nvisium

iOS Testing Tools

David Lindner

Director of Mobile and IoT Security

Who is this guy?

David Lindner

- @golfhackerdave
- david.lindner@nvisium.com
- 15+ years consulting experience
- I hack and golf, sometimes at the same time.

nvisium

- Web Assessments
- Code Remediation
- Secure Development
- Training
- Continuous Security
- Mobile & IoT Assessments

Expertise in …

Microsoft .NET

Java™

iOS

RAILS

… and more

Disclaimer

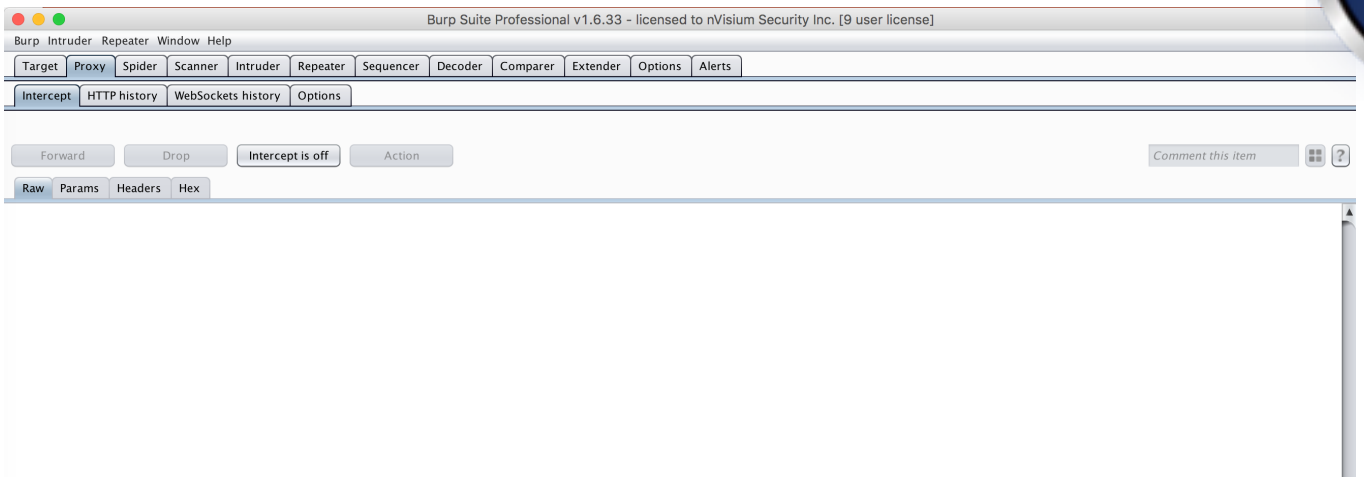
Hacking of App Store apps is not condoned or encouraged in any way. What you do on your own time is your responsibility. @golfhackerdave & nVisium take no responsibility if you use knowledge shared in this presentation for unsavory acts.

Agenda

- Proxy Traffic
- Runtime Analysis
- Memory Analysis
- Binary Analysis

Proxy iOS

- Step 1 – Pick your Proxy



Proxy iOS

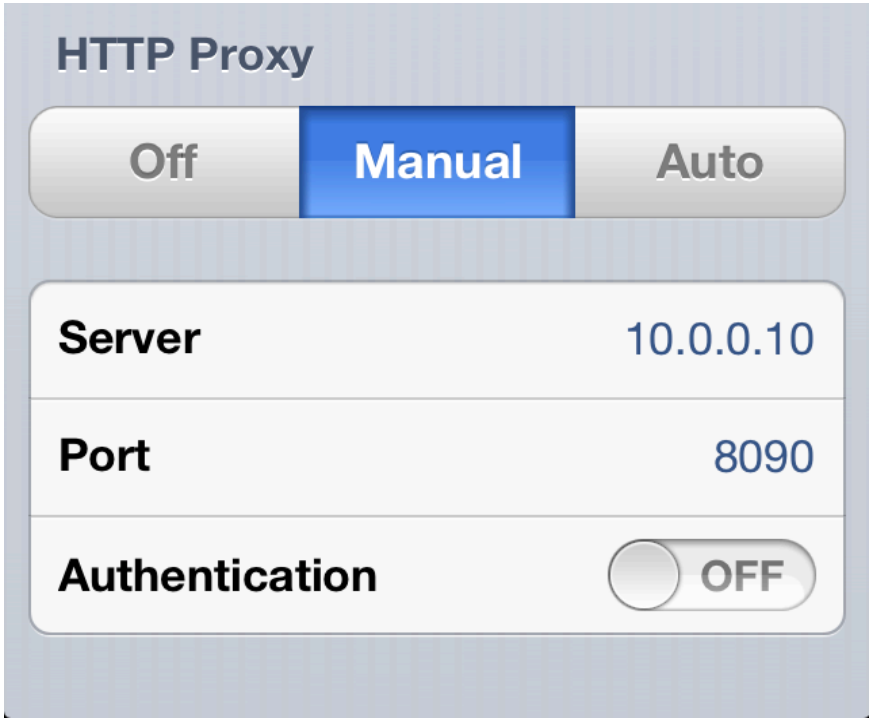
- Step 2 – Add CA to device

You Can Remove them

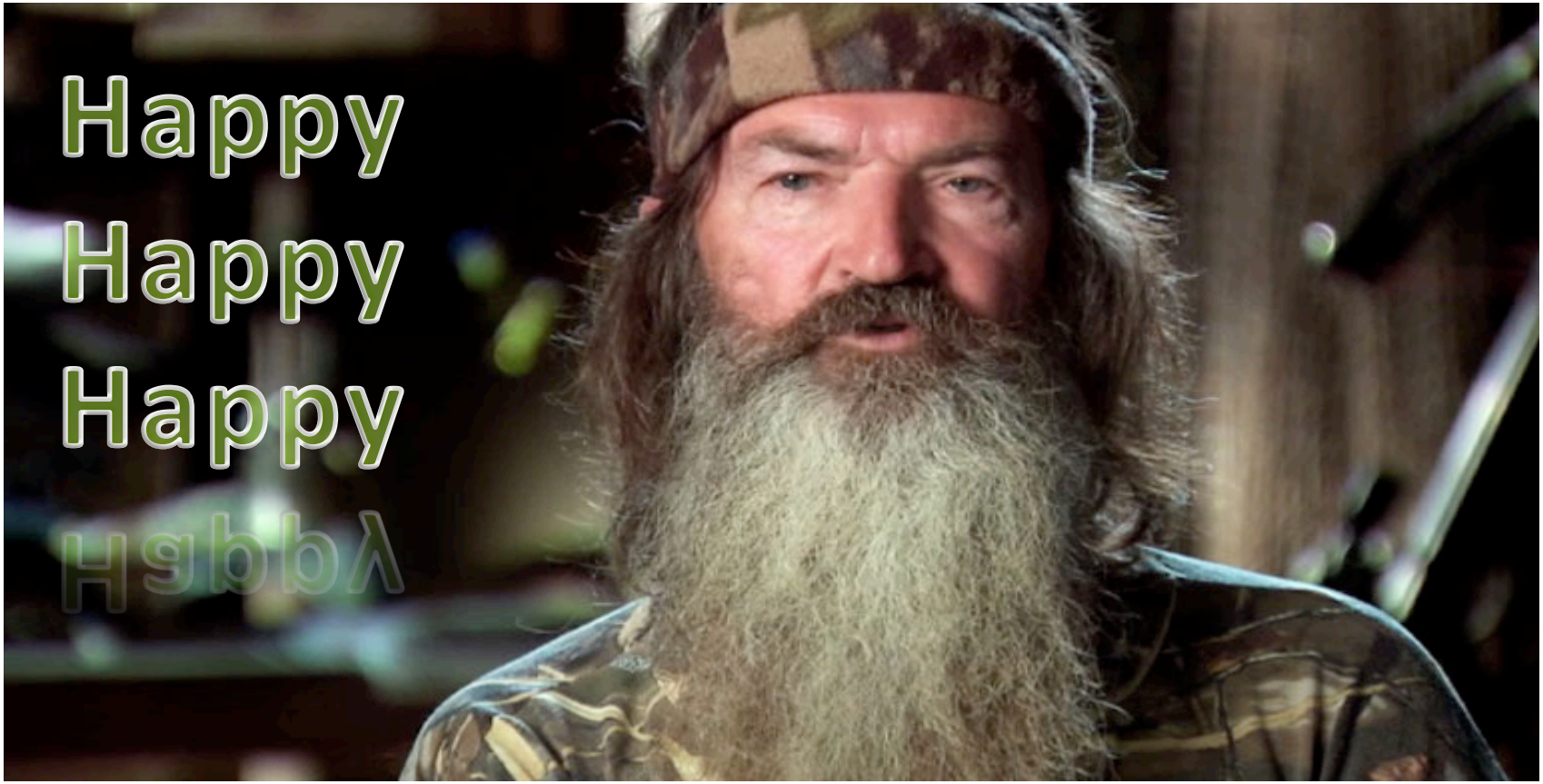
This works too but not as easy to manage!

Proxy iOS

- Step 3 – Set your system proxy
  - Settings -> WiFi -> YourNetwork



iOS Proxy Success!



iOS – Certificate Pinning bypass

- SSLKillSwitch
  - https://github.com/iSECPartners/ios-ssl-kill-switch

"MobileSubstrate extension to disable certificate validation within NSURLConnection in order to facilitate black-box testing of iOS Apps."

Runtime Analysis and Manipulation

- Monitor and modify the running application.
- Essentially “debugging.”
- Use cases:
  - Client-only applications.
  - Custom protocol (non-HTTP) communication.

Runtime Analysis: iOS

- iOS: Need a jailbroken device
  - Pangu for iOS 9-9.0.2
  - Possible to deploy to simulator (clunky, ugly)
  - Great tools for testing on devices
- Current Tools:
  - idb
  - cycript
  - snoop-it
- Resources
  - http://www.slideshare.net/jasonhaddix/pentesting-ios-applications
  - http://stackoverflow.com/questions/15076510/gdb-on-ipad-failing-to-dump-memory

idb

idb

Selected Device

USB device: Manually connect via SSH as root@localhost:2222

Status

Disconnect

Selected Application

Selected App: Photo Vault (com.enchantedcloud.photovault)

UUID: 72A5A34D-8447-4A24-A01D-B050E403772F

Select App...

App Info | Storage | URL Handlers | Binary | Filesystem | Tools | Log | Keychain | Pasteboard

App Details

Bundle ID	com.enchantedcloud.photovault
Bundle Name	Photo Vault
UUID	72A5A34D-8447-4A24-A01D-B050E403772F
URL Handlers	fb287622107920982 privatephotovault
Platform Version	9.1
SDK Version	iphoneos9.1
Minimum OS	7.0
Data Directory	/C199F76F-1733-4286-B706-5E094D45BC39

Launch App

Open Local Temp Folder

App Binary

Analyze Binary...

Encryption?	true
Cryptid	
PIE	true
Stack Canaries	true
ARC	true

App Entitlements

com.apple.developer.pass-type-identifiers	["P84C98Q322.*"]
com.apple.developer.team-identifier	P84C98Q322
com.apple.developer.default-data-protection	NSFileProtectionComplete
application-identifier	P84C98Q322.com.enchantedcloud.photovault
com.apple.security.application-groups	["group.enchantedcloud.shareExtension"]

https://github.com/dmayer/idb

idb Features

- File System Access
  - View all current and created files
  - Keychain access
  - Check for auto screenshots
  - Check iOS Logs
  - Check iOS Pasteboard
- App Analysis
  - Analyze app binary for encryption, PIE, ARC, etc
  - Run strings on App
  - Dump class information

idb: Installing

- Requires a ruby environment
- Follow instructions at http://www.idbtool.com/installation/

Live Demo: idb

```
[App Bundle]/PhotoVault.app/PlugIns/ImportExtension.appex/MainInterface.storyboardc/Info.plist => NSFileProtectionNone
[Data Dir]/Library/Albums.plist => NSFileProtectionComplete
[Data Dir]/Library/0/Photos.plist => NSFileProtectionComplete
[Data Dir]/Library/Decoy/Albums.plist => NSFileProtectionComplete
[Data Dir]/Library/Preferences/com.enchantedcloud.photovault.plist => NSFileProtectionComplete
[Data Dir]/Library/iTunes/Photos.plist => NSFileProtectionComplete
[Data Dir]/Library/Decoy/iTunes/Photos.plist => NSFileProtectionComplete
```

Snoop-it

Snoop-it

Photo Vault | Connection Status:

- Monitoring
  - Filesystem
  - Keychain
  - Network
  - Sensitive API
  - Common Crypto
- Analysis
  - Objective-C Classes
  - View Controller
  - URL Schemes
- Runtime Manipulation
  - Hardware Identifier
  - Fake Location
  - Method Tracing

Home | Filesystem | Keychain | Network | Sensitive API | Common Crypto | Objective-C Classes

Refresh Tree

- TKGlobal
- CSerializedJSONData
- CJSONSerializer
- TTStyleContext
- CDataScanner
  - CJSONScanner
- CJSONDeserializer
- TTShape
  - TTSpeechBubbleShape
  - TTRoundedRightArrowShape
  - TTRoundedRectangleShape
  - TTRoundedLeftArrowShape
  - TTRectangleShape
- CJSONDataSerializer
- AppDelegate
- TTModel
  - TTURLRequestModel
    - MockPhotoSource
- TTLayout
  - TTGridLayout
  - TTFlowLayout
- MockPhoto
- TTURLRequestQueue
- TTURLRequest
- TTURLImageResponse
- TTURLDataResponse
- TTURLCache
- TTRequestLoader
- TTUserInfo

Selected Method: - (void) pinLockControllerDidFinishUnlocking; | Setup and Invoke

```
//Classtype: App-Class
//Instances: 0x2c76e0
@Interface AppDelegate : NSObject<UIApplicationDelegate
MFMailComposeViewControllerDelegate UIAlertViewDelegate> //Protocols {
//Instance Variables

    UIViewController* viewController;
    UITabBarController* aTabBarController;
    NSMutableArray* openedAlbums;
}
//Properties
@property (nonatomic) UITabBarController* aTabBarController; // = aTabBarController
//Methods

- (void) emailPin;
- (void) lockController:(id)arg0 didFinish:(id)arg1;
- (void) pinLockControllerDidFinishUnlocking;
- (void) pinLockControllerDidCancel;
- (void) pinLockController:(id)arg0 didFinishSelectingNewPin:(id)arg1;
- (void) pinLockControllerDidFinishRemovingPin;
- (void) runOnce;
- (id) documentsDirectory;
- (void) setATabBarController:(id)arg0;
- (id) aTabBarController;
- (void) addAlbumToOpenedAlbums:(id)arg0;
- (bool) checkInstaLockForAlbum:(id)arg0;
- (void) promptForEmail;
- (void) pinManagement;
- (bool) navigator:(id)arg0 shouldOpenURL:(id)arg1;
- (void) lockControllerDidCancel:(id)arg0;
- (bool) application:(id)arg0 handleOpenURL:(id)arg1;
- (void) applicationWillResignActive:(id)arg0;
```

Debug Report | Search :

https://code.google.com/p/snoop-it/

Snoop-it Features

- Monitoring
  - File system access (print data protection classes)
  - Keychain access
  - HTTP(S) connections (NSURLConnection)
  - Access to sensitive API (address book, photos etc.)
  - Debug outputs (NSLog)
  - Tracing App internals (objc_msgSend)
- Anaylsis/Manipulation
  - Fake hardware identifier (UDID, Wireless MAC, etc.)
  - Fake location/GPS data
  - Explore and force display of available ViewController

Snoop-it: Installing

1. Add the Cydia repository repo.nesolabs.de and install the provided snoop-it package
2. After installing, run the Snoop-it Configuration App by tapping the Snoop-it icon on SpringBoard
3. Using the Snoop-it Configuration App, select the Apps to analyze.
4. Adjust the Snoop-it settings if desired (like e.g. the listening port of the web interface, authentication, tracing, etc.)
5. Run the selected App & point the browser in a computer to the Snoop-it web interface.

nvisium

Live Demo: Snoop-it

Snoop-it

Monitoring
- Filesystem
- Keychain
- Network
- Sensitive API
- Common Crypto

Analysis
- Objective-C Classes
- View Controller
- URL Schemes

Runtime Manipulation
- Hardware Identifier
- Fake Location
- Method Tracing

Photo Vault | Connection Status:

Home | View Controller

List of all available ViewControllers

- TTNavigatorWindow
  - TabbarController
    - TTNavigationController
      - AlbumList
    - TTNavigationController
      - ImportViewController
    - TTNavigationController
      - SettingsViewController
  - DTPinLockController
    - UIViewController
  - UITextEffectsWindow

ID:
Class Name:
Title:
Displayed Modal:
Type:
canBeDismissed:
Visible:
VC Index:

Display Controller

Enter Passcode

1 | 2 ABC | 3 DEF
4 GHI | 5 JKL | 6 MNO
7 PQRS | 8 TUV | 9 WXYZ
0

Debug Report

- TTURLImageResponse
- TTURLDataResponse
- TTURLCache
- TTRequestLoader
- TTUserInfo

- (void) pinManagement;
- (bool) navigator:(id)arg0 shouldOpenURL:(id)arg1;
- (void) lockControllerDidCancel:(id)arg0;
- (bool) application:(id)arg0 handleOpenURL:(id)arg1;
- (void) applicationWillResignActive:(id)arg0;

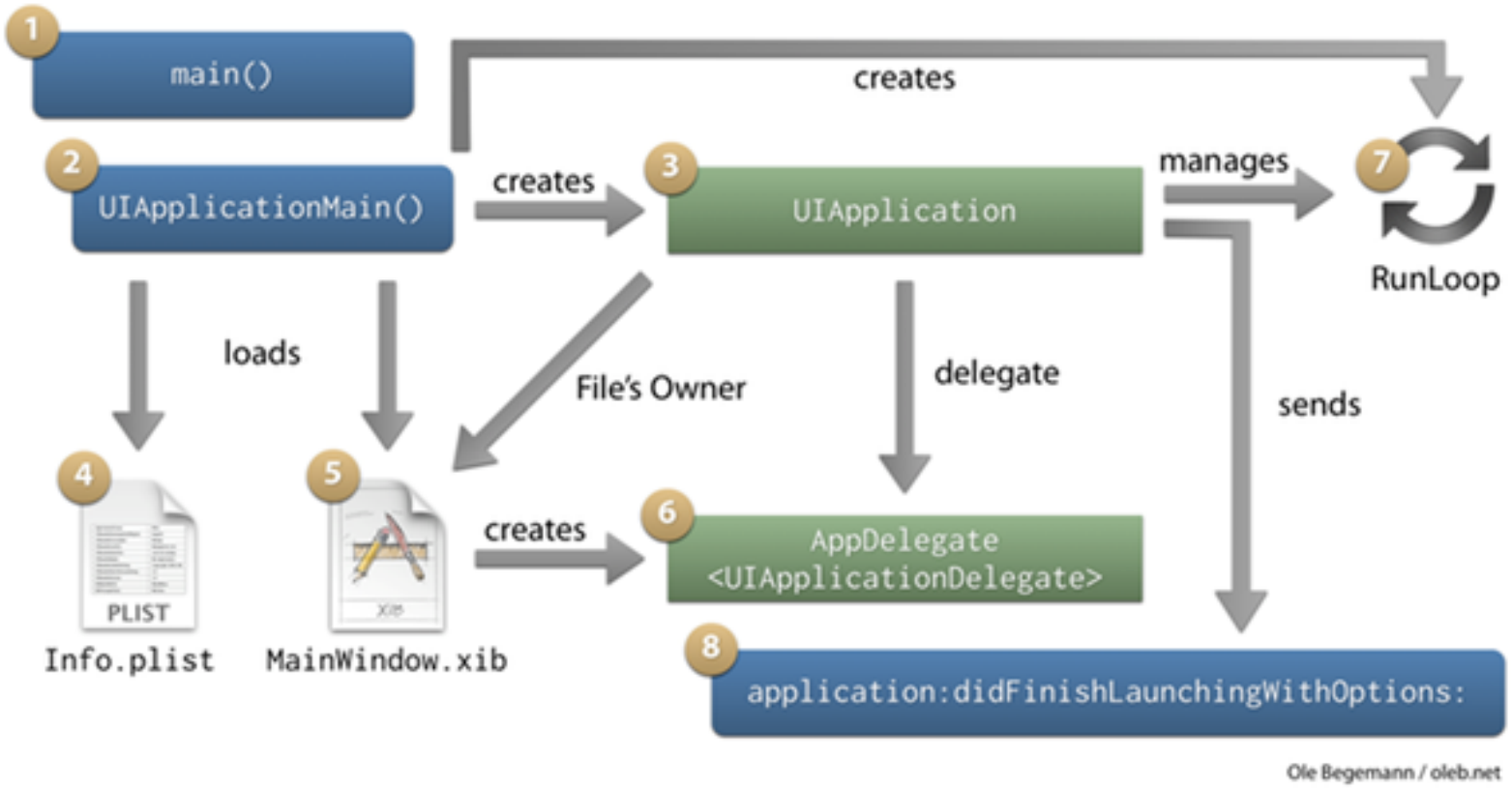
Search :

Cycript

"Cycript allows developers to <u>explore and modify running applications</u> on either iOS or Mac OS X using a hybrid of Objective-C++ and JavaScript syntax through an interactive console that features syntax highlighting and tab completion."

http://www.cycript.org/

iOS Execution Flow



What is Cycript

- A programming language designed to blend Objective-C and JavaScript.
- Allows hooking into an iOS process or application.
- It grants access to all of the classes and instance variables and methods within the application.
- Can write and execute scripts.

Cycript: How?

- Available for jailbroken iOS devices (Cydia installation)
- Command line interface.

```
cy# @"hello"
@"hello"
cy# @[2, 4, 5]
@[2,4,5]
cy# @{thing: 9, field: 10}
@{"thing":9,"field":10}
cy# @YES
@true
cy# @null
@null
cy# @(5 + 7)
@12
```

Cycript: Why?

- Bypass client-side validations.
- Obtain sensitive data stored on memory (passwords, private keys, certificates, etc).
- Call methods directly.
- Overwrite methods (aka Method Swizzling).
- Similar capabilities to editing HTML/JS on a Web App (but much more complicated).

Cycript: Installing

- Cycript is available in Cydia:
  1. Open cydia on a jailbroken iOS device
  2. In case cycript is now showing up in the search results page make sure the "Developer" option is selected under "Manage"->"Settings".
  3. Select the cycript package and install it.
- If the Cycript package is not seen in Cydia:
  1. Navigate to http://www.cycript.org/debs/ and download the latest available.
  2. Copy this file to the iDevice by using SFTP.
  3. SSH into the iDevice and install it, you may need root/sudo for this:
     ```
     # dpkg -i cycript_iphoneos-arm.deb
     ```
  4. Verify the installation by executing the following command:
     ```
     # cycript
     ```
  5. If the installation was successful a cycript interactive shell will be displayed:
     ```
     cy#
     ```

Cycript: Usage

- Obtain command line access to the device using SSH.
- Cycript needs to be attached/hooked to a process.

```
# cycript -p Application
```

- Where "Application" is the name of the application running on the device
- If cycript is not able to start a process, an ID can be provided.

```
# ps aux
# cycript -p {process id}
```

- Get the name of the application delegate class.

```
cy# UIApp.delegate
#"<AppDelegate: 0x28a600>"
cy# UIApp.keyWindow.rootViewController
```

Cycript: Usage Continued

- Dump all classes

```
cy# ObjectiveC.classes
```

- Get a class memory address

```
cy# choose(SomeClass)
#"<SomeClass: 0x28a600>”
```

- Attach to instance of Class

```
cy# var somcls = new Instance(0x28a600)
```

- Run Methods in Class

```
cy# [somcls someMethod: someParm]
```

Cycript Common Functions

```
function tryPrintIvars(a){ var x={}; for(i in *a){ try{ x[i] = (*a)[i]; } catch(e){} } return x; }

function printMethods(className) {
  var count = new new Type("I");
  var methods = class_copyMethodList(objc_getClass(className), count);
  var methodsArray = [];
  for(var i = 0; i < *count; i++) {
    var method = methods[i];
    methodsArray.push({selector:method_getName(method), implementation:method_getImplementation(method)});
  }
  free(methods);
  free(count);
  return methodsArray;
}

function methodsMatching(cls, regexp) { return [[new Selector(m).type(cls), m] for (m in cls.messages) if (!regexp ||
regexp.test(m))]; }
```

http://iphonedevwiki.net/index.php/Cycript_Tricks

Cycript Method Swizzling

```
cy# SomeClass.messages

cy# SomeClass.messages['someMethod'] = function(){return true;}
```

Live Demo: Cycript

```
cy# function printMethods(className) {
cy>   var count = new new Type("I");
cy>   var methods = class_copyMethodList(objc_getClass(className), count);
cy>   var methodsArray = [];
cy>   for(var i = 0; i < *count; i++) {
cy>     var method = methods[i];
cy>     methodsArray.push({selector:method_getName(method), implementation:method_getImplementatio
n(method)});
cy>   }
cy>   free(methods);
cy>   free(count);
cy>   return methodsArray;
cy> }
cy#
```

What About Swift

- No tools…yet
- Write custom hooks through Mobile Substrate
  - https://www.securify.nl/blog/SFY20150302/hooking_swift_methods_for_fun_and_profit.html
  - https://www.uraimo.com/2015/10/23/effective-method-swizzling-with-swift/

Memory Dumping and Analysis

- Why do we care about memory?
  - What type of info is retained? How long?
  - Is it necessary?
  - Could an attacker recover it if lost / stolen?

Memory Analysis: iOS

- Back to Snoop-it/Cycript/idb

```
cy# @"hello"
@"hello"
cy# @[2, 4, 5]
@[2,4,5]
cy# @{thing: 9,
@{"thing":9,"fie
cy# @YES
@true
cy# @null
@null
cy# @(5 + 7)
@12
```

Snoop-it

Photo Vault | Connection Status:

- Monitoring
  - Filesystem
  - Keychain
  - Network
  - Sensitive API
  - Common Crypto
- Analysis
  - Objective-C Classes
  - View Controller
  - URL Schemes
- Runtime Manipulation
  - Hardware Identifier
  - Fake Location
  - Method Tracing

Home | View Controller

List of all available ViewControllers

- TTNavigatorWindow
  - TabbarController
    - TTNavigationController
      - AlbumList
    - TTNavigationController
      - ImportViewController
    - TTNavigationController
      - SettingsViewController
  - DTPinLockController
    - UIViewController
  - UITextEffectsWindow

ID:	0x493a6e0
Class Name:	ImportViewController
Title:	Import
Displayed Modal:	false
Type:	UIViewController
canBeDismissed:	false
Visible:	false
VC Index:	0

Display Controller | Close/Hide Controller | Reset Display

Debug Report

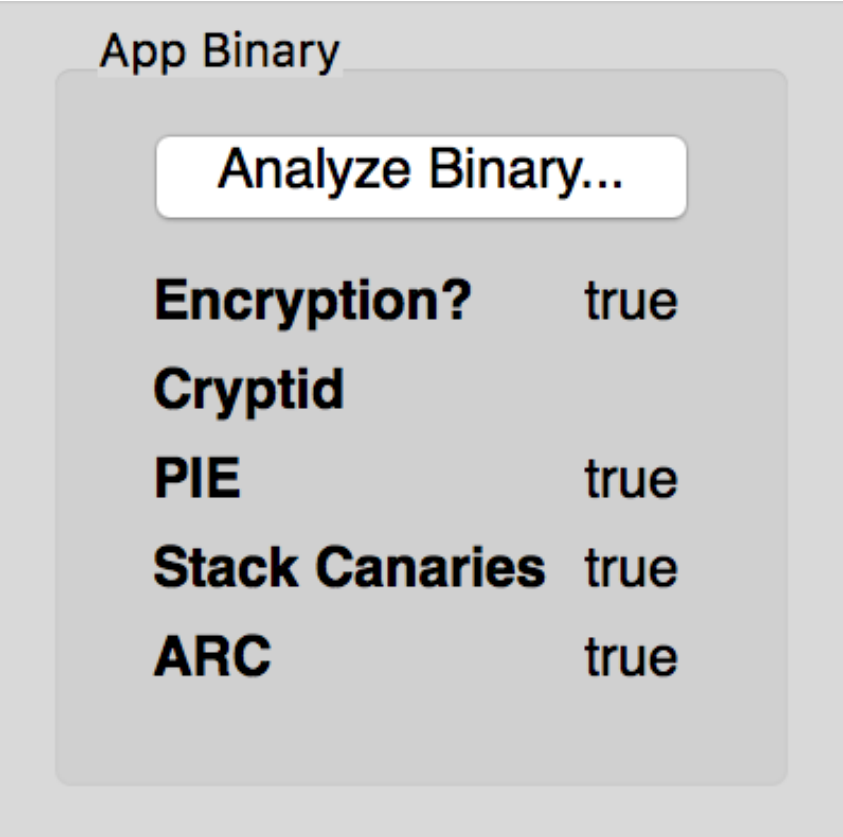
Search :

iOS Binary Analysis

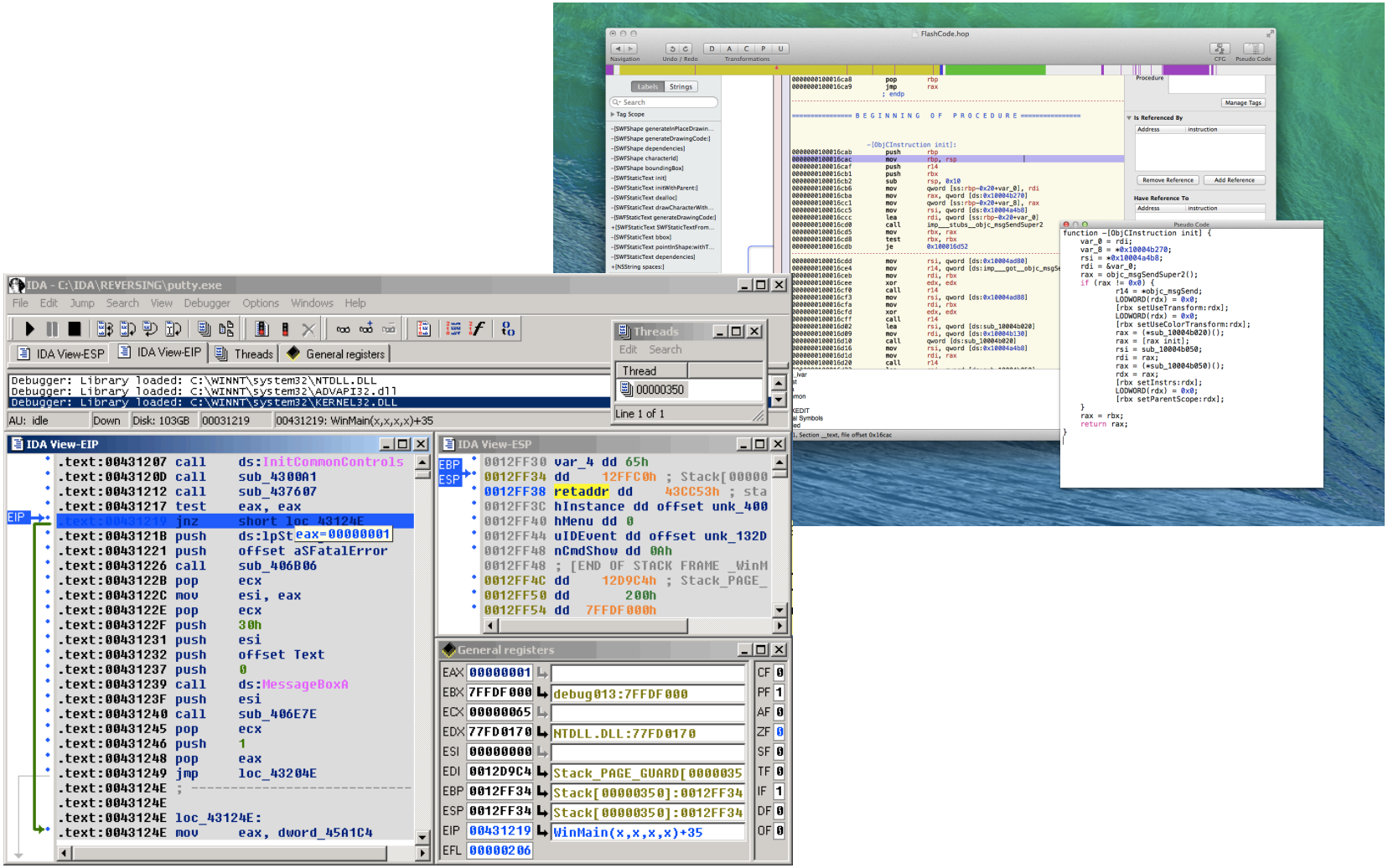
- iOS binaries are native code (read: cannot decompile).
- IPA files from iTunes have their binary code encrypted with Fairplay
- Disassembler tools:
  - Hopper http://www.hopperapp.com/
  - IDA pro https://www.hex-rays.com/products/ida/
- Resources:
  - https://rstforums.com/forum/79368-ios-app-decompilation.rst
  - http://resources.infosecinstitute.com/penetration-testing-for-iphone-applications-part-5/

But the app is encrypted???

- idb FTW!
- Uses dumpdecrypted



Disassembler



Other Tools

- Methods
  - class-dump
  - nm
  - strings
- Logging
  - idevicesyslog
- Networking
  - rvictl
  - iproxy

david.lindner@nvisium.com

@golfhackerdave

https://linkedin.com/in/dlindner