹下的DecompilingTest.app所在目录，显示包内容，拿到**二进制文件**。

然后我是复制到桌面，然后执行下述命令，即可拿到工程中的.h文件。

OK，得到下述结果，我们看看拿到的ViewController.h里面，能拿到什么方法

**事实证明，没有声明的方法也被dump出来了。**

下一步我们要做的就是看看能不能拿到这两个方法的具体实现了，接下来我们用**Hopper Disassembler**来试试。

**Hopper Disassembler**的用法很简单，只要将二进制文件拖进去就行了。看看拖进去之后的结果。

此时心中一万头草泥马飞奔而过，what’s the fk！！一堆汇编语言宝宝看不懂啊。（当然那个ret应该是return的意思我猜）

**在感兴趣的地方生成伪代码**

Don’t worry！点右上角的**if(b)f(x);**按钮，我们能看到这个方法的伪代码，大部分的时候我们能从伪代码中看出我们需要的信息。

结果如下：

至此，两个方法都被反编译出来了！！

**这里需要注意，就算方法没有声明在.h中，也能被dump，之后就能被反编译了。**

**如何预防反编译**

说了这么多，我们预防呢，是不是需要像Java的一样加上各种复杂的**混淆**呢。

其实我觉得大可不必，本身反编译成本就很大，代码这么多，一个个反编译过来是在蛋疼，就算有伪代码也需要理解，而且有些代码就算有伪代码也很难理解。

**只要做好核心代码，做好混淆就行了，比如涉及到密码，核心算法。**

**总结：**

**没有绝对安全的系统，也没有黑客破不掉的系统，所有需要的只是时间而已。**