iOS Reverse Engineering

A.K.A. Haxing iOS apps

By Osama Gamal

Introduction

- Reverse engineering is taking a finished product and analyzing its functions, structures, or behaviors.

- When we are very interested in a certain software feature while not having the access to the source code, we can try to analyze it by reverse engineering.

- Helpful for better understanding how software works
  — Finding security issues
  — May be used illegally :
    — Software Piracy
    — Stealing local cached data (IP and sensitive stored information)

What can reverse engineering really do ?

- All Objective-C interfaces, all properties, all global variables, even all logics are EXPOSED
- We can do reverse engineering on system APIs to use some private functions, which are not documented
- Many companies that develop apps are not aware of existience of this technique, and think their applications are unbreakable.

A small example of how
I recently used reverse engineering with
an application to let me login..
without being an active user

So there is this app that doesn’t load anything except “**ActivationViewController**”, because obviously I’m not an active user

The app stores a key in NSUserDefaults of “isActive” if true then it just moves on with the following views

Until this moment we are fine.. **but ..**

<u>the API DOES NOT CARE</u>

so simply changing the value of the key from false to true, aaaand.. profit.

And even worse..
After taking a look at the API response, it exposes information like email, number and data the app doesn’t even use

Well i know this is not our topic.. heh.

DEMO

Charles with iPhone App

- Charles
  — HTTP/HTTPS sniffing
  — Repeat/Edit request
  — Breakpoints

What can reverse engineering do with iOS specifically ?

- You can hook into existing methods of the application, implement your own modification.. and you could even access the original return of the method !
- Example follows..

ApplicationController.m

```
-(void)setUserBadge:(NSString *)username isAdmin: (BOOL)isAdmin {
   if( [username isEqualToString:@“Osama”] ){
    isAdmin = true;
   }
}
```

— My modification would look like this

```
%hook ApplicationController
-(void)setUserBadge:(NSString *)username isAdmin: (BOOL)isAdmin {
       // whatever your function does was ignored
      // “Osama” can still be accessed from username ^
 isAdmin = true;
}
%end
```

```
%hook ApplicationController
-(void)setUserBadge:(NSString *)username isAdmin: (BOOL)isAdmin {
      // whatever your function does was ignored
     // “Osama” can still be accessed from username ^
 isAdmin = true;
}
%end
```

So for everytime your application calls setUserBadge method, my %hook will load the modification i used into that method

Jailbreak your iPhone

- You really need to be jailbroken to have a the required tools to work without a problem
- Jailbreak currently supports up to iOS 11.3.1 (and 11.4 is yet to be released)
- Jailbreak breaks signing requirements for applications
- Gives you root access
- SSH access into the phone

Tools used

- Charles
  — HTTP/HTTPS sniffing
  — Repeat/Edit request
  — Breakpoints

- Clutch
  — For decrypting App Store apps (jailbreak only !)

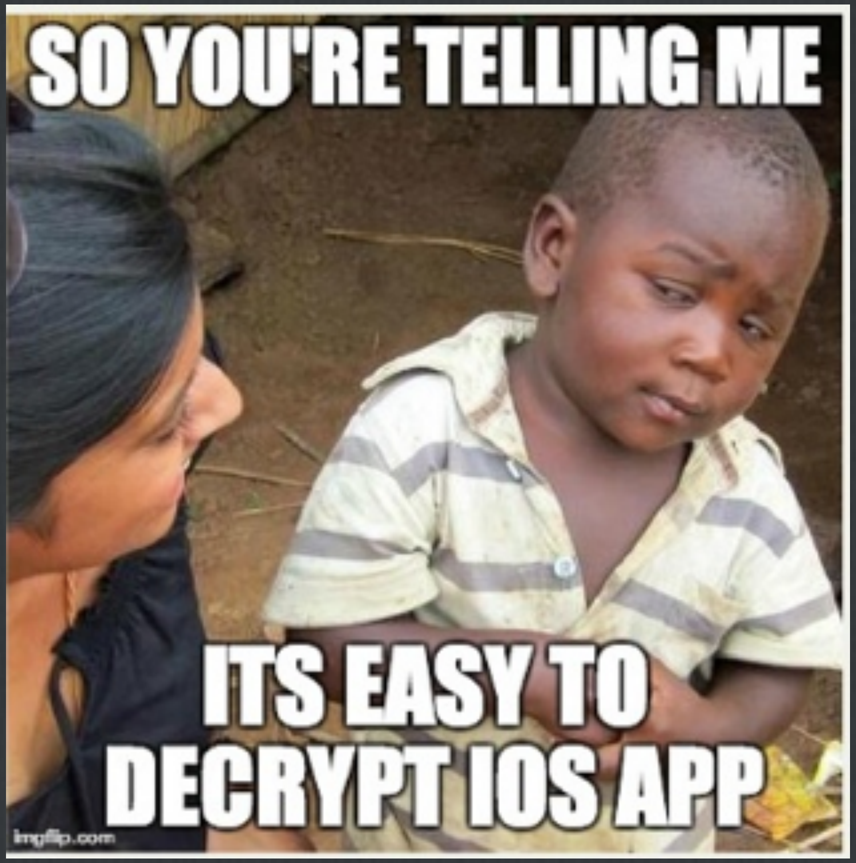
- class-dump
  — for dumping the application header files

- Hopper Disassembler
  — For disassembling decrypted IPA's

- Inspective-C
  — Tool to log Objective-C message hierarchies, watch class or object, its more like showing footsteps of a class

Tools used

- Clutch
  — For decrypting App Store apps (jailbreak only !)



- NO you can’t decrypt apps using online tools or such, only though jailbroken iOS Device

DEMO

class-dump

DEMO

Hopper Disassembler

DEMO

Injecting my modifications, permanently

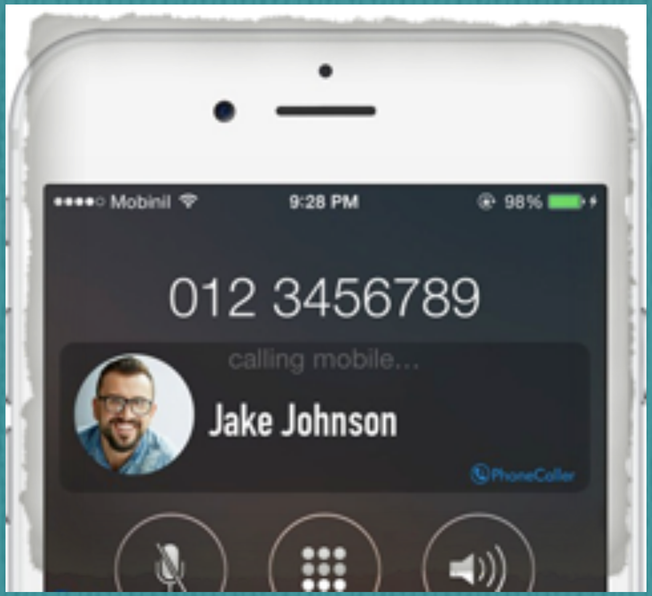
into the app using THEOS

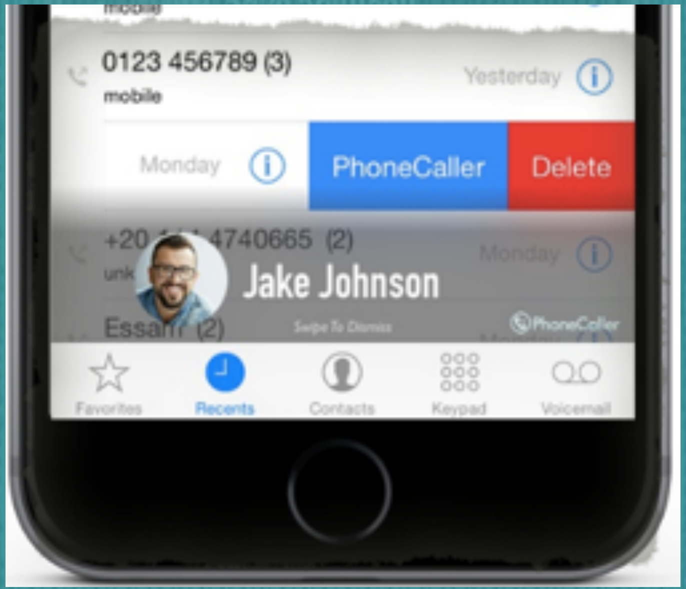
DEMO

Inspective-C

So what did i do?

- One of my great accomplishments is PhoneCaller
- It displays caller info when making/ receiving a call .. using TrueCaller's unofficial API
- Hooks into InCallServices, MobilePhone.app and TrueCaller.app
- 4000 lines of boring code ..
- More than 50,000 downloads (legal & illegal)


Mobinil
9:28 PM
98%
012 3456789
calling mobile...
Jake Johnson
PhoneCaller


0123 456789 (3)
mobile
Yesterday
Monday
PhoneCaller
Delete
Jake Johnson
PhoneCaller
Favorites
Recents
Contacts
Keypad
Voicemail