CHAMPLAIN COLLEGE | LCDI Leahy Center for
Digital Investigation

iOS 9 Jailbreak

175 Lakeside Ave, Room 300A
Phone: (802)865-5744
Fax: (802)865-6446
http://www.lcdi.champlain.edu

05/01/2016

Disclaimer:

This document contains information based on research that has been gathered by employee(s) of The Senator Patrick Leahy Center for Digital Investigation (LCDI). The data contained in this project is submitted voluntarily and is unaudited. Every effort has been made by LCDI to assure the accuracy and reliability of the data contained in this report. However, LCDI nor any of our employees make no representation, warranty or guarantee in connection with this report and hereby expressly disclaims any liability or responsibility for loss or damage resulting from use of this data. Information in this report can be downloaded and redistributed by any person or persons. Any redistribution must maintain the LCDI logo and any references from this report must be properly annotated.

Contents

Introduction

Jailbreaking is an increasingly popular method of retaking control of your device from the manufacturer. Apple phones in particular are well-known for having a highly secure platform, which makes jailbreaking their phone an attractive option for many consumers. For this project, we wanted to gain a better understanding of the jailbreaking process while researching what kind of data is available for extraction from a phone before and after a jailbreak. We believe that this research could help further both law enforcement's and corporate digital forensics personnel's understanding of jailbreaking iOS and Apple products as a whole, while also aiding in future research.

Background:

Jailbreaking is a long-standing process that is just now beginning to become a widely-used practice with iPhone users. It removes Apple's application signing and allows any third party applications to be installed and run, both well-meaning and not. Apple's current operating system, as of publication, is 9.2.1 with iOS 9.3 being released on April 2, 2016. The last jailbreak created by a third-party platform was for iOS 9.0- 9.0.2; since then, the jailbreak community has waited for a current jailbreak with the updated systems to be released by one of the research teams, the most prominent of which being taig9 and Pengu. Apple's operating system is heavily controlled which makes the possibility of a jailbreak attractive to many consumers; however, Apple is constantly updating their iOS versions and tightening any back-door access that allows jailbreaks to be created. Prior to this project, the LCDI had not done substantial research on the subject of jailbreaking iPhones or jailbreaking in general.

Purpose and Scope:

The purpose of this report is to offer an insight into how jailbreaking works, what type of data becomes available during forensics analysis both prior to a jailbreak and after, and finally to offer a better understanding of the iOS file structure to aid investigators and future researchers.

Research Questions:

1) What data is accessible from a non-jailbroken phone?
2) What data is accessible from a jailbroken phone?
3) How does jailbreaking elevate to root, i.e. how does the exploit work?

Terminology:

Jailbreaking – An act of privilege escalation on a device running Apple's mobile operating system, iOS. Jailbreaking removes the restrictions and limitations that are implemented by default in iOS, granting root level access to the file system. This allows for the download and installment of applications that are not native to the official App Store, third party extensions, and customization options such as themes for the OS, which are also not available on the App Store.

Artifacts – Any data generated by user interaction that can be collected and examined. Any user data retrieved from the browser is considered an artifact, including cookies, caches, geolocation, search history, etc.

UFED or **Cellebrite –** One of the most frequently used forensics tools, used to extract data from mobile devices. Cellebrite gives the user the option to go through the system files file by file in order to obtain file paths that may not be available otherwise.

Image – Refers to a copy of a hard drive, or disk image, which is compressed into a series of files. Physical images include all information (zeroes and ones) on the hard drive whether the space is being used or not, and is close to the same size as the actual hard drive itself. As opposed to a physical image, a logical image only acquires the parts of the hard drive that have active data and dismisses the rest of the drive. Compared to a physical image, the size can be extremely small or the same size as the drive depending on the amount of data stored.

Root – The highest level a user can be elevated to. Root is capable of running any commands. This is highly sought after in jailbreaking, as it allows the user to remove the application signing enabled by Apple.

Methodology and Methods

For this project we used an Apple iPhone 6 running the latest version of iOS 9 (9.2.1). We chose this as we thought it was the most applicable to the field and would offer us the most interesting project as it was the newest software on the newest hardware. This meant that there was not nearly as much research done previously, leaving room for us to set a precedent and do our own studying.

We generated all our data on this phone manually and used Cellebrite and XRY to create images of the phone. The data generation was created by performing simple user actions you might expect to see on a normal iPhone so as to try and replicate standard data. After a few days of generating actions, we used the software Cellebrite to pull an image of the unlocked phone, then one with it locked and we compared the two images we now had.

We also looked at the phone image through Magnet Forensics' Internet Evidence Finder (IEF) tool. This allowed us to index the image and subsequently search it more quickly for specific words or phrases. This streamlined the process of going through our images and made comparing the data pulled much easier when looking at our control data.

We looked at the phone prior to jailbreaking, and unfortunately only after a semi-jailbreak. We acquired the phone and started with the data generation, which is elaborated upon in the Data Collection: section below. After generating our data, we imaged the phone first using Cellebrite and then using XRY. The image was then looked at through IEF. The iOS 9.2.1 beta jailbreak was downloaded; however, we found that we were not given full root access. While waiting for a new jailbreak version to come out, we compared the data we extracted, but at the time of publication, iOS 9.3 was set to be released April 2nd with no new jailbreaks.

Equipment Used:

Table 1: Hardware

Device	OS Version
Apple iPhone 6	iOS 9.2.1

Table 2: Software

Software	Version	Comments
TAIG9- Jailbreak tool	BETA for iOS 9.2.1	This is a jailbreak tool found at http://taig9.com/beta/ and doesn't provide root access to the phone as a beta version.
UFED or Cellebrite	V4.5.1	Imaging Tool
XRY	V6.16.6	Imaging Tool
Windows 7	Research Workstation	Used to analyze image data
Internet Evidence Finder (IEF)	V6.7.5.1029	Forensic tool used to look through iPhone data

Data Collection:

We began by generating data on the phone, ranging from text messages to emails to web browser activity, and part of our recorded activity can be seen in **Error! Reference source not found.**4. The documented data was used to not only keep track of what we did, but also for us to compare our findings to the generated data. Please refer to the appendix for the rest of the table. We compared each image we created of the phone to the data we recorded generating, and doing this allowed us to find any discrepancies in the data. We compared the data by looking at each image to see what data was provided by each and to see if different software provided different results.

Analysis

At the beginning of the project, we had the goal of jailbreaking the device ourselves and subsequently analyzing how many privileges you have as root on the phone since Apple is well-known for limiting even root's ability to traverse and write to directories in an effort to increase device security. We quickly discovered that we did not have the skills necessary in order to jailbreak the phone ourselves and it would take a significant amount of time to learn, so we decided to use a publicly available partial jailbreak. We simply did not have the necessary knowledge to discover flaws in boot kernels, nor did we have experience scouring the GNU kernel in order to completely jailbreak the device. Once a flaw is found in the kernel, jailbreakers leverage it to gain access to parts of memory they're not normally able to, and subsequently abuse these memory holes to leverage root access. The partial jailbreak we used didn't remove the application signing, meaning we could install third party applications, but were not able to run them due to the application signing.

We shifted our focus from jailbreaking the phone to doing a forensic examination of the phone both locked and unlocked, as well as non-jailbroken and partially jailbroken. Our first pull was from the phone in an unlocked state. This allowed us to establish a control to compare the rest of the data pulls to. This was imperative in allowing us to conduct as thorough an investigation as possible. By comparing the subsequent pulls to the control, we can establish just how much data each phone state provides. We used the same method of extracting data for each test, which was a full logical file extraction. This level of continuity throughout the investigation allowed us to have consistent, reliable results.

As could be expected, the locked iPhone image provided the least amount of readable data, only file names and dates created, while the extraction from the other unlocked images contained a significant amount more data. The image we pulled of the partially jailbroken device gave us a better insight into the actual process that the partial jailbreak used to install its own applications and related downloadable content. From the data pull, we were able to determine that the partial jailbreak was not granting root access and instead just installing a new profile containing proprietary third party applications. This is why the applications were not runnable.

Results

The table below provides a general overview of what data was viewable after each extraction; unfortunately, we were not able to find a full jailbreak in time for the project, so we decided to use a small tool that provided similar capabilities but could not root the device. Instead, it essentially installed was a third party application store that wasn't able to run applications installed through it because of application signing enabled by Apple. The iPhone with its password off was our first pull, which provided all the data normally viewable on the phone and acted as a control for us. The next image was a partial file extraction of the iPhone with its password enabled. This was a much more limited extraction and provided us with only the file names and dates created/modified. This was to be expected as the rest of the data was encrypted using iOS's proprietary AES-256 encryption. The last image was of a partially jailbroken device. This provided the same amount of data as the control, which suggests that the partial jailbreak wasn't able to provide us root access to iOS. This also explains why we weren't able to run the applications we downloaded through the third party store.

Table 3: Results at a glance

OS state	Encryption	Amount of data found
Non jailbroken	Off	Full file extraction Control for experiment
Non jailbroken	On	Limited file extraction Much less content than control
Partially Jailbroken	Off	Full file extraction Same as control

Conclusion

We discovered that there are significant differences in accessible artifacts when imaging an iPhone that is locked, unlocked, and semi-jailbroken. Even though we did not accomplish all of our goals for the project that doesn’t mean we were unsuccessful. Since we were not able to successfully jailbreak our device we were not able to determine exactly what artifacts were unique to the jailbroken device. We were not able to elevate to full root access because of Apple restrictions, but through our research and efforts we were able to figure out how the restrictions limited our jailbreaking capabilities.

Upon the completion of this project, we were able to thoroughly document where data is natively stored on an unmodified device. The knowledge where to look for key pieces of data can help law enforcement and other security firms locate imperative evidence during investigations. While we researched iOS, this information can be utilized when looking at other operating systems because they often have commonalities in where certain types of data are stored, which files are related to each other, and what type of data is generated by certain actions.

Further Work

The research conducted in the project will be extremely useful in future projects and will allow for growth in this specific field of research. After exploring how to remove files from locked and encrypted devices, it became increasingly obvious that there were a multitude of ways to accomplish this. Future work involving the iOS file structure could potentially involve doing NAND chip analysis, leveraging a custom iOS jailbreak against devices, and writing custom firmware updates for IP boxes to make the process of brute forcing locked devices much simpler and safer.

Appendix

Table 4: Data Gen Notes

Time	User Action	Machine Action	Comments
2/09/16			
12:50 PM	Installed 9.2.1 Software	Installed newest iOS	Had to update to use Jailbreak
1:07 PM	Went to the webpage *taig9.com/beta/*	safari	
1:07 PM	Started jailbreak download	Jailbroke phone	Did the beta jailbreak- can be removed
1:11 PM	Removed the jailbreak profile	Removed the jailbreak profile	
1:12 PM	Rebooted Device	Rebooted	
2/11/16			
1:51 PM	Went to TaiG beta (taig9.com)	Safari used	
1:52 PM	Downloaded the beta jailbreak	Jailbroke the device	
1:54 PM	Installed iOS Emus		
1:57 PM	Rebooted	Rebooted	
2:25 PM	Turned off Find my Iphone		
2/16/16			
11:33 AM	Downloaded uncle sam image	safari	
11:34 AM	Set uncle sam as background	settings	
11:40 AM	Browsed to Taig9	safari	
11:41 AM	Downloaded partial JB		
11:45 AM	Installed 3K assistant	profiles	
12:31 PM	Took photo(s) & videos	Camera	
12:35 PM	Removed 3k assistant/semi-jailbreak		
12:36 PM	Rebooted	rebooted	
12:37 PM	safari	safari	
12:38 PM	Searched “dog”		
12:38 PM	Downloaded dog image		
12:40 PM	Gallery: went to dog image & set as home screen background		
12:43 PM	Launched “lunch” nearby google map app		
12:43 PM	Turned on Locations		
12:54 PM	Googled “pets burlington vt”	Safari	

Time	User Action	Machine Action	Comments
1:00 PM	Clicked on pets craigslist for Burlington		
1:01 PM	Searched “lab”	Craigslist/safari	
1:03 PM	Went into camera app and edited photo of dog		
1:27 PM	Went to app store	App store	
1:28 PM	Went to maps	Maps	
1:29 PM	Typed “directions to lake george ny”	Maps	Chose “lake George escapes campground”
	Typed “north creek, ny” in “TO” slot	Maps	
1:34 PM	Chose “North Pole” for directions	Map	
2/16/16			
1:41 PM	Went to app store App Store & installed two dots	Installed two dots	Put in password for Apple ID & chose “require after 15 minutes”
I:44 PM	Opened two Dots- Click “OK” for notifications	Opened app	Generated data via the app (played game)
1:46 PM	Opened App Store & Downloaded “Burlington Free Press”	Downloaded app	Clicked “Allow” on App store allowing access to location & “save password for Free items- clicked yes
1:48 PM	Opened Burlington Free Press App	Burlington Free Press App	Clicked “Allow” on knowing location & notifications
1:49 PM	Browsed App & opened articles	Burlington Free Press App	
1:53 PM	Opened App Store & downloaded “triangle Dash!”	Downloaded & opened app	Generated data on the app
1:59 PM	Deleted Triangle dash!	Deleted App	
2:06 PM	Added Reminder in Reminders app		“Remove Garbage”
2:07 PM	Opened wallet App		Allowed Location Use
2:07 PM	Opened & Added Note		The Note was just “suggested text” being clicked on & added a drawn photo & camera picture
2:33 PM	Went to Apple.com	Safari	Clicked “learned more” on iPhone 6s & watched video
2/17/16			
1:18 PM	Sent email to joseph.cozzi@mymailchamplain.edu	Mail	
1:22 PM	Took photo of water bottle	Camera	
1:22 PM	Deleted photo of water bottle	photos	
1:23 PM	Received email from joseph.cozzi@mymail.champlain.com	mail	

Time	User Action	Machine Action	Comments
1:24 PM	Opened email from joseph.cozzi@mymail.champlain.edu	mail	
1:29 PM	Took photo of phone box	Camera	
1:30 PM	Sent photo to joseph.cozzi@mymail.champlain.edu	Mail	
1:30 PM	Deleted photo of phone box	photos	
1:33 PM	Set alarm for 1:30pm	Clock	
1:33 PM	Disabled the alarm	Clock	
1:34 PM	Added Rome, Italy to world clock	clock	
1:34 PM	Ran stopwatch	Clock	Ran for 2:59.82. (2 minutes, 59 seconds, 82 milliseconds. Hit “lap” button at 2:29.99. hit reset button after
1:39 PM	Wrote a note	notes	Note says “Hello World”
1:41 PM	Used calculator, entered 123 +456	calculator	Cleared calc after
1:41 PM	Opened app store and went to “top charts, then “free”	App store	
1:43 PM	Installed “solitaire”	App store	
1:44 PM	Opened solitaire	Opened app	
1:44 PM	Played a game of solitaire to generate app data		
1:47 PM	Restarted game and played a second one		
1:54 PM	Won game		
1:54 PM	Closed popup ad.		
1:55 PM	Closed app		
1:55 PM	Deleted “solitaire” app		
1:56 PM	Opened settings	Settings	
1:56 PM	In settings, went to sounds	Settings	
1:59 PM	Changed ringtone from “opening” to “by the seaside”	settings	
2:00 PM	Changed ringtone back to opening	settings	
2:03 PM	Opened stocks app	Stocks	
2:04 PM	Closed stocks app	Stocks	
2:05 PM	Opened reminders app	reminders	
2:06 PM	Added reminder	reminders	Reminder says “take morning shower”
2:10 PM	Opened weather app	weather	

Time	User Action	Machine Action	Comments
2:10 PM	Added Sacramento California to weather list	weather	

References

Klosowski, Thorin. "How to Jailbreak Your IPhone." *Lifehacker*. N.p., 14 Oct. 2015. Web. 23 Jan. 2016. <http://lifehacker.com/5771943/how-to-jailbreak-your-iphone-the-always-up-to-date-guide-ios-61>.

"TaiG Beta - Jailbreak IOS 9.2.1." *TaiG9 Beta*. N.p., 20 Jan. 2016. Web. 28 Jan. 2016. <http://taig9.com/beta/>.