



black hat[®]
ASIA 2016

Su-a-Cyder: Home-Brewing iOS Malware Like a BO\$\$!

Chilik Tamir

chilik@mi3security.com

Twitter: [@_coreDump](https://twitter.com/_coreDump)

This talk will cover the latest iOS malware creation & capabilities running on a non-jailbroken device with latest version of iOS

This talk will not discuss targeting a jailbroken device

Who Am I

Security Researcher – iOS iNalyzer PT framework

Security Trainer:     

Security Speaker:   

Chief Architect of R&D at Mi3 Security 

B.Sc. Biomedical Engineering

Twitter addict From Israel

Machine - Meta - Mobile
Intelligence for the next billion apps

Overview

- ⌘ iOS Sandbox and Ecosystem
- ⌘ Historic iOS Malware Properties
- ⌘ Malware Reincarnations
- ⌘ Corporate Targeting Malware and Personal Targets
- ⌘ Detection and Mitigation

About This Talk

This talk will cover the current state of iOS malware. It will disclose a new approach for iOS malware creation.

It will address the impact on corporate & private sector. It will outline updated mitigation and best practices.

Questions to be addressed

- ⌘ Can iOS be targeted by malware?
 - ⌘ What about Apple mitigation against malware?
- ⌘ Who may be affected by iOS malware?
 - ⌘ I'm running iOS 9.2.1 non-jailbroken, am I affected?
- ⌘ What can iOS malware achieve?
 - ⌘ What's the worst case scenario?



black hat[®]
ASIA 2016

The iOS Ecosystem

iOS Playground Rules

- ⌘ All code must be signed



iOS Playground Rules – Code Sign

Every code running on an iOS device must be properly signed with an Apple provided certificate (Developer or Distributor).

Code replacing, application patching, and repackaging of an iOS application **should** not be possible.

iOS Playground Rules

- ⌘ All code must be signed
- ⌘ All apps are subjected to a review process

App Review

The app review process ensures that apps on the App Store and Mac App Store are reliable, perform as expected, and are free of explicit and offensive material. We review every app submitted based on a set of technical, content, and design criteria.

<https://developer.apple.com/support/app-review/>

iOS Playground Rules – App Review Process

Apple requests that developers submit their application for a peer review for stability and functionality.

This process attempts to ban unwanted apps from the Apple App Store.

Top 10 reasons for app rejections during the 7-day period ending February 3, 2015.

- 14% More information needed
- 9% Guideline 2.2: Apps that exhibit bugs will be rejected
- 6% Guideline 10.6: Apple and our customers place a high value on simple, refined, creative, well thought through interfaces. They take more work but are worth it. Apple sets a high bar. If your user interface is complex or less than very good, it may be rejected
- 5% Guideline 22.2: Apps that contain false, fraudulent or misleading representations or use names or icons similar to other Apps will be rejected
- 4% Guideline 3.3: Apps with names, descriptions, screenshots, or previews not relevant to the content and functionality of the App will be rejected
- 4% Guideline 17.2: Apps that require users to share personal information, such as email address and date of birth, in order to function will be rejected
- 4% Guideline 3.1: Apps or metadata that mentions the name of any other mobile platform will be rejected
- 3% Guideline 3.8: Developers are responsible for assigning appropriate ratings to their Apps. Inappropriate ratings may be changed/deleted by Apple
- 3% Guideline 3.4: App names in iTunes Connect and as displayed on a device should be similar, so as not to cause confusion
- 3% Guideline 2.16: Multitasking Apps may only use background services for their intended purposes: VoIP, audio playback, location, task completion, local notifications, etc.

iOS Playground Rules

- ⌘ All code must be signed
- ⌘ All apps are subjected to a review process
- ⌘ Gaining certificates requires proper identification

iOS Playground Rules – Developer Certificates

Apple requires iOS Developers to supply proper identification during registration.

Apple can request further proof of identification at their will.

What You Need to Enroll



Enrolling as an Individual

If you are an individual or sole proprietor/single person business, sign in with your Apple ID to get started. You'll need to provide basic personal information, including your legal name and address.



Enrolling as an Organization

If you're enrolling your organization, you'll need an Apple ID as well as the following to get started:

A D-U-N-S[®] Number

Your organization must have a D-U-N-S Number so that we can verify your organization's identity and legal entity status. These unique nine-digit numbers are assigned by Dun & Bradstreet and are widely used as standard business identifiers. You can check to see if your organization already has a D-U-N-S Number and request one if necessary. They are free in most jurisdictions. [Learn more >](#)

Legal Entity Status

Your organization must be a legal entity so that it can enter into contracts with Apple. We do not accept DBAs, Fictitious Businesses, Trade names, or branches.

Legal Binding Authority

As the person enrolling your organization in the Apple Developer Program, you must have the legal authority to bind your organization to legal agreements. You must be the organization's owner/founder, executive team member, senior project lead, or have legal authority granted to you by a senior employee.



iOS Playground Rules

- ⌘ All code must be signed
- ⌘ All apps are subjected to a review process
- ⌘ All certificates require identification
- ⌘ All installations are validated on device

iOS Playground Rules – Installation Validation

Every installation on the device requires a signed package.

In addition, since version 9, iOS validates the application developer certificate against Apple to identify misused and abused Development certificates (such as Provision any Certs).

iOS Playground Rules

- ⌘ All code must be signed
- ⌘ All apps are subjected to a review process
- ⌘ All certificates require identification
- ⌘ All installation are validated on device
- ⌘ Any misbehaving developer will be accountable**

iOS Playground Rules – Developer & App Bans

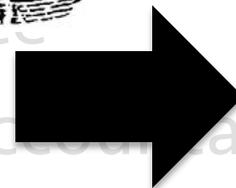
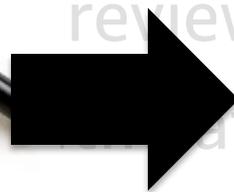
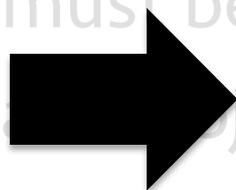
Apple has an identity to target with sanctions

It has been proven that misbehaving Developers were banned

Applications removed from store

Apple can pull an app from a device remotely

iOS Playground Rules



- ⌘ All installation are validated on device
- ⌘ Any misbehaving developer will be accessed



black hat[®]
ASIA 2016

iOS Malware Distribution Review

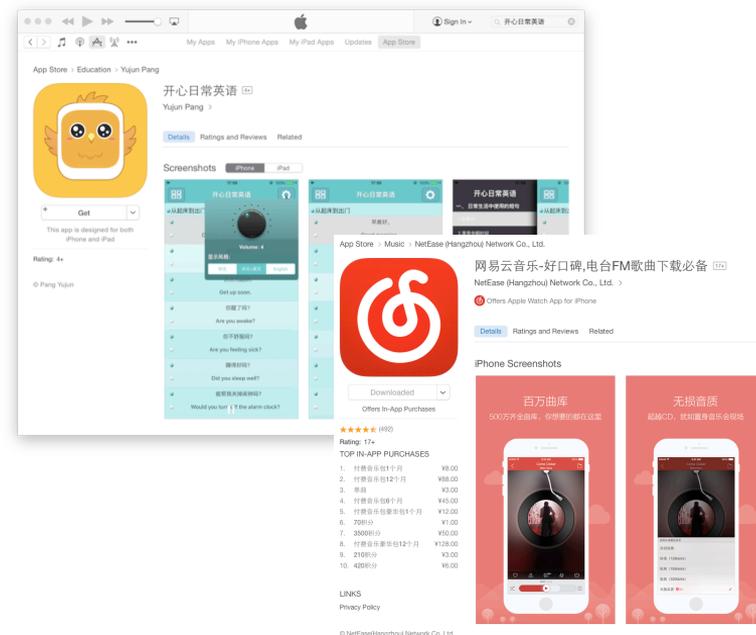
Distribution Tracks

Malware in the Apple App Store

Malware from a Distributor / Developer

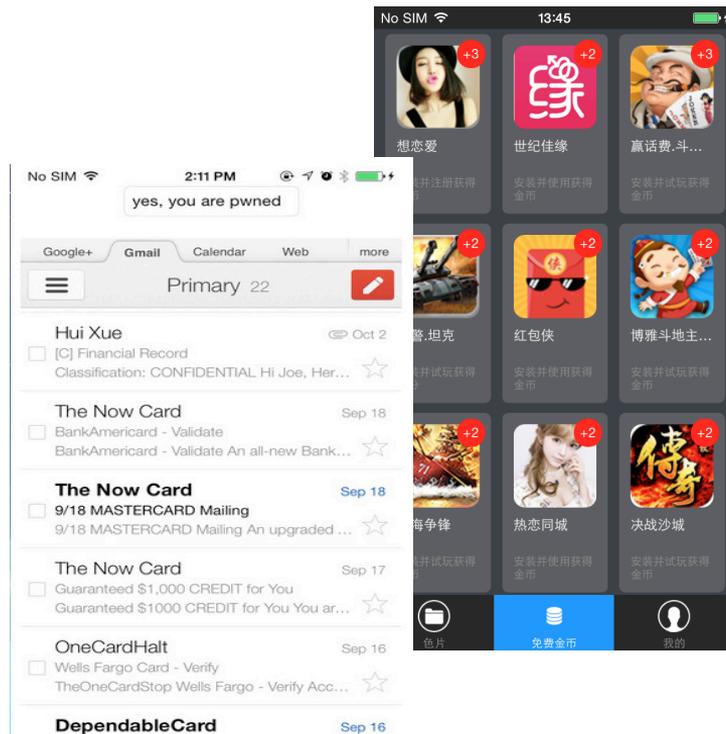
Distribution Tracks – App Store Malware

- ⌘ ZergHelper (Claud Xiao, paloalto networks)
- ⌘ xCodeGohst (Claud Xiao, paloalto networks)



Distribution Tracks – Distributor Malware

- ⌘ Yispector, WireLurker (Claud Xiao, paloalto networks)
- ⌘ masque-attack (Hui Xue, Tao Wei, Yulong Zhang, FireEye)



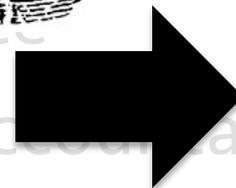
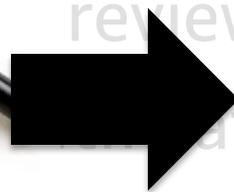
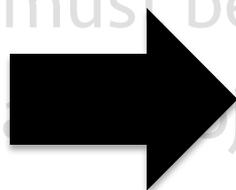
Historical Malware Capabilities

- ⌘ Abuse private API to install and remove apps programmatically
- ⌘ Abuse access to Address Book
- ⌘ Abuse access to Calendar
- ⌘ Abuse access to Photo EXIF metadata
- ⌘ Abuse access to Microphone recording
- ⌘ Abuse pin-point GPS Locationing

Historical Malware capabilities

- ⌘ Deanonymization of user
- ⌘ Hijacking of legitimate CFURL calls
- ⌘ Phishing credentials
- ⌘ Polymorphism by remote updates

iOS Playground Rules



- ⌘ All installation are validated on device
- ⌘ Any misbehaving developer will be accessed

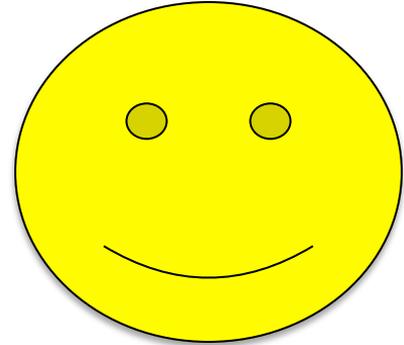
Corporate Security Response

Invest in Jailbreak detection

Vendor will install an agent on the device



Is device
Jailbroken



2016

BlackBerry.

INSIDE BlackBerry Blog

INSIDE BlackBerry BUSINESS BLOG
The official BlackBerry Blogs

Follow      

News

Case Studies

Apps

Software Solutions

Security

Home > BYOD | Devices | Enterprise | Security > Jailbroken Security: Why Your Enterprise Needs Jailbreak Detection

Jailbroken Security: Why Your Enterprise Needs Jailbreak Detection

BYOD

09.28.15 / Jay Barbour
2 Comments

Share 











Recently, a piece of iOS malware by the name of Keyraider stole the account information of over 225,000 Apple device users, along with thousands of certificates, private keys, device IDs and purchasing receipts. It's possibly one of the largest malware attacks suffered by the platform in history, and has already impacted users in over 18 countries. In addition to downloading massive amounts of data, the malware's distributors have also used it to hold devices for ransom.

Interestingly, but unsurprisingly, Keyraider targets only jailbroken devices.

Although jailbreaking and rooting gives a device's user more



<http://bizblog.blackberry.com/2015/09/jailbroken-security-why-your-enterprise-needs-jailbreak-detection/>

2014



[BLOG HOME](#)

[ATTEND A WEBINAR](#)

ELIMINATE THE UNKNOWN OF JAILBROKEN DEVICES

March 19, 2014

Posted by: Jackie Roewe

Systems Manager can now automatically detect enrolled jailbroken devices.

<https://meraki.cisco.com/blog/2014/03/eliminate-the-unknown-of-jailbroken-devices/>

MAY **MOBILITY**

11
2012



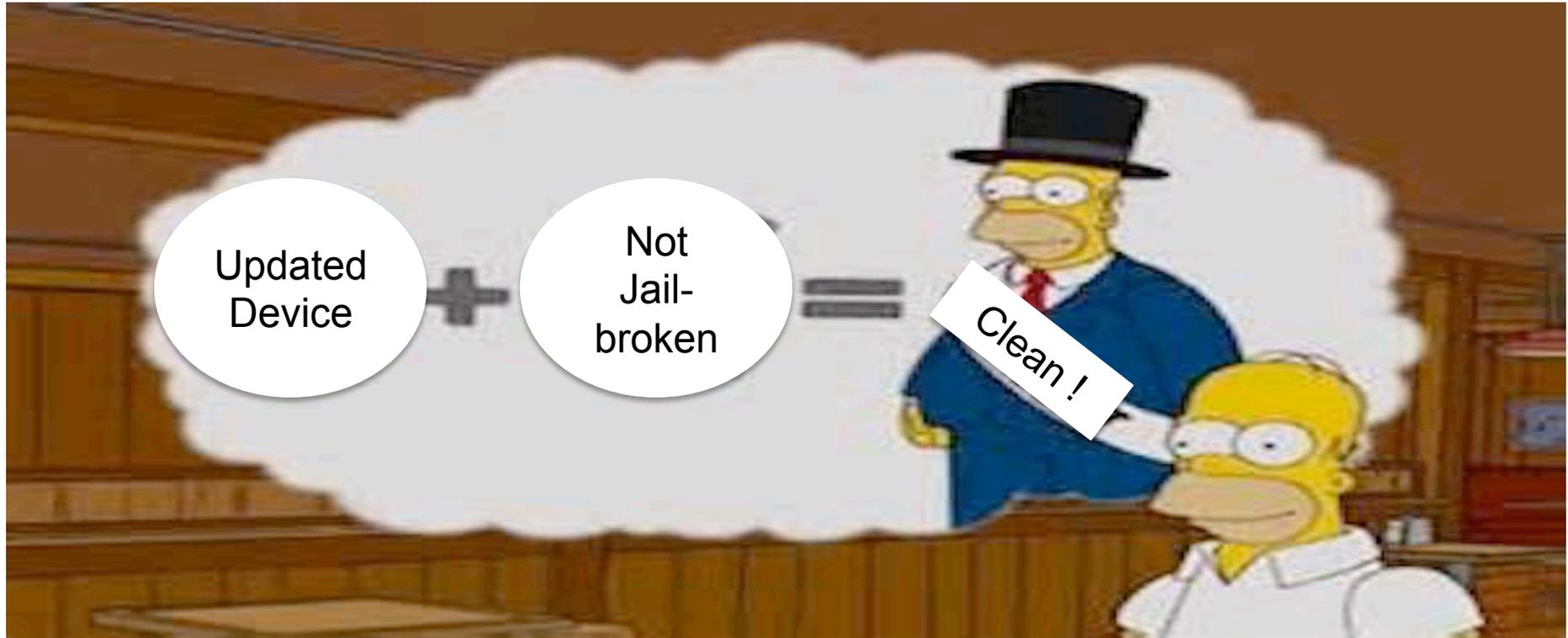
Why Jailbroken Devices Are a Security Risk and How MDM Can Detect Them

Enterprises need to be on the lookout for jailbroken devices as they potentially put corporate data at risk.

by [BizTech Staff](#)

<http://www.biztechmagazine.com/article/2012/05/why-jailbroken-devices-are-security-risk-and-how-mdm-can-detect-them>

Security Vendor Assumption





black hat[®]
ASIA 2016

Sign in with Apple ID

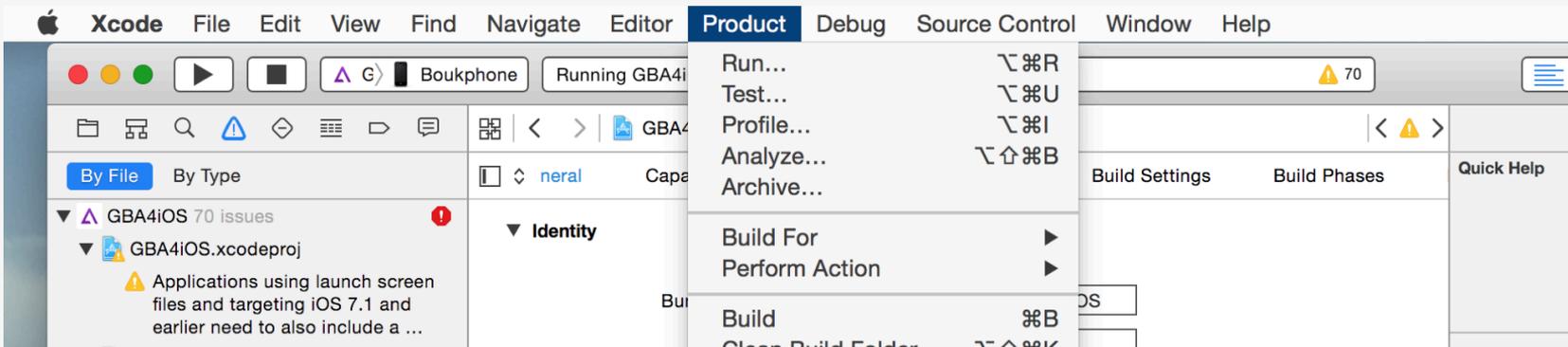


JUNE 10, 2015

Xcode 7 allows anyone to download, build and 'sideload' iOS apps for free

Benjamin Mayo - 9 months ago [🐦 @bzamayo](#)

[APPS DEVELOPERS](#) [IOS](#) [IOS DEVICES](#) [TECH](#) [INDUSTRY](#)



Support

Overview — Development — Distribution — /

Individuals

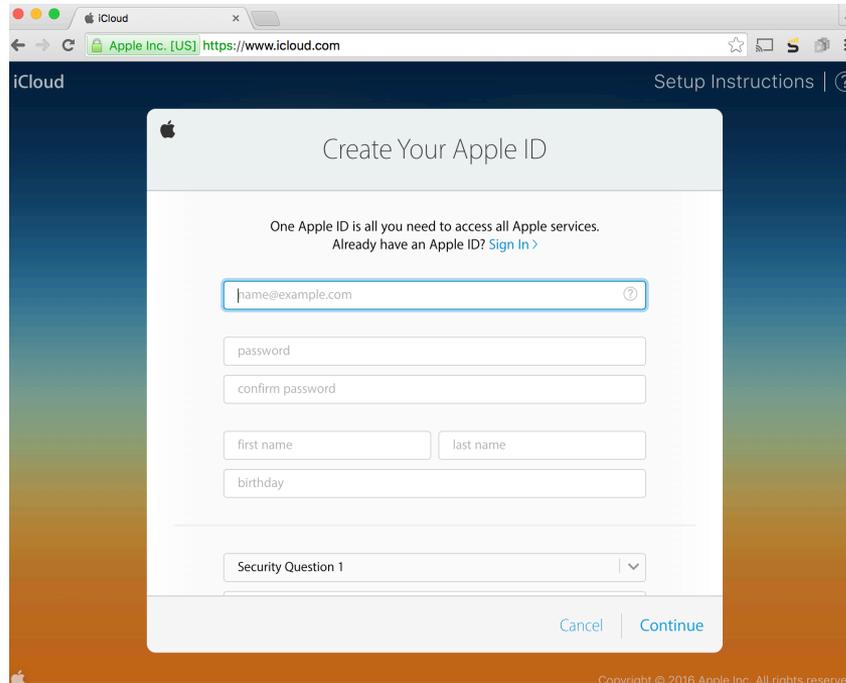
If you're looking to develop apps for Apple platforms, download the Xcode developer tools, SDKs, and resources for free on the [Xcode downloads page](#). **Program membership is not required.** If you don't already have an Apple ID, you can [create one here](#).

Apple Developer Program. If you're an individual or sole proprietor/single person business interested in creating apps for distribution on the App Store for iPhone, iPad, Mac, and Apple Watch, enroll in the Apple Developer Program. Membership includes access to beta OS releases, advanced app capabilities, and tools to develop, test, and distribute apps and Safari extensions. Developers enrolled as individuals will sell apps on the App Store using their personal name.

99 USD per membership year

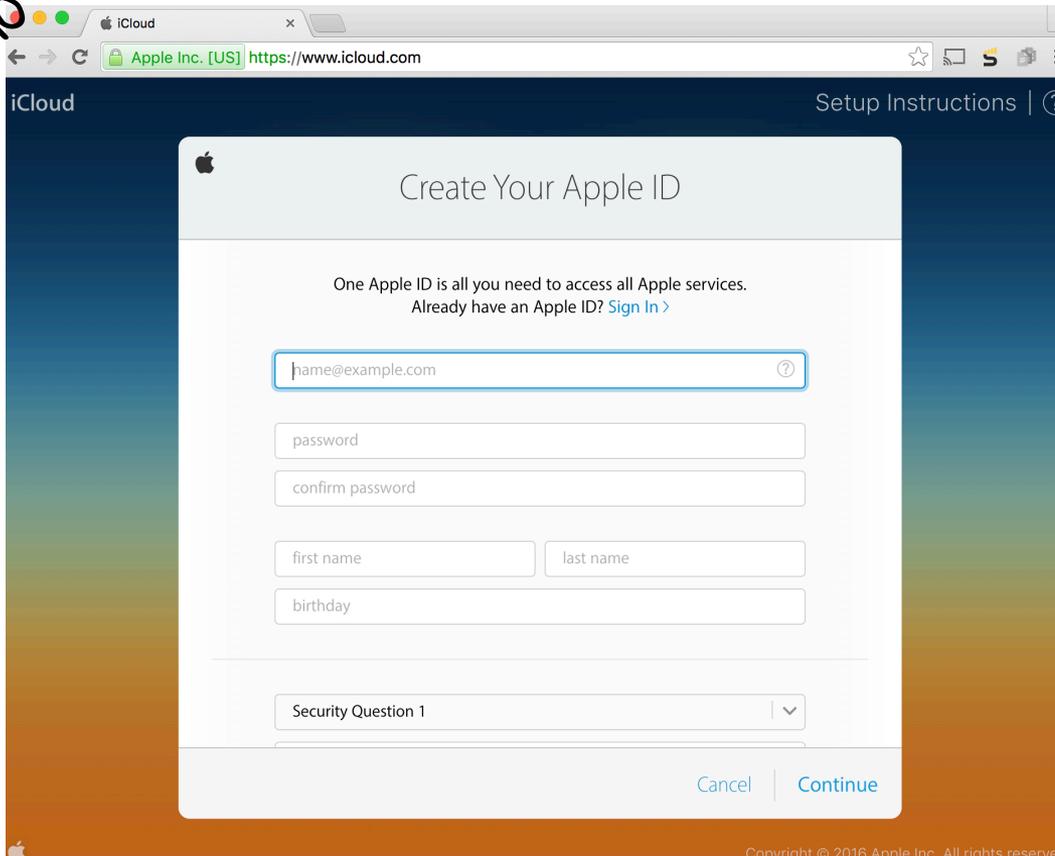
Sign in With Apple ID:

- ⌘ Does not require Identification documents
- ⌘ Any valid email address is sufficient



The screenshot shows a web browser window displaying the Apple ID creation page. The page title is "Create Your Apple ID". Below the title, there is a message: "One Apple ID is all you need to access all Apple services. Already have an Apple ID? [Sign In >](#)". The form contains several input fields: an email address field (pre-filled with "jame@example.com"), a password field, a confirm password field, first name and last name fields, and a birthday field. At the bottom, there is a "Security Question 1" dropdown menu. The "Continue" button is highlighted in blue. The browser's address bar shows "https://www.icloud.com".

DEMO



The screenshot shows a web browser window with the URL `https://www.icloud.com`. The page title is "iCloud" and the main heading is "Create Your Apple ID". Below the heading, there is a message: "One Apple ID is all you need to access all Apple services. Already have an Apple ID? [Sign In >](#)". The form contains several input fields: "name@example.com" (with a help icon), "password", "confirm password", "first name", "last name", and "birthday". At the bottom, there is a "Security Question 1" dropdown menu. The "Continue" button is highlighted in blue.

iCloud Setup Instructions | ?

Create Your Apple ID

One Apple ID is all you need to access all Apple services.
Already have an Apple ID? [Sign In >](#)

?

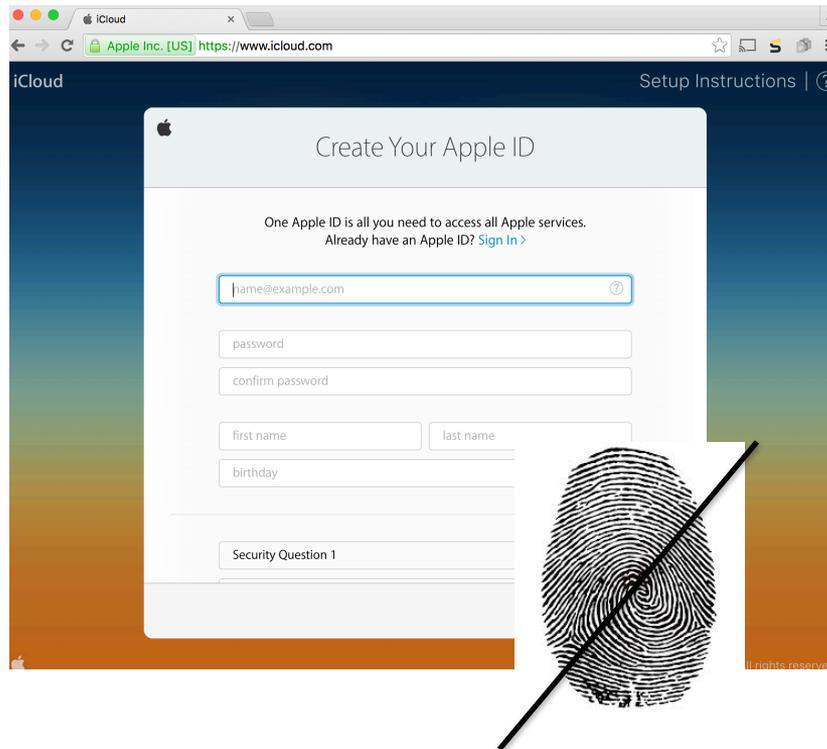
Security Question 1 ▾

[Cancel](#) | [Continue](#)

Copyright © 2016 Apple Inc. All rights reserved.

Sign in With Apple ID

- ⌘ Anonymous Developer
- ⌘ No target for attribution
- ⌘ Can always regenerate
- ⌘ Resigning with new Certs



Capabilities Available to Developers

	Sign in with Apple ID	Apple Developer Program members
App Groups	•	•
Background Modes	•	•
Data Protection	•	•
HealthKit	•	•
HomeKit	•	•
Inter-App Audio	•	•
Keychain Sharing	•	•
Maps	•	•
Wireless Accessory Configuration	•	•

Apple Pay

Associated Domains

Game Center

iCloud / CloudKit

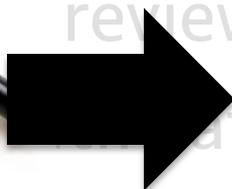
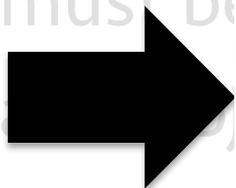
In-App Purchase

Passbook / Wallet

Personal VPN

Push Notifications

iOS Playground Rules



- ⌘ All installations are validated on devices
- ⌘ Any misbehaving developer will be accessed

WARNING: PSEUDO - MATH

EXPLOITABILITY: DOES THIS AFFECT ME / MY ORG?

EXPLOITABILITY $\frac{F(\text{PROFIT})}{G(\text{LOSS})} = \frac{F(\text{MOTIVE, OPPORTUNITY})}{G(\text{COST, ATTRIBUTION})}$

OPPORTUNITY JUST SCALED
ATTRIBUTION DECREASED

Most Common Malware

Either targeting an application or a user:

- ⌘ Evil-Client: Replacing an original application (hiding in plain sight)
- ⌘ Evil-Sample: Providing a new Sample (hiding the evil functionality)



black hat[®]
ASIA 2016

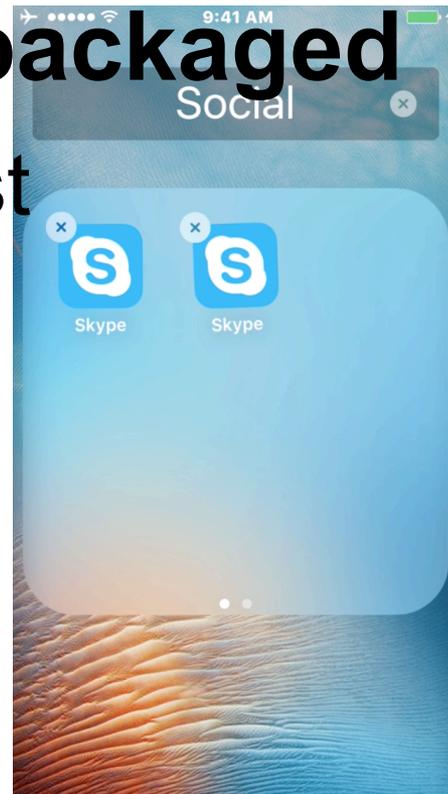
Malware as an Evil Client Building Blocks



Evil Client – Malicious / Repackaged

Using Original Application as a host

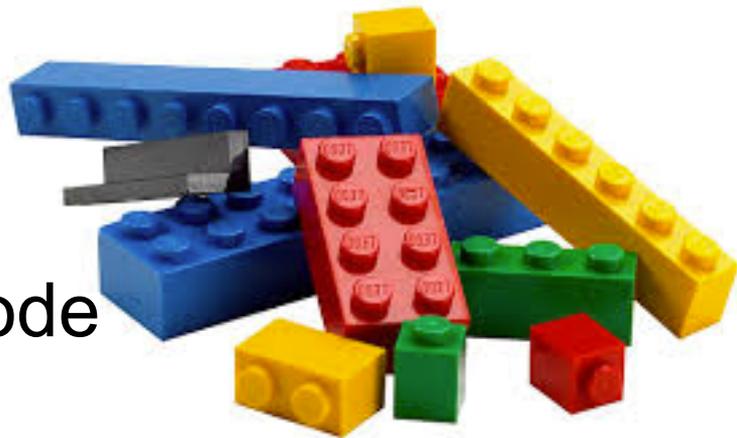
- ⌘ Identical Icon
- ⌘ Identical Bundle Name
- ⌘ Using Original Binary as a Host
- ⌘ Modified or additional functionality



ObjC Dylib injection

Dylib injection into memory

- ⌘ Overwrite functionality
- ⌘ No need for original source code
- ⌘ Shares memory
- ⌘ Shares iOS policy & entitlements



Cycript (@saurik)

Javascript & ObjC hybrid parser

- ⌘ Scriptable
- ⌘ SDK framework
- ⌘ Console based
- ⌘ Allows remote connection & manipulation of running app



cycript

Cycript allows developers to explore and modify running applications 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.

(It also runs standalone on Android and Linux and provides access to Java, but without injection.)

current version: 0.9.592

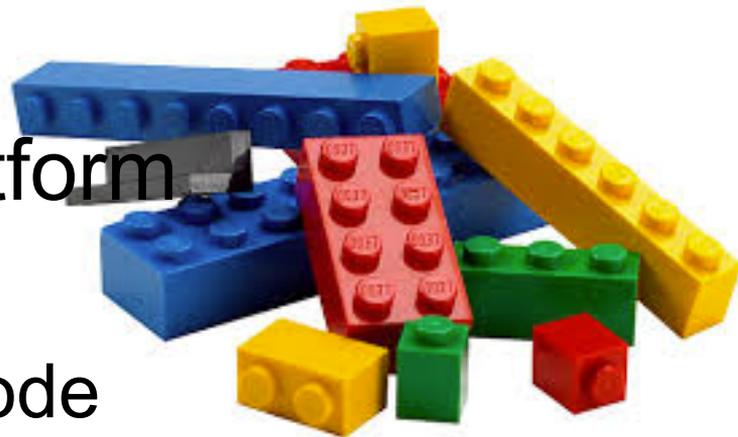
[Download SDK](#) [Read Manual](#)

www.cycript.org

Theos

An open source hooking platform

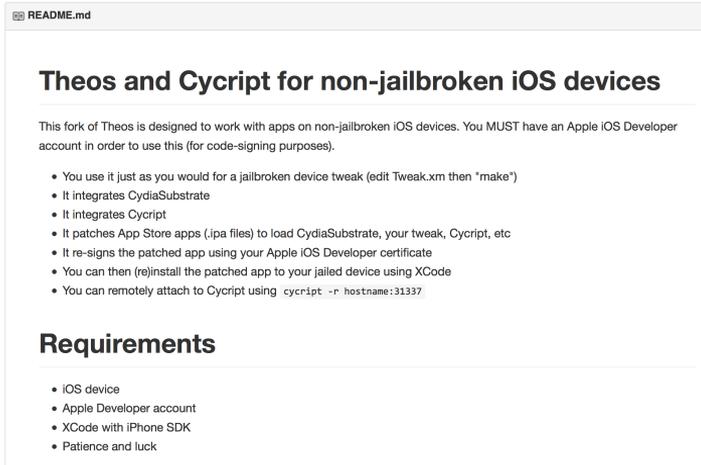
- ⌘ Create dylib out of templates
- ⌘ No need for original source code
- ⌘ Attaches to process via Cydia substrate injection



Theos-Jailed

A side project from Bishop Fox

- ⌘ Targeting non-Jailbroken devices
- ⌘ Only needs Dev account
- ⌘ Injects Dylib inside binary



<https://github.com/BishopFox/theos-jailed>

Theos-Jailed

- LC_LOAD_DYLIB (CoreGraphics)
- LC_LOAD_DYLIB (UIKit)
- LC_LOAD_DYLIB (Foundation)
- LC_LOAD_DYLIB (libobjc.A.dylib)
- LC_LOAD_DYLIB (libc++.1.dylib)
- LC_LOAD_DYLIB (libSystem.B.dylib)
- LC_LOAD_DYLIB (CoreFoundation)
- LC_LOAD_DYLIB (CoreVideo)
- LC_FUNCTION_STARTS
- LC_DATA_IN_CODE
- LC_CODE_SIGNATURE
- LC_LOAD_DYLIB (jaba.dylib)

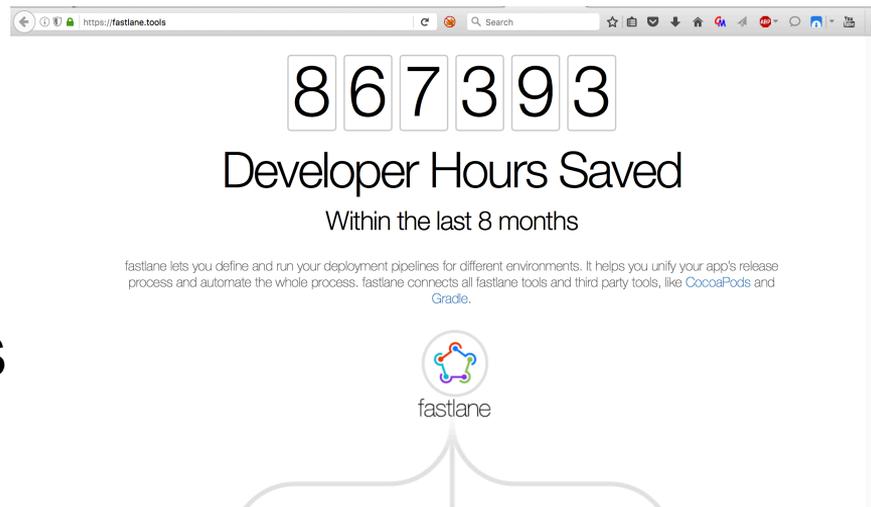
- Section (__TEXT,__text)
- Section (__TEXT,__picsymbolstub4)
- Section (__TEXT,__stub_helper)
- Section (__TEXT,__gcc_except_tab)

Offset	Data	Description	Value
000010F0	0000000C	Command	LC_LOAD_DYLIB
000010F4	00000038	Command Size	56
000010F8	00000018	Str Offset	24
000010FC	00000000	Time Stamp	Thu Jan 1 02:00:00 1970
00001100	00000000	Current Version	0.0.0
00001104	00000000	Compatibility Version	0.0.0
00001108	4065786563757461626C655...	Name	@executable_path/jaba.dylib

Fastlane.tools

Automation framework
for iOS/Mac Developers

- ⌘ Open-source
- ⌘ Interacts with developers portal
- ⌘ Ruby-based



<https://fastlane.tools/>



black hat[®]
ASIA 2016

Su-a-Cyber: Home-Brewed iOS Malware PoC Generator

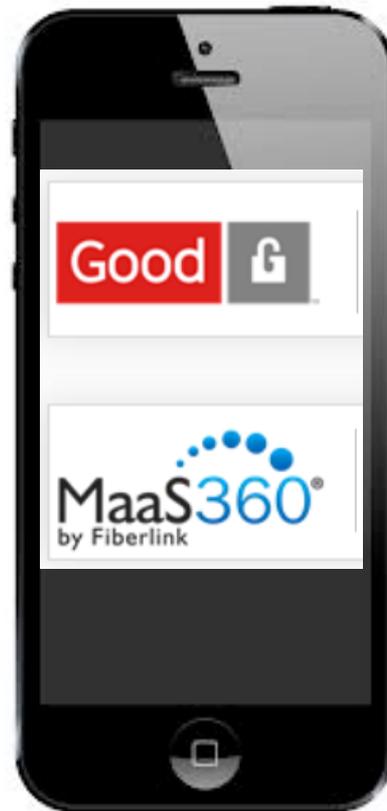


black hat[®]
ASIA 2016

Home-Brewed iOS Malware vs. the Corporate

Potential Corporate Targets

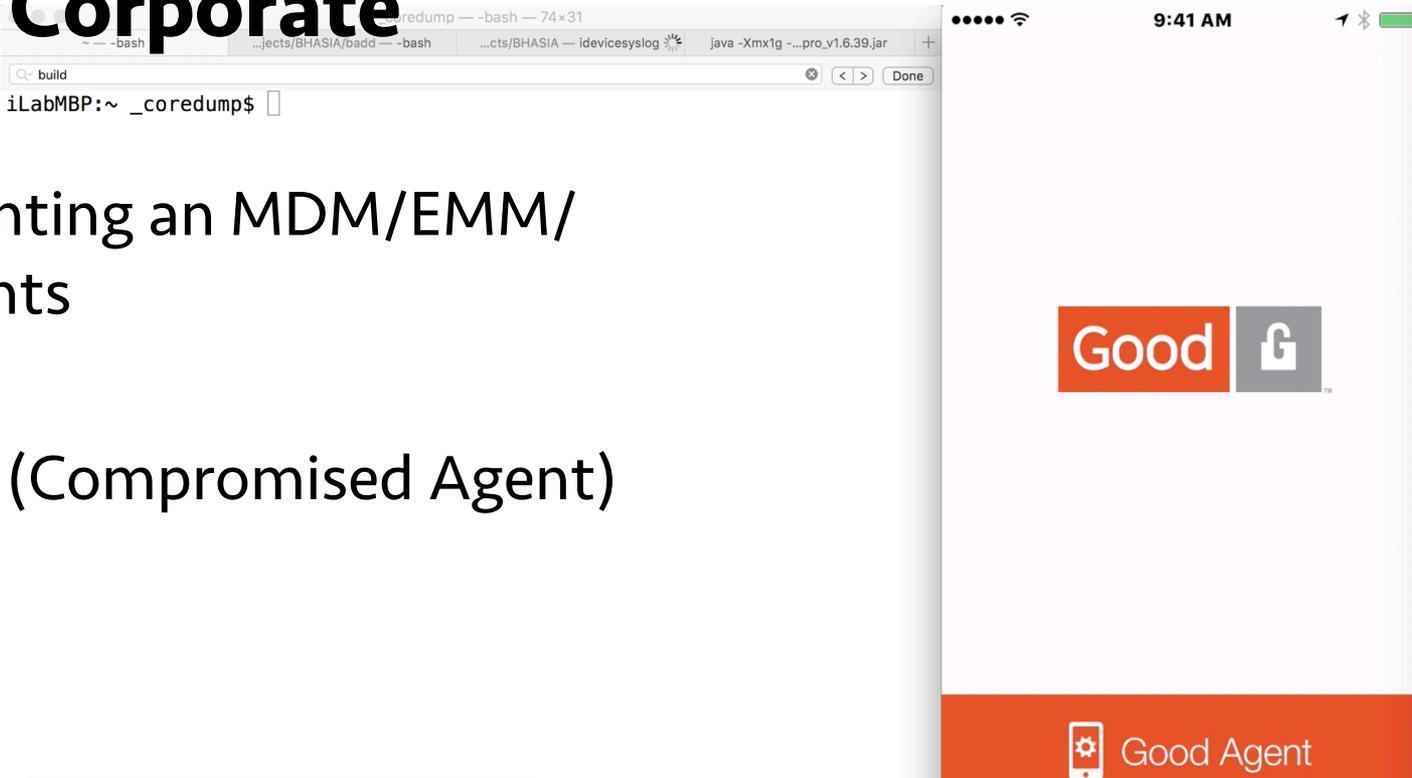
- ⌘ Circumventing MDM/EMM/MRM Clients
- ⌘ Circumventing internal stores (Emails, Docs, Apps)
- ⌘ Profit \$\$\$



Potential Corporate Targets

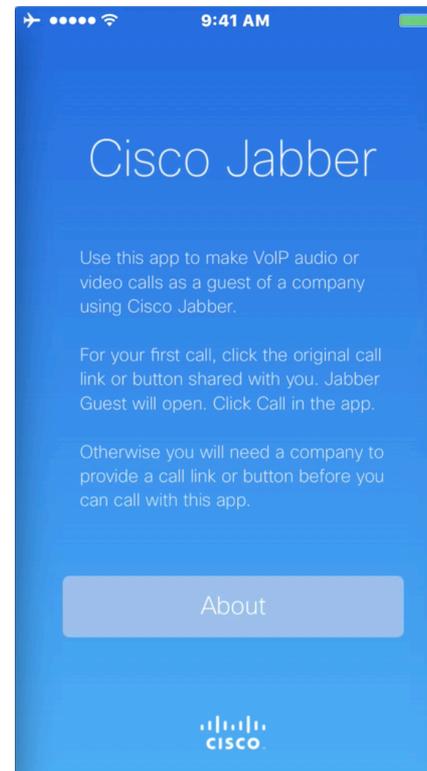
- ⌘ Circumventing an MDM/EMM/
MRM clients

Demo (Compromised Agent)



Potential Corporate Targets

- ⌘ Circumventing MDM/EMM/MRM Clients
- ⌘ VPN Clients, Corporate Messenger Applications

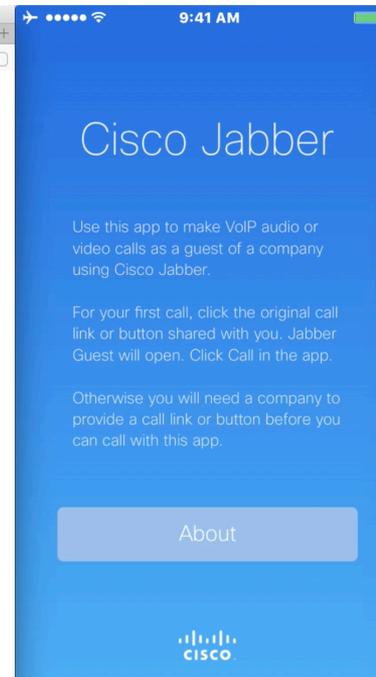


Potential Corporate Targets

- ⌘ Circumventing MDM/EMM/VRM Clients
- ⌘ VPN Clients, Corporate Messenger Applications

(Demo)

```
java -Xmx...1.6.39.jar idevicesyslog .../VPN/jaba --- -bash - Python >B?k?o?????h?????...
Q- vpn
#####
# Cydia & Theos tweaking system (@saurik & Dustin Howett)
#####
# libimobiledevice utilities (https://www.libimobiledevice.org)
#####
# Bisho-fox, theos-jailed (https://github.com/BishopFox/theos-jailed)
#####
# Asger Hautop Drewsen, insert_dylib (https://github.com/Tyilo/insert_dylib)
#####
# Spaceship, an Apple development automation platform (https://fastlane.tools/)
#####
#####
LICENSE:
#####
This tool is licensed under the GNU General Public License, version 3 (or later), available here:
http://www.gnu.org/licenses/gpl-3.0.html
#####
Projects created using Theos and/or Logos are not considered derivative works
(from a licensing standpoint, or, for that matter, any other standpoint) and
are, as such, not required to be licensed under the GNU GPL.
#####
The included project templates are license-free. The use of a template does
not confer a license to your project.
#####
#####
DISCLAIMER:
#####
This tool is an education tool for demonstration PoC of iOS Malware, Only!
#####
#####
!# Connect the Device to the USB port, and press any key to continue or ^C to abort....
#####
!# Running Make to Create the evil .dylib
#####
!# Running Make to Create the evil .dylib: Done
!# Preparing App Containers..
!# Preparing App Containers: Done
!# Preparing Application provisioning Profile..
#####
The login information you enter will be stored in your Mac OS Keychain
You can also pass the password using the "FASTLANE_PASSWORD" env variable
More information about it on GitHub: https://github.com/fastlane/credentials_manager
#####
Username: merryppines@gmail.com
#####
!# Preparing Application provisioning Profile: Done
!# Creating Evil Client with provisioning Profile..
!# Creating Evil Client with provisioning Profile: Done
!# Installing Evil Client provisioning Profile to USB device..
!# Installing Evil Client provisioning Profile to USB device: Done
!# Installing Evil Client to USB device.## Installing Evil Client to USB device: Done
!# PoC app is ready....
!#LAMB: jaba_coredump$
```



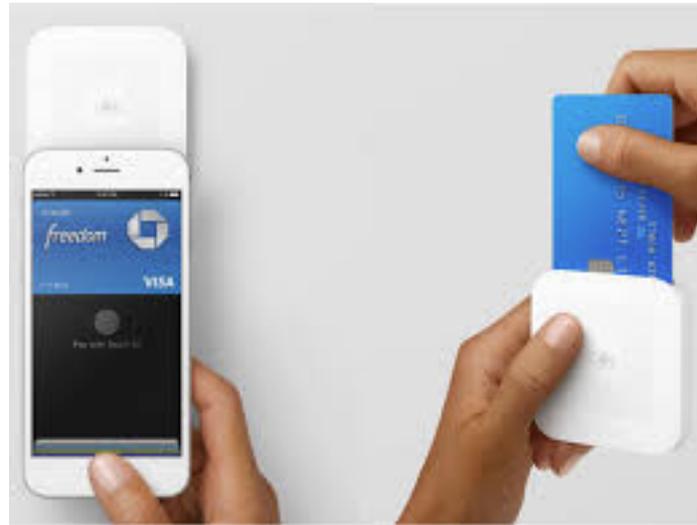
Potential Corporate Targets

- ⌘ Circumventing MDM/EMM/MRM Clients
- ⌘ VPN Clients, Corporate Messenger Applications
- ⌘ Healthcare



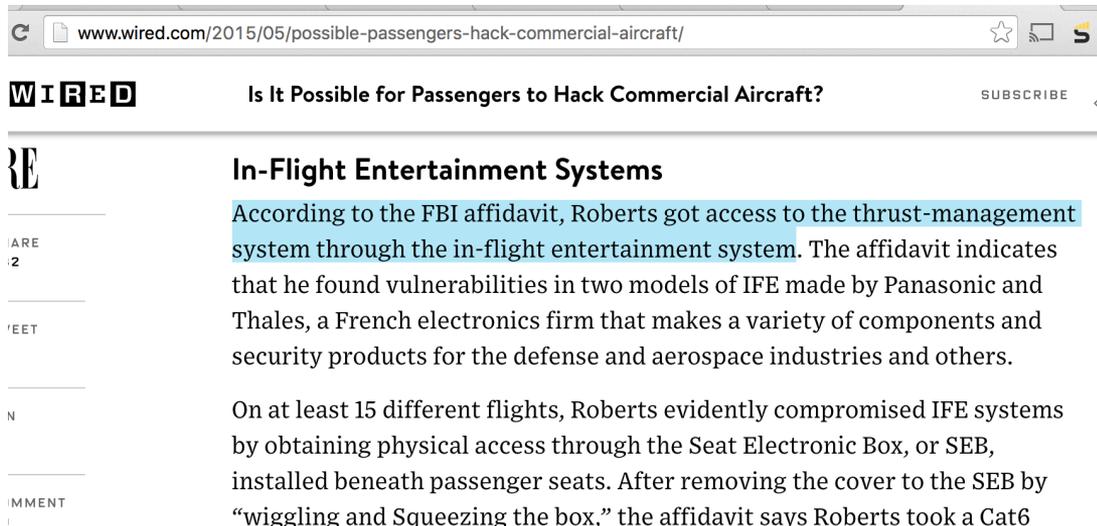
Potential Corporate Targets

- ⌘ Circumventing MDM/EMM/MRM Clients
- ⌘ VPN Clients, Corporate Messenger Applications
- ⌘ Healthcare
- ⌘ Finance Applications



Potential Corporate Targets

- ⌘ In Flight Entertainment Systems (Aviation Network)



www.wired.com/2015/05/possible-passengers-hack-commercial-aircraft/

WIRED Is It Possible for Passengers to Hack Commercial Aircraft? SUBSCRIBE

In-Flight Entertainment Systems

According to the FBI affidavit, Roberts got access to the **thrust-management system through the in-flight entertainment system**. The affidavit indicates that he found vulnerabilities in two models of IFE made by Panasonic and Thales, a French electronics firm that makes a variety of components and security products for the defense and aerospace industries and others.

On at least 15 different flights, Roberts evidently compromised IFE systems by obtaining physical access through the Seat Electronic Box, or SEB, installed beneath passenger seats. After removing the cover to the SEB by “wiggling and Squeezing the box,” the affidavit says Roberts took a Cat6

<http://www.wired.com/2015/05/possible-passengers-hack-commercial-aircraft/>

Potential Corporate Targets – The future...

Running Metasploit on a non-jailbroken corporate device?



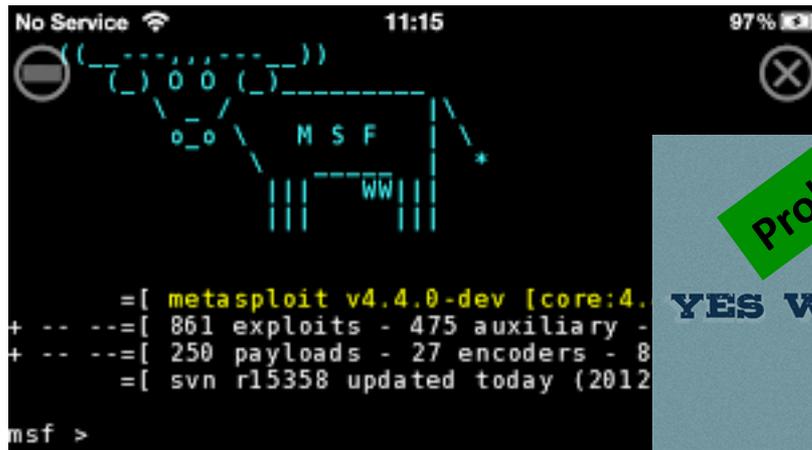
```
msf >

=[ metasploit v4.4.0-dev [core:4.4 api:1.0]
+ -- --=[ 861 exploits - 475 auxiliary - 144 post
+ -- --=[ 250 payloads - 27 encoders - 8 nops
=[ svn r15358 updated today (2012.05.31)

msf >
```

Potential Corporate Targets – The future...

Running Metasploit on a non-jailbroken corporate device?



A screenshot of a mobile terminal window. The status bar at the top shows "No Service", a Wi-Fi icon, the time "11:15", and "97%" battery. The terminal displays a Metasploit ASCII art logo with "MSF" and "WW" characters. Below the logo, the terminal shows the following output:

```
= [ metasploit v4.4.0-dev [core:4.
+ -- -- [ 861 exploits - 475 auxiliary -
+ -- -- [ 250 payloads - 27 encoders - 8
      = [ svn r15358 updated today (2012
msf >
```





black hat[®]
ASIA 2016

Home-Brewed iOS Malware Targeting the Individual

Personal Malware Capabilities

⌘ Pinpoint GPS Locationing – Abusage



```
- (void) locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)  
    CLLocation *location;  
    location = [manager location];  
    CLLocationCoordinate2D coordinate = [location coordinate];  
    _currentLocation = [[CLLocation alloc] initWith:  
        _currentLocation = newLocation;  
        _longitude = [NSString stringWithFormat:@"%f", coordinate.longitude];  
        _latitude = [NSString stringWithFormat:@"%f", coordinate.latitude];
```

Personal Malware Capabilities

⌘ Address-Book Stealing



```
case CNAuthorizationStatus.Authorized :
```

```
NSArray *keysToFetch = @[CNContactGivenNameKey, CNContactFamilyNameKey, CNContactPhoneNumbersKey];  
NSString *containerId = [self.CN_contacts defaultContainerIdentifier];  
NSPredicate *predicate = [CNContact predicateForContactsInContainerWithIdentifier:containerId];  
self.allContacts = [self.CN_contacts unifiedContactsMatchingPredicate:predicate keysToFetch:keysToFetch  
error:nil];
```

Personal Malware Capabilities

⌘ EXIF data extraction (GPS...)

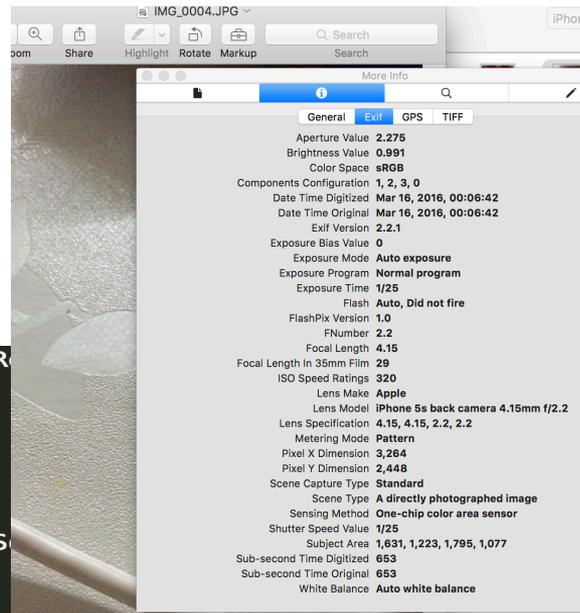
```
CGImageSourceRef source = CGImageSourceCreateWithURL((__bridge CFURLR

if (source == NULL) {
    NSLog(@"Source is NULL");
}

//get all the metadata in the image
NSMutableDictionary *metadata = (__bridge NSMutableDictionary *)CGImageS
, NULL);

//make the metadata dictionary mutable so we can add properties to it
NSMutableDictionary *metadataAsMutable = [metadata mutableCopy];

NSMutableDictionary *EXIFDictionary = [[metadataAsMutable objectForKey:(NSString *)
    kCGImagePropertyExifDictionary]mutableCopy];
NSMutableDictionary *GPSDictionary = [[metadataAsMutable objectForKey:(NSString *)
    kCGImagePropertyGPSDictionary]mutableCopy];
NSMutableDictionary *RAWDictionary = [[metadataAsMutable objectForKey:(NSString *)
    kCGImagePropertyRawDictionary]mutableCopy];
NSMutableDictionary *GIFDictionary = [[metadataAsMutable objectForKey:(NSString *)
    kCGImagePropertyGIFDictionary]mutableCopy];
```



Personal Malware Capabilities

⌘ Calendar Access

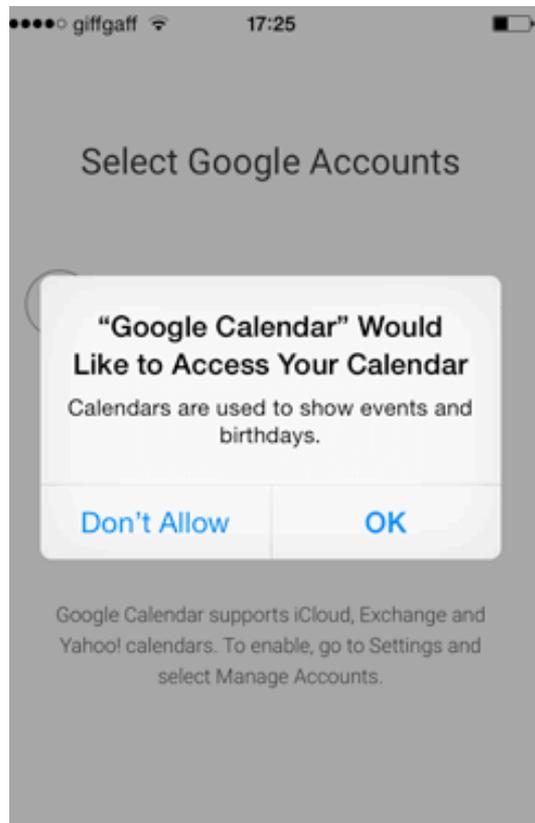
```
// Get the appropriate calendar
NSCalendar *calendar = [NSCalendar currentCalendar];

// Create the start date components
NSDateComponents *oneDayAgoComponents = [[NSDateComponents alloc] init];
oneDayAgoComponents.day = -1;
NSDate *oneDayAgo = [calendar dateByAddingComponents:oneDayAgoComponents
                                                    toDate:[NSDate date]
                                                    options:0];

// Create the end date components
NSDateComponents *oneYearFromNowComponents = [[NSDateComponents alloc] init];
oneYearFromNowComponents.year = 1;
NSDate *oneYearFromNow = [calendar dateByAddingComponents:oneYearFromNowComponents
                                                         toDate:[NSDate date]
                                                         options:0];

// Create the predicate from the event store's instance method
NSPredicate *predicate = [store predicateForEventsWithStartDate:oneDayAgo
                                                         endDate:oneYearFromNow
                                                         calendars:nil];

// Fetch all events that match the predicate
NSArray *events = [store eventsMatchingPredicate:predicate];
```



Personal Malware Capabilities

⌘ Health Kit Access

HealthKit Framework Reference

Search iOS Developer Library

In addition, your app must not access the HealthKit APIs unless the app is primarily designed to provide health or fitness services. Your app's role as a health and fitness service must be clear in both your marketing text and your user interface. Specifically, the following guidelines apply to all HealthKit apps.

- Your app may not use information gained through the use of the HealthKit framework for advertising or similar services. Note that you may still serve advertising in an app that uses the HealthKit framework, but you cannot use data from the HealthKit store to serve ads.
- You must not disclose any information gained through HealthKit to a third party without express permission from the user. Even with permission, you can only share information to a third party if they are also providing a health or fitness service to the user.
- You cannot sell information gained through HealthKit to advertising platforms, data brokers or information resellers.
- If the user consents, you may share his or her HealthKit data with a third party for medical research.
- You must clearly disclose to the user how you and your app will use their HealthKit data.

You must also provide a privacy policy for any app that uses the HealthKit framework. You can find guidance on creating a privacy policy at the following sites:

- Personal Health Record model (for non-HIPAA apps): <http://www.healthit.gov/policy-researchers-implementers/personal-health-record-phr-model-privacy-notice>
- HIPAA model (for HIPAA covered apps): <http://www.hhs.gov/ocr/privacy/hipaa/modelnotices.html>

These models developed by the ONC are designed to improve user experience and understanding by using plain language and approachable designs to explain how user data is collected and shared. These are not intended to replace a web based privacy policy, and developers should consult ONC guidance regarding which model is appropriate for a given app. These models are provided for your reference only, and Apple expressly disclaims all liability for your use of such models.

https://developer.apple.com/library/ios/documentation/HealthKit/Reference/HealthKit_Framework/

Personal Malware Capabilities

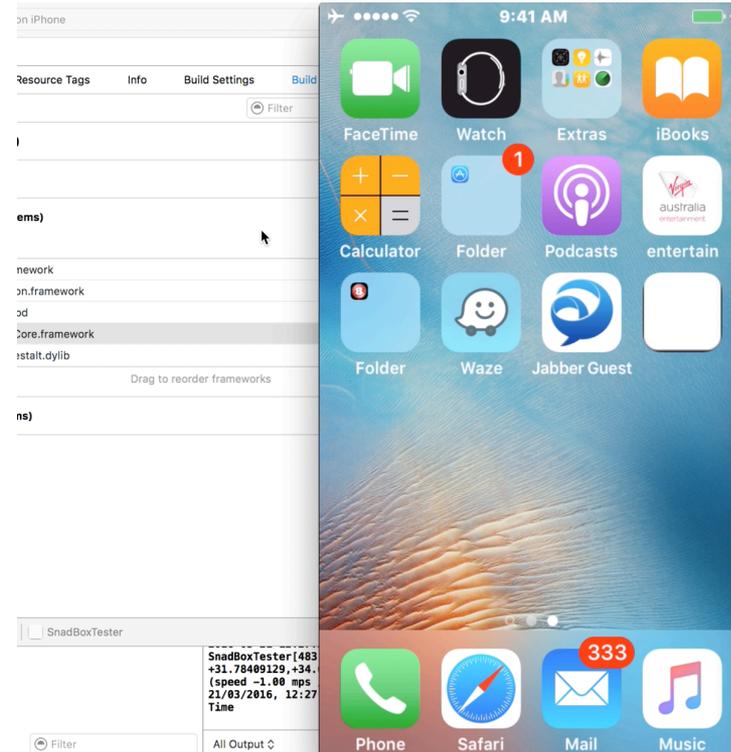
⌘ Evil Skype Demo

The image displays a terminal window on the left and a screenshot of the Skype app on the right. The terminal window shows the execution of the 'evilskype' tool, which is a proof-of-concept for iOS malware. The terminal output includes a license notice, a disclaimer, and instructions for connecting the device to the USB port. The tool then proceeds to create an evil .dylib, prepare app containers, and install the evil client provisioning profile to the USB device. The terminal output also shows the installation of the evil client to the USB device and the execution of the 'evilskype' tool. The Skype app screenshot on the right shows the 'Sign in' screen with the text 'Skype is now controlled by Attacker' and a blue arrow button.

```
java -X..6.39.jar idevicesyslog ...acker --- -bash ..skype --- -bash lrb rvm...vmlbin ... cypri...270.0.1 +
# General Public License, version 3 (or later), available here:
# http://www.gnu.org/licenses/gpl-3.0.html.
# Projects created using Theos and/or Logos are not considered derivative works
# (from a licensing standpoint, or, for that matter, any other standpoint) and
# are, as such, not required to be licensed under the GNU GPL.
# The included project templates are license-free. The use of a template does
# not confer a license to your project.
#
# DISCLAIMER:
# This tool is an education tool for demonstration PoC of iOS Malware, Only!
#
# # Connect the Device to the USB port, and press any key to continue or ^C to abort....
# # Running Make to Create the evil .dylib
# # Running Make to Create the evil .dylib: Done
# # Preparing App Containers..
# # Preparing App Containers: Done
# # Preparing Application provisioning Profile..
#
# The login information you enter will be stored in your Mac OS Keychain
# (you can also pass the password using the 'FASTLANE_PASSWORD' env variable
# More information about it on GitHub: https://github.com/fastlane/credentials_manager
#
# Jsername: merryppines@gmail.com
# # Preparing Application provisioning Profile: Done
# # Creating Evil Client with provisioning Profile..
# # Creating Evil Client with provisioning Profile: Done
# # Installing Evil Client provisioning Profile to USB device..
# # Installing Evil Client to USB device..# Installing Evil Client to USB device: Done
# # PoC app is ready...
# # LabMMP:evilskype_coredump# cypri -r 127.0.0.1:31337
# # UIApp
# # SKPApplication: 0x16c19de0#
# # UIApp.keyWindow.firstResponder
# # UITextField: 0x16e1dc00; frame = (40 184.5; 240 24); text = ''; clinoToBounds = YES; opaque = NO; onstureRecogin
# # zers = <NSArray: 0x16f8c45b>; layer = <CA
# # UIApp.keyWindow.firstResponder.text=
# # Skype is now controlled by Attacker!
# #
```

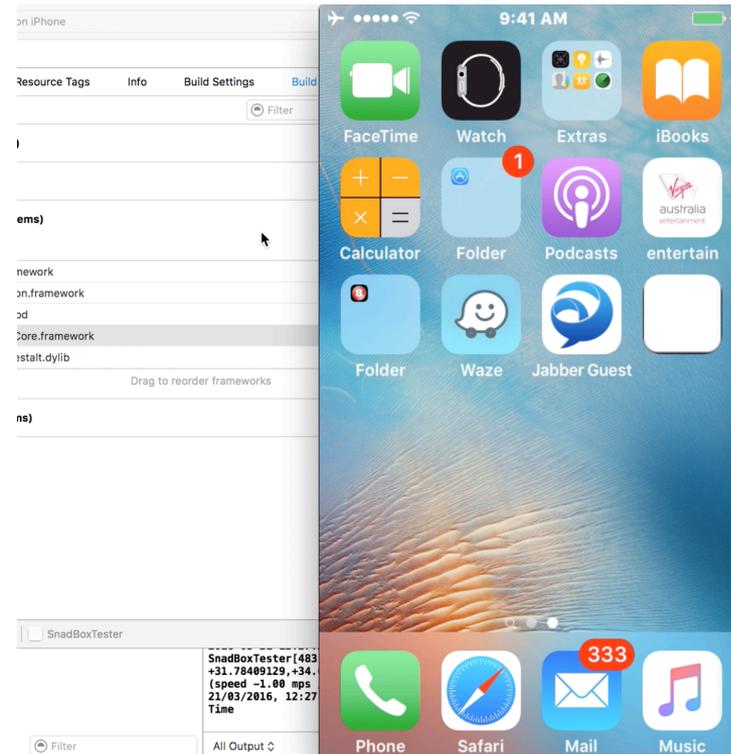
Hiding Personal Malware

- ⌘ Invisible Icon + No Bundle Executable name + UI glitch



Hiding Personal Malware

- ⌘ Invisible Icon + No Bundle Executable name + UI glitch
- ⌘ No Taskbar lookup (no name)



Hiding Personal Malware

- ⌘ Invisible Icon + No Bundle
Executable name + UI glitch
- ⌘ No Taskbar persistency
- ⌘ Abusing CFURLs



Hiding Personal Malware

⌘ Abusing CFURLs

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)
  launchOptions {
    // Override point for customization after application launch.
    [self PostIt];
    [self recorder];
    [NSTimer scheduledTimerWithTimeInterval:1.1
                                     target:self
                                     selector:@selector(targetMethod)
                                     userInfo:nil
                                     repeats:NO];

    return YES;
}

-(void) targetMethod{
    // Call Here ...
    NSURL *actionURL = [NSURL URLWithString:@"skype://" ];
    [[UIApplication sharedApplication] openURL:actionURL];

    //Invalidate the time
    [myTimer invalidate];
    myTimer = nil;
}
```

Hiding Personal Malware

Abusing CFURLs – Demo (Evil Skype Launching)

Detection & Mitigation

Personal Risk

Passcode Lock

Preferences – Management

Battery Drain

Weird Icons

Unknown Bundles

Corporate Risk

App Profiling

Brand Protection

Network Awareness

Responsive MDM/EMM



black hat[®]
ASIA 2016

Questions & Answers

Other Resources

Claud Xiao, Palo-Alto Networks <http://researchcenter.paloaltonetworks.com/author/claud-xiao/>

Cycript & Cydia (www.cycript.org) @saurik

Theos tweaking system (Dustin Howett)

libimobiledevice utilities (<https://www.libimobiledevice.org>)

Bishop-fox, theos-jailed (<https://github.com/BishopFox/theos-jailed>)

Asger Hautop Drewsen, insert_dylib (https://github.com/Tyilo/insert_dylib)

Fastlane & Spaceship, an Apple development automation platform (<https://fastlane.tools/>)



black hat[®]
ASIA 2016

Mi³ Security 

Machine - Meta - Mobile
Intelligence for the next billion apps

Su-a-Cyder: Home-Brewing iOS Malware Like a BO\$\$!

Chilik Tamir
chilik@mi3security.com
Twitter: @_coreDump