《iOS应用逆向工程》学习笔记（六）使用dumpdecrypted砸壳

时间 2014-08-03 21:44:46 CSDN博客
原文 http://blog.csdn.net/jymn_chen/article/details/38361161
主题 iOS Xcode 程序员

本来是打算用AppCrackr砸壳的，结果砸壳都是失败的，开始以为是App的加密太厉害了，后来才知道是因为AppCrackr太暴力了，引起公愤，结果被人投诉招致核心功能被迫关闭了。

幸好在RE官网搜到一个 用dumpdecrypted砸壳 的帖子。下面是我砸壳的经历。

一、造锤

1.下载dumpdecrypted源码

下载地址： https://github.com/stefanesser/dumpdecrypted/archive/master.zip ，接着在Mac中解压。

2.确认iOS设备的版本

iOS 7.1.x，原帖中snakeninny略啰嗦。。。

3.Makefile

cd到dumpdecrypted目录，看看Makefile文件的内容：

```
CC_BIN=`xcrun --sdk iphoneos --find gcc`
GCC_UNIVERSAL=$(GCC_BASE) -arch armv7 -arch armv7s -arch arm64
SDK=`xcrun --sdk iphoneos --show-sdk-path`

CFLAGS =
GCC_BASE = $(GCC_BIN) -Os $(CFLAGS) -Wimplicit -isysroot $(SDK) -F$(SDK)/System$

all: dumpdecrypted.dylib

dumpdecrypted.dylib: dumpdecrypted.o
  $(GCC_UNIVERSAL) -dynamiclib -o $@ $^
```

```
%.o: %.c
  $(GCC_UNIVERSAL) -c -o $@ $<

clean:
  rm -f *.o dumpdecrypted.dylib
```

大多数看不懂。。。

接下来我们需要确认的是 GCC_UNIVERSAL 和 SDK 这两个变量的值和iOS设备的环境保持一致。

4.确保Makefile的配置和真机环境一致

在Mac中打开终端，输入 xcrun --sdk iphoneos --show-sdk-path 命令，查看SDK版本：

```
/Applications/Xcode 5.1.1.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs
```

Xcode的SDK版本是7.1.x，GCC_UNIVERSAL这个变量值可以略过。

5.创建动态库文件

（1）一错

在确保Makefile中对动态库的设置和iOS真机环境一致后，在当前目录下输入：make。

但是失败了，错误信息如下：

```
`xcrun --sdk iphoneos --find gcc` -Os  -Wimplicit -isysroot `xcrun --sdk iphoneos --show-sd
/bin/sh: /Applications/Xcode: No such file or directory
make: *** [dumpdecrypted.o] Error 127
```

原因是找不到/Applications/Xcode来执行其中的一些脚本。 好吧，我的Mac中有3个Xcode：/Applications/Xcode 5.0.2, /Applications/Xcode 5.1.1, /Applications/Xcode 6 Beta4，就是没有/Applications/Xcode。

没事，将Xcode 5.1.1重命名为Xcode就行了：

```
$ sudo mv Xcode\ 5.1.1.app/ Xcode.app/
```

（2）再错

再make，还是报错，错误信息和上面一样。

不怕，我们还有xcode-select这个小伙伴，通常Xcode找不到之类的错误都应该找它帮忙：

```
$ xcode-select -p
/Applications/Xcode 5.1.1.app/Contents/Developer
```

原来xcrun查找cmd tool时的路径还是Xcode 5.1.1/，当然什么都找不到了。这时候将它重置就行了（默认是/Applications/Xcode.app/）：

```
$ sudo xcode-select -r
$ xcode-select -p
/Applications/Xcode.app/Contents/Developer
```

（3）成功

再make，成功，输出如下：

```
$ make
`xcrun --sdk iphoneos --find gcc` -Os  -Wimplicit -isysroot `xcrun --sdk iphoneos --show-sd
`xcrun --sdk iphoneos --find gcc` -Os  -Wimplicit -isysroot `xcrun --sdk iphoneos --show-sd

$ ls
Makefile          dumpdecrypted.c          dumpdecrypted.o
README               dumpdecrypted.dylib
```

可以看到目录中多了两个文件，其中dylib后缀的就是我们要创建的动态库文件，也就是用来砸壳的锤子。

二、砸壳

1.将“锤子”放入设备中

查看iOS设备的IP地址，然后在Mac上使用scp命令将 dumpdecrypted.dylib 文件放到iOS设备中：

```
$ scp dumpdecrypted.dylib root@192.168.xxx.xxx:/var/tmp
root@192.168.xxx.xxx's password:
dumpdecrypted.dylib                           100%   81KB  81.0KB/s   00:00
```

2.砸

选定一个让你觉得非常不爽或非常感兴趣的app，我就随便选了一个HBGC。在iOS设备上打开iFile，查到它的可执行文件的路径为：/var/mobile/Applications/EBBD26E9-DDBA-481E-9403-84D159436889/HBGC.app/HBGC

然后用SSH连到iOS设备上，cd到刚刚动态库的路径：/var/tmp。

```
$ ssh root@192.168.xxx.xxx
root@192.168.xxx.xxx's password:
root# cd /var/tmp/
root# ls
FlipswitchCache/                              com.apple.audio.hogmode.plist
L65ancd.sock=                                 com.apple.tccd/
L65d.sock=                                    com.apple.timed.plist
MediaCache/                                   cydia.log
RestoreFromBackupLock*                        dumpdecrypted.dylib*
SpringBoard_reboot_flag                       launchd/
com.apple.assistant.bundleservicecache.plist  mobile_assertion_agent.log
```

砸壳（久等了）：

```
root# DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib /var/mobile/Applications/EBBD26E9-DDBA-481E
mach-o decryption dumper

DISCLAIMER: This tool is only meant for security research purposes, not for application cra

[+] detected 32bit ARM binary in memory.
[+] offset to cryptid found: @0xd5a90(from 0xd5000) = a90
[+] Found encrypted data at address 00004000 of length 3047424 bytes - type 1.
[+] Opening /private/var/mobile/Applications/EBBD26E9-DDBA-481E-9403-84D159436889/HBGC.app/
[+] Reading header
[+] Detecting header type
[+] Executable is a FAT image - searching for right architecture
[+] Correct arch is at offset 16384 in the file
[+] Opening HBGC.decrypted for writing.
[+] Copying the not encrypted start of the file
[+] Dumping the decrypted data into the file
[+] Copying the not encrypted remainder of the file
```

```
[+] Setting the LC_ENCRYPTION_INFO->cryptid to 0 at offset 4a90
[+] Closing original file
[+] Closing dump file
```

成果：

```
root# ls
FlipswitchCache/                                com.apple.audio.hogmode.plist
HBGC.decrypted                                  com.apple.tccd/
L65ancd.sock=                                   com.apple.timed.plist
L65d.sock=                                      cydia.log
MediaCache/                                     dumpdecrypted.dylib*
RestoreFromBackupLock*                          launchd/
SpringBoard_reboot_flag                         mobile_assertion_agent.log
com.apple.assistant.bundleservicecache.plist
```

其中的HBGC.decrypted就是目标产物，接下来IDA各种斧头水果刀上吧。

三、附录

1.xcrun

首先简单看看xcrun的帮助信息：

```
$ xcrun -h
Usage: xcrun [options] <tool name> ... arguments ...

Find and execute the named command line tool from the active developer
directory.

The active developer directory can be set using `xcode-select`, or via the
DEVELOPER_DIR environment variable. See the xcrun and xcode-select manual
pages for more information.

Options:
  -h, --help                  show this help message and exit
  --version                   show the xcrun version
  -v, --verbose               show verbose logging output
  --sdk <sdk name>            find the tool for the given SDK name
  --toolchain <name>          find the tool for the given toolchain
  -l, --log                   show commands to be executed (with --run)
  -f, --find                  only find and print the tool path
  -r, --run                   find and execute the tool (the default behavior)
  -n, --no-cache              do not use the lookup cache
  -k, --kill-cache            invalidate all existing cache entries
  --show-sdk-path             show selected SDK install path
  --show-sdk-version          show selected SDK version
  --show-sdk-platform-path    show selected SDK platform path
  --show-sdk-platform-version show selected SDK platform version
```

xcrun的作用在于从一个激活的开发者目录（active developer directory）中查找一个command line tool，并执行这个工具。

例如上面的Makefile中： GCC_BIN=`xcrun --sdk iphoneos --find gcc`

分解来看：

（1）xcrun --find gcc

```
$ xcrun --find gcc
/Applications/Xcode 5.1.1.app/Contents/Developer/usr/bin/gcc
```

这一步获取了gcc这个tool的路径，设为cmd_tool_path。

（2）xcrun --sdk iphoneos cmd_tool_path

这一步通过路径名获取到了具体的工具程序，这个工具对应iphoneos的SDK，并执行该工具。

（3）GCC_BIN是一条shell命令，对应的就是这个查找和执行工具的过程。

再如： xcrun --sdk iphoneos --show-sdk-path

它的作用就是查找对应于iphoneos SDK的SDK并执行。

```
$ xcrun --show-sdk-path
/Applications/Xcode 5.1.1.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/M

$ xcrun --sdk iphoneos --show-sdk-path
/Applications/Xcode 5.1.1.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs
```

2.xcode-select

首先看看简单的帮助信息：

```
$ xcode-select -h
Usage: xcode-select [options]

Print or change the path to the active developer directory. This directory
controls which tools are used for the Xcode command line tools (for example,
xcodebuild) as well as the BSD development commands (such as cc and make).

Options:
  -h, --help                  print this help message and exit
  -p, --print-path            print the path of the active developer directory
  -s <path>, --switch <path>  set the path for the active developer directory
  -v, --version               print the xcode-select version
  -r, --reset                 reset to the default command line tools path
```

它的作用在于打印或改变active developer directory，而 xcrun**就是从这个**directory**中查找对应的工具**。 通常它的值为：

```
/Applications/Xcode 5.1.1.app/Contents/Developer
```

例如，在/Applications/Xcode 5.1.1.app/Contents/Developer/usr/bin中，可以看到一些上文需要的gcc：

```
$ ls
BuildStrings          gcc               ndisasm
CpMac               gcov-4.2          opendiff
DeRez               git               projectInfo
GetFileInfo          git-cvsserver          resolveLinks
ImageUnitAnalyzer     git-receive-pack     scntool
MergePef          git-shell          sdef
MvMac               git-upload-archive     sdp
ResMerger          git-upload-pack          svn
Rez               gnumake               svnadmin
RezDet              hdxml2manxml          svndumpfilter
RezWack              headerdoc2html          svnlook
SetFile              ibtool               svnrdump
SplitForks          ibtool3               svnserve
TextureAtlas          ibtoold               svnsync
UnRezWack          ictool               svnversion
actool              instruments          symbols
agvtool              iprofiler          xcodebuild
amlint              ld               xcrun
```

以上只是部分输出。

注：以上是我个人在自己的机子上的砸壳经历，大家要根据自己的实际情况进行，详细请参考： 用dumpdecrypted砸壳

热门文章

- 1. jinfo：Java 进程运行时修改虚拟机参数的利器
- 2. PAPI性能测试工具的安装、使用及实例
- 3. 我忠诚的猎兔犬会在危险靠近时警告我，对，他的名字叫BBG
- 4. Docker 1.7.1 正式发布下载
- 5. linux线程间同步方式汇总
- 6. Samba 4.2.3 发布，最新稳定版本

分享本文

用户登陆

邮箱

密码

登　陆

收藏到推刊　　创建推刊

收 藏　取消

已收藏到推刊!

推刊名(必填)　请填写推刊名

推刊描述

描述不能大于100个字符!

权限设置：　公开　仅自己可见

创建　取消

网站相关

- 关于我们
- 移动应用
- 建议反馈

关注我们

- 推酷网
- tuicool2012
- QQ群:164644910

友情链接

人人都是产品经理 TMTForum 魔部网 PM256 品途网 移动信息化 行晓网 Code4App 智城外包网 LAMP人 安卓航班网 虎嗅 缘创派 IT耳朵 艾瑞网 创媒工场 雷锋网 经理人分享 市场部网 砍柴网 CocoaChina 北风网 云智慧 我赢职场 大数据时代 奇笛网 咕噜网 红联linux Win10之家 鸟哥笔记 爱游戏 投资潮 31会议网 极光推送 Teambition Cocos引擎中文官网 硅谷网 更多链接>>