

从编译文件和源码角度分析微信小程序

1. 官方源码分析编译后 js 群文件结构

(1)官方源码快速创建的工程目录结构，如图 1：其中包含可执行 js 为，app.js、index/index.js、logs/logs.js、utils/sutils.js；根目录下的 app.js 是整个程序的入口，app.json 配置整个程序的信息

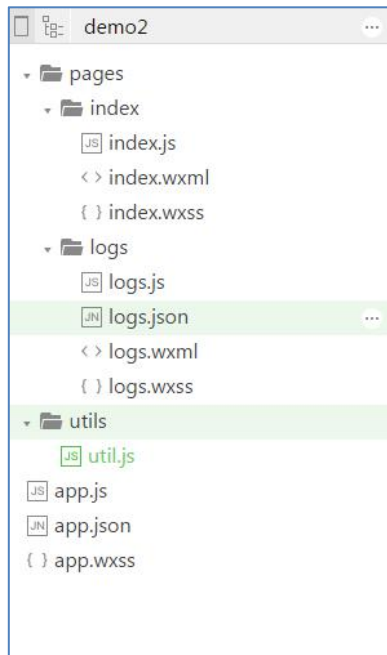


图 1 工程结构

(2)微信开发工具中，调试功能中，显示的编译后的源码文件目录结构，其中包含可执行的 js：

pages/index/index.js, pages/logs/logs.js, utils/utlis.js, app.js,

新增文件：appservice, WAService.js, asdebug.js；其中 WAService 还有有很多地方依赖 window 对象，所以它在微信中和开发者工具内一样，依然运行于 webview 组件之内。如下所示：

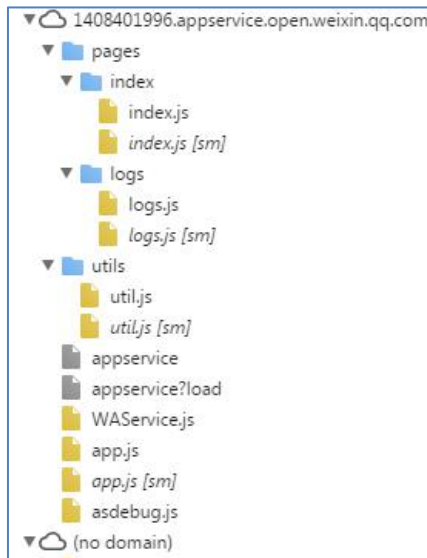


图 2 编译器编译后的文件

2. 发布后的小程序分析

通过微信小程序入口，搜索到相关的小程序名称，初次会从微信服务器 servicewechat.com 下载到该发布的小程序。如图 3

下载的小程序的文件类型为 data, 并非一个 zip 压缩文件，如图 4 所示。由此，微信在加载该小程序时的过程为：缓存中解析出该文件的入口 app.js 和 app.json 以及程序的 entryPagePath, 再由 webView 加载以及执行程序。



图 3 网络下载小程序

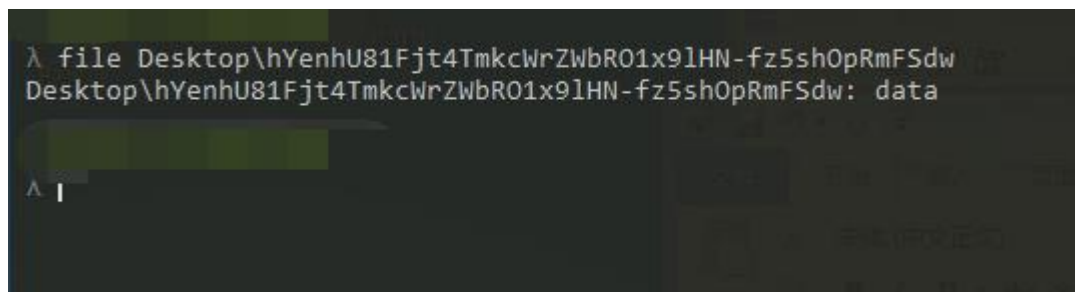


图 4 小程序文件类型

(3) 下载的文件 HEX 数据观察分析

通过对附件 2 和附件 3 中 demo 和微信小程序，进行 16 进制的初步分析，得出小程序的构造，如下图 5 所示。

00000000	BE 00 00 00 00 00 00 00 9B	00 00 5B 6F ED 00 00	?.....?.lo?.e.j
00000010	00 05 00 00 00 10 2F 61 70 70 2D 63 6F 6E 66 69	/app-confi
00000020	67 2E 6A 73 6F 6E 00 00 00 A9 00 00 01 52 00 00		g.json...Q...R..
00000030	00 0F 2F 61 70 70 2D 73 65 72 76 69 63 65 2E 6A		../app-service.j
00000040	73 00 00 01 FB 00 00 0F 3A 00 00 00 10 2F 70 61		s...?.../paa
00000050	67 65 2D 66 72 61 6D 65 2E 68 74 6D 6C 00 00 11		ge-frame.html...
00000060	35 00 00 46 CA 00 00 00 17 2F 70 61 67 65 73 2F		5..F?.../pages/
00000070	69 6E 64 65 78 2F 69 6E 64 65 78 2E 68 74 6D 6C		index/index.html
00000080	00 00 57 FF 00 00 02 60 00 00 00 15 2F 70 61 67		..W ...'..../pa
00000090	65 73 2F 6C 6F 67 73 2F 6C 6F 67 73 2E 68 74 6D		es/logs/logs.htm
000000A0	6C 00 00 5A 5F 00 00 01 B9 7B 22 70 61 67 65 22		l..Z_... "page"
000000B0	3A 7B 22 70 61 67 65 73 2F 6C 6F 67 73 2F 6C 6F		:{"pages/logs/lo
000000C0	67 73 2E 68 74 6D 6C 22 3A 7B 22 77 69 6E 64 6F		gs.html":{"windo

BE 00 00 00 00 00 00 00 9B : 固定值?
00 00 5B ED: value=filesize-first_file_off
ED 00 00: ?似乎是固定值
00 05:文件个数
00 00 00 10 : 字符个数 16
2F 61 70 70 2D 63 6F 6E 66 69 67 2E 6A 73 6F 6E: app-config.json
00 00 00 A9: app-config.json文件偏移起始地址
00 00 01 52:app-config文件大小

图 5 微信小程序的结构

依照图 5 中的结构,可以提取到小程序中的图片/images/ad_area.png,也可得到 ad_area.png 路径为 images/目录下。

附件 6 app-service.js 即为提取到的 js 文件,其中包含了与程序相关相关的所有的.js 文件代码。

由此:可以依次提取到小程序开发的所有文件。如果研究某个小程序的代码逻辑,那完全可以将小程序的代码进行提取以及逆向。

结论:

小程序在微信中的运行,受到微信的保护机制,运行在微信的 webview 中,较为安全;但是明显的,微信的小程序包含了所有的执行 js 以及相关组件,其代码逻辑结构遭到泄露,难以采取有效的保护措施。加强小程序保护需要对 js 代码混淆。