For an international communication atmosphere, English please! 收不到激活邮件的朋友看这里

Dumpdecrypted 砸壳失败

The Book

iceiPhone Apr 2

Continuing the discussion from 用dumpdecrypted给App砸壳 6 :

*** 以下部分内容摘自《iOS应用逆向工程》第二版，以iOS 8为环境编写，应该也支持iOS 7，请大家注意。 ***

在《iOS应用逆向工程》4.6.2节中，我们曾推荐使用iPhoneCake源的AppCrackr 1.7版给App砸壳。这种方式简单粗暴，省时省力，但正是因为它过于方便有木有，导致几乎所有iDevice用户都可轻松上手，随便亵玩，所以不少用户都拿它来破解程序，而不是学习《iOS应用逆向工程》，简直可以说是婶可忍叔不可忍！这也导致了iOS越狱开发社区普遍认为这个软件助长了盗版的气焰，没有脱离低级趣味，对iPhoneCake源进行了强烈谴责，责令其限期整改。迫于压力，iPhoneCake在前段时间将AppCrackr下架，而书中提到的xsellize源中的AppCrackr则是1.5旧版，已不能在高级系统中使用。所以，为了响应业界反盗版的呼声，提倡毛主席“自己动手丰衣足食”的革命精神，让“砸壳”这件事恢复单纯的研究目的，在这里我们会使用更偏geek一些的dumpdecrypted方式来给App砸壳，不再推荐AppCrackr、Clutch、Crackulous等纯UI方式。由于dumpdecrypted刚经历过一次大升级，目前网上可以找到的使用教程均已过期，所以这里我们以一个虚构的TargetApp.app为例，手把手带大家进行一次完整的“砸壳 + class-dump”，请大家准备板凳瓜子汽水，开始围观。如果能对着电脑，边看边做，善莫大焉！楼猪才疏学浅，如有纰漏，敬请斧正，洗耳恭听，污言秽语，免开尊口，感谢支持！

1. 下载dumpdecrypted的源码
源码下载地址是“https://github.com/stefanesser/dumpdecrypted/archive/master.zip”，下载后请将其解压至你习惯的位置，例如楼猪为“/Users/snakeninny/Code/”，解压后生成“/Users/snakeninny/Code/dumpdecrypted-master/”。

2. 编译源码

```
snakeninnysiMac:~ snakeninny$ cd
/Users/snakeninny/Code/dumpdecrypted-master/
snakeninnysiMac:dumpdecrypted snakeninny$ make
`xcrun --sdk iphoneos --find gcc` -Os -Wimplicit -isysroot `xcrun
--sdk iphoneos --show-sdk-path` -F`xcrun --sdk iphoneos --show-
sdk-path`/System/Library/Frameworks -F`xcrun --sdk iphoneos --
show-sdk-path`/System/Library/PrivateFrameworks -arch armv7 -arch
armv7s -arch arm64 -c -o dumpdecrypted.o dumpdecrypted.c
```

```
`xcrun --sdk iphoneos --find gcc` -Os -Wimplicit -isysroot `xcrun
--sdk iphoneos --show-sdk-path` -F`xcrun --sdk iphoneos --show-
sdk-path`/System/Library/Frameworks -F`xcrun --sdk iphoneos --
show-sdk-path`/System/Library/PrivateFrameworks -arch armv7 -arch
armv7s -arch arm64 -dynamiclib -o dumpdecrypted.dylib
dumpdecrypted.o
```

上面的make命令执行完毕后，会在当前目录下生成一个dumpdecrypted.dylib文件，这就是我们等下砸壳所要用到的榔头。此文件生成一次即可，以后可以重复使用，下次砸壳时无须再重新编译。

3. 用ps命令定位待砸壳的可执行文件
在iOS 8中，来自AppStore的App全部位于/var/m 1 / 14 中可执行文件位于/var/mobile/Containers/Bundle/Application/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/Target.app/下。我们不知道X是什么，肉眼定位需要手工遍历所有目录，劳民伤财，但一个简单的小技巧就可以省时省力：首先在iOS中关掉所有App，然后打开Target，接着ssh到iOS上，打印出所有进程：

```
snakeninnysiMac:~ snakeninny$ ssh root@yourIP
FunMaker-5:~ root# ps -e
PID TTY TIME CMD
1 ?? 3:28.32 /sbin/launchd
……
5717 ?? 0:00.21
/System/Library/PrivateFrameworks/MediaServices.framework/Support/
mediaartworkd
5905 ?? 0:00.20 sshd: root@ttys000
5909 ?? 0:01.86
/var/mobile/Containers/Bundle/Application/03B61840-2349-4559-B28E-
0E2C6541F879/TargetApp.app/TargetApp
5911 ?? 0:00.07
/System/Library/Frameworks/UIKit.framework/Support/pasteboardd
5907 ttys000 0:00.03 -sh
5913 ttys000 0:00.01 ps -e
```

因为iOS上只打开了一个StoreApp，所以唯一的那个含有“/var/mobile/Containers/Bundle/Application/”字样的结果就是TargetApp可执行文件的全路径。

4. 用Cycript找出TargetApp的Documents目录路径。
App的Documents目录位于/var/mobile/Containers/Data/Application/ YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY/下，Y与之前的X值不同，而且这次PS也帮不上忙了。因此，需要借助强大的Cycript，让App告诉我们Documents的路径。命令如下：

`

```
FunMaker-5:~ root# cycript -p TargetApp
cy# [NSFileManager defaultManager] URLsForDirectory:NSDocumentDirectory
inDomains:NSUserDomainMask][0]

#"file:///var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-
2146128611EE/Documents/"
```

`

5. 将dumpdecrypted.dylib拷贝到Documents目录下。拷贝命令如下：

```
snakeninnysiMac:~ snakeninny$ scp
/Users/snakeninny/Code/dumpdecrypted/dumpdecrypted.dylib
root@192.168.2.2:/var/mobile/Containers/Data/Application/D41C4343-
63AA-4BFF-904B-2146128611EE/Documents/
dumpdecrypted.dylib 100% 193KB 192.9KB/s 00:00
```

这里采用的是scp方式，也可以使用iFunBox等工具来操作。

6. 开始砸壳
dumpdecrypted.dylib的用法是：

```
DYLD_INSERT_LIBRARIES=/path/to/dumpdecrypted.dylib
/path/to/executable
```

实际操作起来就是：

`
FunMaker-5:~ root# cd /var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-2146128611EE/Documents/
FunMaker-5:/var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-2146128611EE/Documents root# DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib /var/mobile/Containers/Bundle/Application/03B61840-2349-4559-B28E-0E2C6541F879/TargetApp.app/TargetApp
mach-o decryption dumper

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

+] detected 32bit ARM binary in memory.
+] offset to cryptid found: @0x81a78(from 0x81000) = a78
+] Found encrypted data at address 00004000 of length 6569984 bytes - type 1.
+] Opening /private/var/mobile/Containers/Bundle/Application/03B61840-2349-4559-B28E-0E2C6541F879/TargetApp.app/TargetApp for reading.
+] Reading header
+] Detecting header type
+] Executable is a plain MACH-O image
+] Opening TargetApp.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 a78
+] Closing original file
+] Closing dump file
`

当前目录下会生成TargetApp.decrypted，即砸壳后的文件：

```
FunMaker-5:/var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-2146128611EE/Documents root# ls
TargetApp.decrypted dumpdecrypted.dylib OtherFiles
```

赶紧把砸壳后的文件拷贝到OSX上备用吧，class-dump、IDA等工具已经迫不及待啦。
以上6步还算简洁明了，但可能会有朋友问，为什么要把dumpdecrypted.dylib拷贝到Documents目录下操作?
问得好。我们都知道，StoreApp对沙盒以外的绝大多数目录没有写权限。dumpdecrypted.dylib要写一个decrypted文件，但它是运行在StoreApp中的，与StoreApp的权限相同，那么它的写操作就必须发生在StoreApp拥有写权限的路径下才能成功。StoreApp一定是能写入其Documents目录的，因此我们在Documents目录下使用dumpdecrypted.dylib时，保证它能在当前目录下写一个decrypted文件，这就是把dumpdecrypted.dylib拷贝到Documents目录下操作的原因。
最后来看看如果不放在Documents目录下，可能会出现什么问题:

```
FunMaker-5: /var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-2146128611EE/Documents root# mv dumpdecrypted.dylib /var/tmp/
FunMaker-5: /var/mobile/Containers/Data/Application/D41C4343-63AA-4BFF-904B-2146128611EE/Documents root# cd /var/tmp
FunMaker-5:/var/tmp root# DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib /private/var/mobile/Containers/Bundle/Application/03B61840-2349-4559-B28E-0E2C6541F879/TargetApp.app/TargetApp
dyld: could not load inserted library 'dumpdecrypted.dylib' because no suitable image found. Did find:
dumpdecrypted.dylib: stat() failed with errno=1

Trace/BPT trap: 5
```

这里errno的值是1，即“Operation not permitted”，砸壳失败。

7. class-dump丫
在崭新的21世纪，App的可执行文件一般都是fat binary，也就是说一个二进制文件里包含适合多个CPU架构使用的可执行文件，虽然CPU架构是向下兼容的（也就是说armv64可以执行armv7s的指令，反之则不行），但向下兼容一般会导致一部分性能的牺牲。这样如果一个binary既包含适用于armv7架构的可执行文件，又包含armv7s的，还包含arm64的，就意味着它可以在iPhone 4（armv7），4s（armv7），5（armv7s），5s（arm64）上都发挥100%的性能。虽然除了处女座以外的其他星座用户一般是感受不到这个性能的提升的~但是，机器比处女座还要处女座，它在执行一个fat binary时，会选择最适合自己CPU的那个可执行文件，其他的可执行文件其实是没有得到执行的。因此dumpdecrypted.dylib起作用的只有实际得到执行的那一个可执行文件，举个例子，如果Victim里含有armv7和armv7s这2种架构，而我们的操作机是一台iPhone5/4，那么dumpdecrypted砸掉的是armv7s/armv7那部分的壳，armv7/armv7s部分仍是有壳的。自然地，class-dump的作用对象必须是砸掉壳的binary，所以我们要在class-dump时指定目标，在本例中，就是

```
class-dump --arch armv7s Target.decrypted
```

或

```
class-dump --arch armv7 Target.decrypted
```

> 可以明白吧？没关系，慢一点，多看几遍，理一理，还是有问题的话，在下面留言啦！

1 Reply

created	last reply	13	466	3	1	5
Apr 2	Apr 19	replies	views	users	like	links

iceiPhone Apr 2

> iceiPhone:
>
> cus

最后一步执行出现以下问题，真机是ipad6.1.3越狱设备

DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib "/var/mobile/Applications/2D07B2F4-B543-4488-BF5C-C81B26CF30AE/OPlayer Lite.app/OPlayer Lite"

mach-o decryption dumper

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

[+] detected 32bit ARM binary in memory.
[+] offset to cryptid found: @0x4abc(from 0x4000) = abc
[+] Found encrypted data at address 00004000 of length 23592960 bytes - type 1.
[+] Opening /private/var/mobile/Applications/2D07B2F4-B543-4488-BF5C-C81B26CF30AE/OPlayer Lite.app/OPlayer Lite for reading.
[+] Reading header
[+] Detecting header type
[+] Executable is a plain MACH-O image
dyld: lazy symbol binding failed: Symbol not found: __*strlcpy*chk
Referenced from: dumpdecrypted.dylib
Expected in: /usr/lib/libSystem.B.dylib

dyld: Symbol not found: __*strlcpy*chk
Referenced from: dumpdecrypted.dylib
Expected in: /usr/lib/libSystem.B.dylib

Trace/BPT trap: 5

2 Replies

snakeninny Apr 2

dumpdecrypted是自己编译的吗?

1 Reply

snakeninny Apr 2

> iceiPhone:
>
> dyld: lazy symbol binding failed: Symbol not found: __strlcpychk

根据这个帖子 5 ，貌似是没有装Command Line Tools?

1 Reply

iceiPhone Apr 2

根据教程自己编译的。

iceiPhone Apr 2

本地有装两个xcode就是

1 Reply

snakeninny Apr 2

3.2.2 安装 Theos

1. 安装 Xcode 与 Command Line Tools

一般来说，iOS 开发者都会安装 Xcode，其中附带了 Command Line Tools。如果还没有安装 Xcode，请到 Mac AppStore 免费下载。如果安装了多个 Xcode，需要使用 xcode-select 命令指定一个活动 Xcode，即 Theos 默认使用的 Xcode。假设安装了 3 个 Xcode，并将它们分别命名为 Xcode1.app、Xcode2.app 和 Xcode3.app，若要指定 Xcode3 为活动 Xcode，则运行如下命令：

试试

1 Reply

Chansonyan Apr 19

前辈，我也遇到这个问题了：

```
Cheng-teki-iPhone:~ root# cd /var/mobile/Applications/071B5AF9-
8A0C-4B2C-A24B-B670A24F53DB/Documents

Cheng-teki-iPhone:/var/mobile/Applications/071B5AF9-8A0C-4B2C-
A24B-B670A24F53DB/Documents root#
DYLD_INSERT_LIBRARIES=dumpdecrypted.dylib
/var/mobile/Applications/071B5AF9-8A0C-4B2C-A24B-
B670A24F53DB/MY.app/MY

mach-o decryption dumper

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

[+] detected 32bit ARM binary in memory.
[+] offset to cryptid found: @0x4a78(from 0x4000) = a78
[+] Found encrypted data at address 00004000 of length 7749632
bytes - type 1.
[+] Opening /private/var/mobile/Applications/071B5AF9-8A0C-4B2C-
A24B-B670A24F53DB/MY.app/MY for reading.
[+] Reading header
[+] Detecting header type
[+] Executable is a FAT image - searching for right architecture
[+] Correct arch is at offset 16384 in the file
dyld: lazy symbol binding failed: Symbol not found:
___strlcpy_chk
  Referenced from: dumpdecrypted.dylib
```

我装了两个XCode 6.3的和5.0.2的 xcode－select 的是6.3的那个，新版本的Xcode， Command Line Tools 应该是自带安装的吧 （我又敲过命令sudo xcode-select --install 重新安装CLT）

重新Make，把dumpdecrypted.dylib复制到iOS的Documents目录下，执行砸壳命令，还是同样的输出～～

@iceiPhone 不知道你当初怎么解决这个问题的?

我的手机是IOS 6.1.3 iPhone 4

万分感谢!

1 Reply

snakeninny Apr 19

从这个帖子和这个帖子来看，貌似还是因为没装好Command Line Tools导致的。你装好后重新一下系统，再重新编译dumpdecrypted，砸壳试试看?

1 Reply

Chansonyan Apr 19

重新安装了CTL，重启过了，按照这两个帖子说的，也在.bash_profile文件里加了export PATH=$PATH:/usr/local/git/bin/

重新编译dumpdecrypted，复制进去 咂壳 -> 还是失败了 一样的输出

我在想，和这里说的] 2 有没有关系：

```
Compile:

First adjust the Makefile if you have a different iOS SDK
installed.

And then just: make
```

但是我不知道怎么改这里MakeFile文件的SDK

```
GCC_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/Library/Frameworks -
F$(SDK)/System/Library/PrivateFrameworks

all: dumpdecrypted.dylib

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

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

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

1 Reply

snakeninny Apr 19

> C Chansonyan:
>
> SDK=xcrun --sdk iphoneos --show-sdk-path

这一句就是指定SDK用的，你运行`xcrun --sdk iphoneos --show-sdk-path`看看是什么输出

1 Reply

C

Chansonyan  Apr 19

嗯 真的是编译dumpdecrypted的时候，选择SDK的问题

SDK=xcrun --sdk iphoneos --show-sdk-path的输出结果：

```
Cyan-MBP:dumpdecrypted Cyan$ xcrun --sdk iphoneos --show-sdk-path
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platf
orm/Developer/SDKs/iPhoneOS8.3.sdk
```

我看到github上也有人提问类似问题，热心友人给出了解答 2
我照着他的说明去改了下，咂壳成功啦！哈哈

- 下载Xcode4.6，拷贝出6.1.x的SDK到已安装的目录中；
- dumpdecrypted工程的makefile和c文件 按照热心友人提示的修改下 重新编译出dumpdecrypted.dylib就可以在iOS 6.3的系统上咂壳了

谢谢 @snakeninny 引路！

1 Reply

snakeninny C Apr 19

也就是说，8.3的SDK编译出来的dumpdecrypted不能用在iOS 6上了？

1 Reply

C

Chansonyan  Apr 19

应该是的，

- 我的MakeFile：

  ```
  GCC_BIN=xcrun --sdk iphoneos6.1 --find gcc
  GCC_UNIVERSAL=$(GCC_BASE) -arch armv7 -arch armv7s
  ```

```
SDK=xcrun --sdk iphoneos6.1 --show-sdk-path

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

all: dumpdecrypted.dylib

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

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

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

- dumpdecrypted.c

73行修改一下，去掉 ll lc->cmd == LC_ENCRYPTION_INFO_64

```
-                if (lc->cmd == LC_ENCRYPTION_INFO || lc->cmd ==
LC_ENCRYPTION_INFO_64) {
+                if (lc->cmd == LC_ENCRYPTION_INFO) {
```

- Xcode 6.3下载了iOS 6.1的SDK，放在

```
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform
/Developer/SDKs/iPhoneOS6.1.sdk
```

1 person liked this.

Log In to Reply

Suggested Topics

Topic	Category	Replies	Views	Activity
安装dpkg出现问题，导致无法打包	**The Book**	**3**	125	Apr '14
我又出问题了。 make package install	**The Book**	**1**	21	2d
书本用到的工具-百度云共享	**The Book**	**8**	187	Dec '14

Want to read more? Browse other topics in The Book or view latest topics.

Contact | 微博 | Twitter