Asia Pacific College
School of Computer Science and Information Technology

COMSEC2 PROJECT DOCUMENTATION

iOS JAILBREAK

OCHOTORENA, LEIRRAND CHRISTIAN A.
CURATO, KENT WENDELL
GONZALES, ELOI DENICE
PATANAO, EARL JOSHUA

JUSTIN PINEDA
Computer Security 2 Teacher

December 18, 2015

Background

Apple's ecosystem is follows a closed-source, and proprietary format for all of its devices. For the applications that it distributes, first-party and third-party apps are pushed towards their AppStore which can be accessed via their different product lines. One of its most significant product lines include the iOS devices which make use of its mobile computing platform, these devices include the iPhones, iPod, iPads, Apple TV and Apple Watch.

It also has its own media manager in the form of iTunes which serves as its backup, update, recovery, multimedia content organizer and digital rights management module (where its apps are published in the AppStore and music and videos in the iTunes Store or Apple Music).

The platform imposed by Apple to its consumer devices limits the freedom of customization that the user can do since it implemented stricter controls on its iOS platform (how it manages disk space, what privileges its grants its users and etc.). The problem is that the users would definitely want to make the device their own by being able to customize its interface, optimize its performance, run unsigned applications and add support for popular multimedia formats that Apple does not natively support. This then ushered in the need towards the process of Jailbreaking.

There can be a lot of definitions for Jailbreaking, but basically, jailbreaking is a process of privilege escalation in iOS devices where restrictions are removed, thus, allowing the user to have a root access on the device's configuration and system. Reasons why jailbreaking is done is because of the added customizations that can be made on it not limited to the installation of non-AppStore applications (i.e. including software piracy) but also to the unlocking of iOS devices for use with multiple carriers.

Since the release of the first iPhone, a lot of changes in the process of jailbreaking process where made wherein in the past it included multiple steps but now it is as simple as tapping a single button once the device has been connected to a computer containing the jailbreak tool.

Jailbreak Process

Key terms in jailbreaking includes the following:

- **Cydia** = the application that installs the tweaks
- **jailbreak tool** = any tool used to facilitate the insertion of code to exploit the iOS (system) thus allowing root access for the user.
- **repository** = storage of any tweaks, hacks or tools hosted on a server
- **Signing** = ipsw verification step of Apple
- **.deb** = packages that can be installed on the device which are hosted on a repository.
- **.ipa** = the file extension for the "cracked" applications.
- **.ipsw** = the file extension for the iOS device firmware. Hosted on a lot of warez sites.

In order to be able to determine the steps to be done during the Jailbreak process, it must also be considered as to whether what device does the user has, what firmware is still being signed by Apple, does the jailbreak tool work on the user's current device and what Computer Operating System does the user currently use.

Different iOS firmware entails the use of different jailbreak tools provided by different development teams such as TaiG and Pangu.
In the history of jailbreaking, there are two ways on how it can be done (steps vary based on iOS device type and firmware). Before it was a multi-step process, whereas now it became just a one-step process. In both processes, a device backup via iTunes is suggested so as to save the user's content.

MULTI-STEP JAILBREAKING PROCESS

1. Select the appropriate tool (QuickPwn, PwnageTool,Snowbreeze, etc.) based on the iOS device and iOS version.
2. Plug in the device on the computer and wait for it to be recognized as an Apple device.
3. Put device into DFU mode (Not recovery mode).
4. Select the correct .ipsw.
5. Wait for the process to complete itself then follow on-screen prompts on the device.
6. Tap on Cydia, install the needed tweaks.

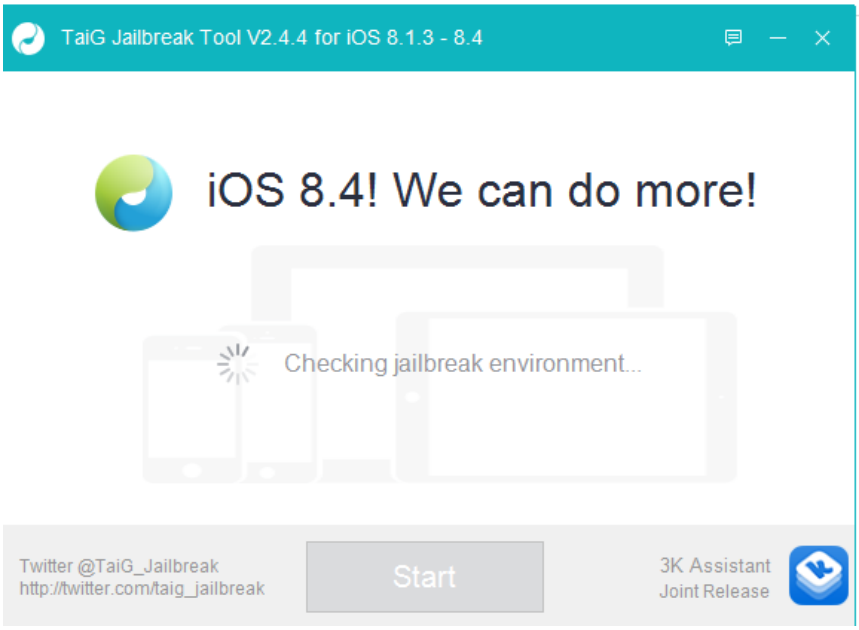
(Misc: Do a backup before jailbreaking)

ONE-CLICK JAILBREAKING PROCESS

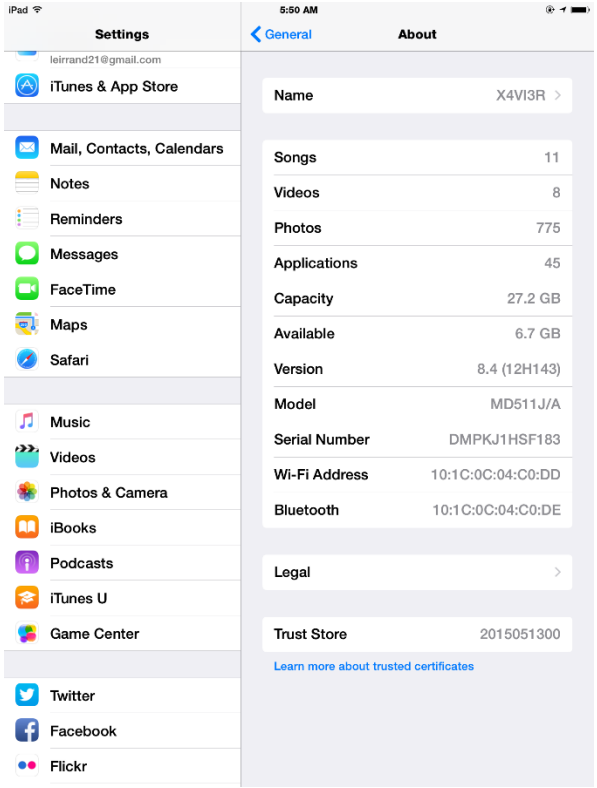
1. Plug-in the device on the computer.
2. Click on start button on the jailbreak tool, wait for the process to finish.
3. Tap on Cydia, install needed tweaks.

(Misc: Do a backup before jailbreaking)

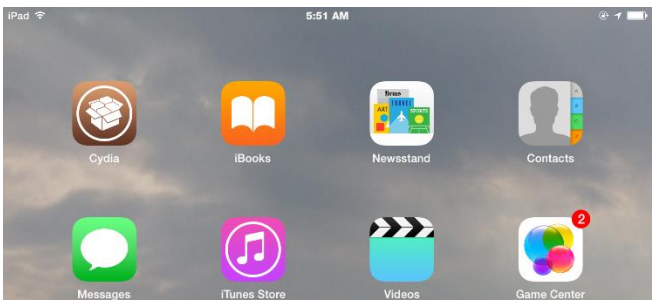
SAMPLE SCREEN FOR iPAD 4 IOS 8.4 JAILBREAK



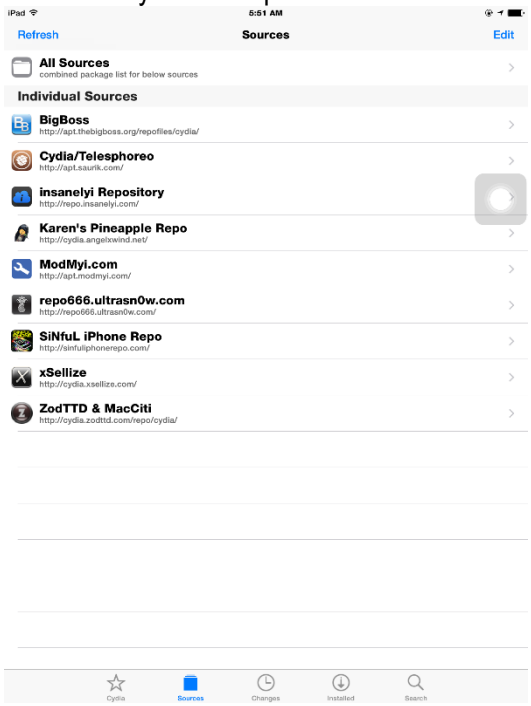
TaiG Jailbreak Tool for iOS 8.4 (Once the device is recognized, just tap on the start button).



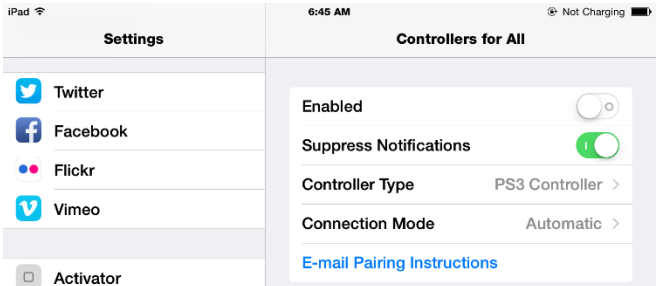
iOS SETTINGS on iPad 4 showing that the device currently has iOS 8.4 installed



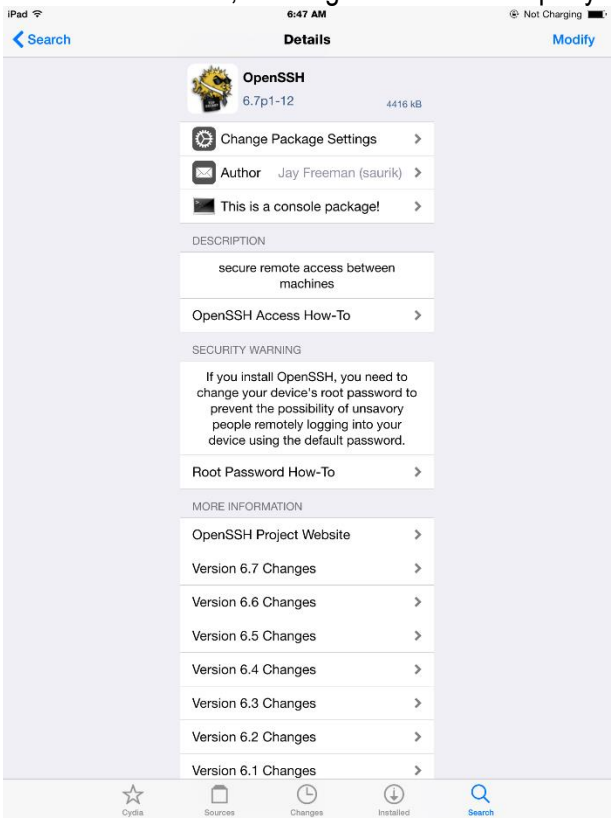
iOS Homescreen shows the Cydia icon upon successful installation of the jailbreak



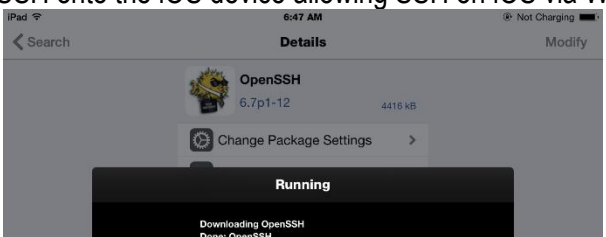
Cydia application allows for the installation of unsigned code distributed via various repositories



Sample Cydia Application Controllers for All, allowing the use of Third-party hardware onto iOS Apps



Installation of OpenSSH onto the iOS device allowing SSH on iOS via WinSCP or other tools



Installation of OpenSSH debian package

DEMONSTRATED CAPABILITIES

The demonstration done in class showed the jailbreak process, manipulation of iOS files via OpenSSH and WinSCP (Executable program on Windows), installation of cracked .ipa files via AppSync (plugin available on Cydia) and controlling the iPad via a PS3 controller using Controllers4All (plugin) and SixAxisPairTool (Executable program on Windows).

The demonstrated capabilities mean that the control over the device is substantially improved since a lot of its filesystem can be accessed by the user to allow for the installation of tweaks and other unsigned modules or code.

AFFECTED SYSTEMS

Any iOS device running on iOS 8.4. For other jailbreaks, iOS 9.0.2 is provided by Pangu.

SPECIFIC EXPLOITS

- DeveloperDiskImage race condition (also used in TaiG for 8.0-8.1.2 but modified) - to mount a fake DDI
- enable-dylibs-to-override-cache - force loading of dynamic libraries from filesystem (where available) instead of the shared cache (overriding libmis)
- Symbolic linking to AFC (CVE-2015-5746)
- Backup exploit to write to protected regions of the disk (CVE-2015-5752)

- Code signing exploit (CVE-2015-3802)
- Code signing exploit (CVE-2015-3803)
- Code signing exploit (CVE-2015-3805)
- Code signing exploit (CVE-2015-3806)
- IOHIDFamily exploit (CVE-2015-5774)
- Air Traffic exploit to allow attackers to access arbitrary filesystem locations via vectors related to asset handling (CVE-2015-5766)

SOLUTIONS

If the user doesn't want to be left vulnerable to the exploits in iOS 8.4, a device firmware update can be done.

CURRENT STATUS

The iOS 9.0.2 is the most recent jailbreakable iOS firmware with iOS 9.2 being the most recent non-jailbreakable firmware being signed by Apple.

Note that not all tweaks are compatible with the most recent version of the jailbreakable iOS, thus, jailbreak users are usually discouraged in jumping into a newer iOS firmware version.

CONCLUSION

Jailbreaking can be a breath of fresh air for those wanting to have a lot of control and customization on their iOS devices, however, such can be a double-edged sword for them since that it also opens up to exploits that are left open by the escalation of user privilege. For those wanting to have better customization options, jailbreaking is warranted provided that the user has basic understanding about the jailbreak process, tweaks and customizations. He or she must also be aware how to be able to close up any open ports on the device so as to not fall victim into hacks initiated by malicious elements.

For the users who want a hassle-free iOS experience, updating to the latest iOS version is almost always recommended to be able to automatically patch any vulnerabilities found on the older iOS firmware.

References:

http://www.cultofmac.com/192850/the-history-of-jailbreaking-feature/
https://en.wikipedia.org/wiki/IOS_jailbreaking
http://www.redmondpie.com/history-of-jailbreaking-iphone-with-saurik-the-creator-of-cydia-video/
https://www.theiphonewiki.com/wiki/Jailbreak
http://www.cheatsheet.com/gear-style/gadgets/apples-product-timeline-the-best-of-the-best.html/?a=viewall
http://www.engadget.com/2006/04/01/30-years-in-apple-products-the-good-the-bad-and-the-ugly/
http://en.pangu.io/
http://www.redmondpie.com/ios-9.1-jailbreak-status-update/