Mobile Digital Forensic Tool using Santoku Linux

Pedro J. Acevedo Rivera
Master in Computer Science
Advisor: Jeffrey Duffany, Ph.D.
Electrical and Computer Engineering & Computer Science Department
Polytechnic University of Puerto Rico

Abstract — *Santoku 0.5 is a free and Open Source Linux Distribution dedicated to mobile forensics, mobile security, and mobile malware analysis. Santoku provides tools for; analyzing and acquiring data including free versions of commercial tools, examining mobile malware including emulators and disassembly tools, and performing assessments of mobile apps with scripts designed to detect common issues in mobile applications, all packaged in an easy to download and use Open Source platform. Additionally, it provides some tools use for analyzing network traffic and images.*

***Key Terms** - Digital Evidence, Mobile Forensics, Mobile Forensic Investigation, Mobile Security.*

INTRODUCTION

Today's world of digital forensics is changing significantly as mobile platforms, and cloud services solutions create a major shift for organization owned and maintained server and desktop computing devices to Bring Your Own Device (BYOD) and Infrastructure and Platform as a Service (IaaS/PaaS) instances.

As the pattern for computing platforms change, digital forensic investigators are expected to adapt their work to continue providing forensic investigative services despite the platform changes. Due to this change in computing platforms, a community shift towards increased knowledge and skill in the area of cloud and mobile forensics is needed and expected by this researcher.

Despite the change in industry norms, many commercial tools for mobile forensics continue to be cost prohibitive to smaller commercial and government forensic shops (Cellebrite UFED Pro CLX starts at $15,999). This has caused many forensic examiners and procurement specialists to look to Open Source tools to fill the gap created by tight acquisition budgets.

Santoku is a free and Open Source Linux distribution developed as a fork of the OWASP (Open Web Application Security Project) MobiSec Ubuntu distribution. With Santoku, a mobile forensic examiner or beginner mobile forensic examiner can download, install, and begin utilizing a free mobile forensic toolset. Santoku can be installed as a virtual machine or as an OS (operating system) since is based on Ubuntu platform. The installment is in a matter of minutes. Additionally, Santoku contains tools that allow the examiner to perform acquisitions and investigations on iOS and Android Mobile OS as well as Samsung devices. Some of the functions a mobile forensic investigator/examiner can accomplish with Santoku include:

1. Extraction of data – this includes messages, phone calls, and Short Messaging Service (SMS). The text messages can be seen in CSV (Commas Separated Value) format from Android devices using AFLogical OSE (Open Source Edition).
2. Create logical iOS device backups with Libimobiledevice
3. Analyze iOS backups using iPBA2 (iPhone Backup Analyzer 2)
4. Flash firmware onto Samsung Galaxy S devices using Heimdall, qallowing a forensic examiner to install custom recovery partitions on Samsung devices.

BACKGROUND

The purpose of choosing this topic as my project is because of my interest in digital forensic.

Digital forensic has been one of my first classes that I took when coursing through my masters. In this class, we had the chance to know more about encryption, hashes, and tools to help us understand the various fields and knowledge a forensic examiner should have. During class the students, had the chance to explore various tool for forensic analysis but only on emails, picture images, and computer devices.

As technology keeps advancing, a professional in the field of computer and technology needs to be on top of their career. This means that they need to keep investigating, reading, and experiment with the plethora of documentation and tools that are out there. The downfall of a forensic examiner is the cost and variation of the tools that are in the market.

Whenever a forensic examiner needs specific tools and machines to do their job, they encounter a problem in terms of limitations. There is not one single tool or machine that cannot do a throughout examination of a device. This is also applicable in mobile forensic. In today's world, there are a ton of different of mobile devices that are storing personal information.

Mobile Forensic is a hard ramification because of all the different phone models and the limitation to do a logical extraction of the data. There is no one single tool that can help an examiner to make a data extraction on different mobile phones model. One of the biggest issue is that if you find a good tool that can help do a Mobile Forensic examination it will cost a good amount of money. Given this some Digital Forensic professional in my workplace recommended me to use Santoku.

Santoku is an open source tool that can help do a logical extraction and examination of android devices and iOS devices. In the forensic classes that I have taken, non-have mentioned an open source tool. I took on the challenge to explore this tool and witness its capability and limitations.

PROBLEM

As mentioned before, Mobile Forensic is a ramification of Digital Forensic with its own problem and limitations. There are limited resources when teaching mobile forensic to student. It is good to have the theory about how to do mobile forensic and what to look for, but it is necessary to have a hands-on experience on what to look and how to obtain it. The use of an open source can help student start on their own extraction of mobile data from their personal phones.

Following is the description of the software used to do a logical extraction of data from a mobile device. This kind of extraction is not brute force, so a pin or password is known to unlock the phone and specific settings must be applied to the phones.

The requirement of linux language is required to troubleshoot in the OS and explore the different folder where the data is residing. This will also understand the student that the use of linux must be known in the world of computer science.

SANTOKU OS

The Santoku Linux provides tools to equip forensic examiners in the following three examination areas:

1. Mobile Forensics
2. Mobile Malware
3. Mobile Security

The Community Edition of Santoku [1] Linux runs natively in the lightweight Lubuntu 14.04 Linux distro and can be run as a Virtual Machine in any hypervisor software, although VMware Player and VirtualBox [2] are mentioned in product documentation with VirtualBox recommended by Santoku's developers. Additionally, Santoku Linux 0.5 is a 64-bit OS that will only work with 64-bit hardware and software and it is available through SourceForge as a 2.5 GB .iso with the following MD5 and SHA1 hashes:

- MD5– c2dcab27e6444730acc9bc351f34e543
- SHA1– 4d39adc01c443ac24a53a33f0ac077980d77c1fe

Alternatively, you can download Lubuntu and update it with new Santoku packages from the Santoku package repository using the following build script provided by the developers:

- "Santoku-05 build.sh"

For the purposes of this project, I chose to download and install the software from the full .iso file as a VM in VirtualBox.

Installing Santoku

The latest download .iso file for Santoku 0.5 can be found on SourceForge at:

https://sourceforge.net/projects/santoku/files/latest/download

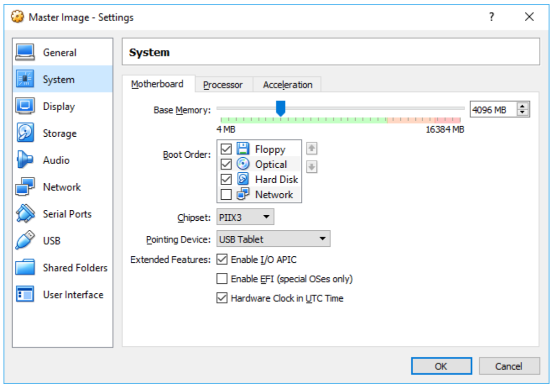
The download takes approximately 20 minutes to complete and was done on a Windows 10, 64-bit. The VM was provisioned using Workstation's standard Ubuntu 64-bit template, with 60GB of thin provisioned storage, 4 GB of RAM (Random Access Memory), 4 processors, using Bridged Networking, adding the usb ports that the mobile devices are going to be connected, and accepting all other defaults.

To install Santoku, on the splash screen tab down or press the arrow key down to the "install – start the installer directly" selection and press <Enter>. Follow the menu prompts and accepted defaults for all other selections and settings choosing not to encrypt the home folder or any other aspect of the drive. The install takes approximately thirty minutes and requires a restart. After the initial install restart, the system prompts the user to install 14.04 updates, this takes about thirty minutes to install and also prompts a restart to complete the install. The user is also prompted to upgrade the OS to 16.04 "Xenial Xerus" an upgrade that is necessary to utilize certain Android emulators developed for 16.04 (AVD (Android Virtual Device Manager) and Android device Emulator). For the purposes of this project, we need to upgraded to 16.04.

The About Santoku page on the website purports that Santoku is a "bootable Linux environment designed to make life easier" and providing the following tools.

- Pre-installed platform SDKs, drivers, and utilities.
- Pre-configured GUI frameworks, such as PyGTK to support GUI tools.
- GUI tools for easy deployment and control of mobile apps.
- Auto Detection and setup of new connected mobile devices.

While in many ways the statement is accurate, in other ways it is very misleading. As I went through the process of my research, Santoku can be somewhat temperamental and it requires a fair time of care. Two helpful hints for experienced users or Linux "newbies" like myself is to configure your VM with generous resources in terms of disk space, memory, and processors.


Master Image - Settings

System

Base Memory: 4096 MB (4 MB – 16384 MB)

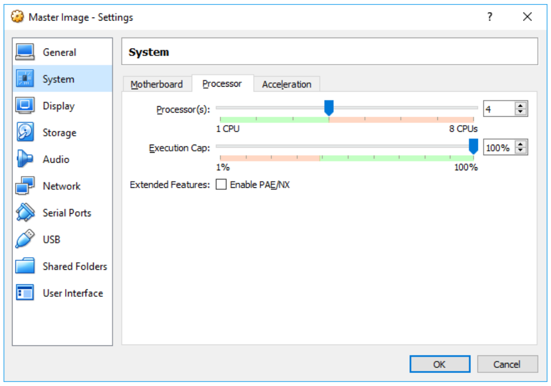
Boot Order:
☑ Floppy
☑ Optical
☑ Hard Disk
☐ Network

Chipset: PIIX3

Pointing Device: USB Tablet

Extended Features:
☑ Enable I/O APIC
☐ Enable EFI (special OSes only)
☑ Hardware Clock in UTC Time

Figure 1
VM Hardware Settings


Master Image - Settings

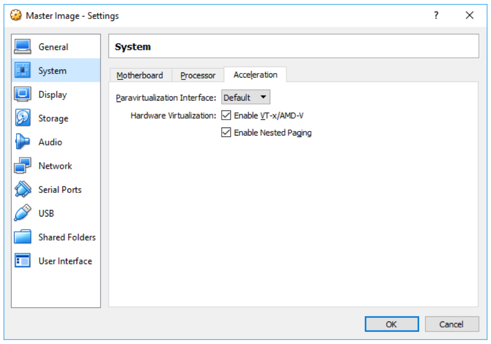
System

Processor(s): 4 (1 CPU – 8 CPUs)

Execution Cap: 100% (1% – 100%)

Extended Features: ☐ Enable PAE/NX

Figure 2
VM Hardware Settings



Paravirtualization Interface: Default

Hardware Virtualization: ☑ Enable VT-x/AMD-V

☑ Enable Nested Paging

Figure 3
Intel-VT/AMD-V Settings

Additionally, some tools and emulators proved unable to run without the appropriate settings for Intel Virtualization Technology (VT) or AMD Virtualization (AMD-V) settings applied to the VM as shown in **Error! Reference source not found.**.

Once the appropriate settings are configured in the VM, performance and usability improved significantly and most of the labs and HOWTOS were able to be completed as depicted in the Santoku web documentation.

Desktop and GUI

The Santoku Linux desktop has a mini-blade at the lower left-hand side of the screen that acts as the menu:

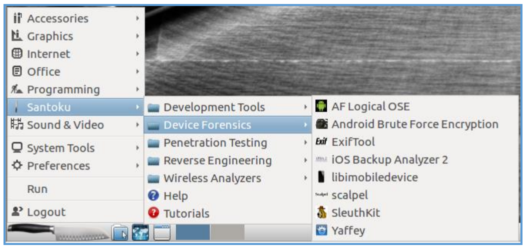


Figure 4
Santoku Menu

The core device forensic tools are displayed under: **Santoku** > **Device Forensics** > as in Figure 4

Santoku Menu.

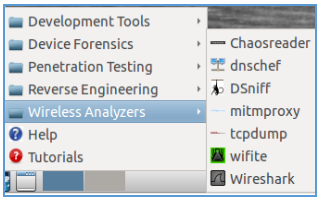


Figure 5
Wireless Analyzers

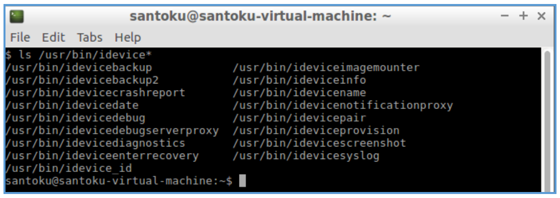
In addition to the "Device Forensics" and "Development Tools" menu headings, Santoku is also equipped with "Penetration Testing" and "Reverse Engineering" tools as well as useful "Wireless Analyzers" like Wireshark, tcpdump, DSniff, and others as shown in Figure 5: Wireless Analyzers.

iPhone Mobile Device Forensics

To aid us in our discovery process, the Santoku site and the distro itself provide a number of useful "HOWTO's". We will begin with the "*create a logical iOS device backup using libimobiledevice on Santoku Linux*" HOWTO [3].

The first step is to create a "logical" backup using libimobiledevice. In order to perform this backup, you physically connect your iPad or iPhone to the computer where you are running the Santoku Linux VM Workstation and then launched *libimobiledevice* from the menu using the GUI. This opened a terminal window with an $ **ls /usr/bin/idevice*** command already run as displayed in Figure 6

Libimobiledevice Commands listing all idevice commands.



```
$ ls /usr/bin/idevice*
/usr/bin/idevicebackup            /usr/bin/ideviceimagemounter
/usr/bin/idevicebackup2           /usr/bin/ideviceinfo
/usr/bin/idevicecrashreport       /usr/bin/idevicename
/usr/bin/idevicedate              /usr/bin/idevicenotificationproxy
/usr/bin/idevicedebug             /usr/bin/idevicepair
/usr/bin/idevicedebugserverproxy  /usr/bin/ideviceprovision
/usr/bin/idevicediagnostics       /usr/bin/idevicescreenshot
/usr/bin/ideviceenterrecovery     /usr/bin/idevicesyslog
/usr/bin/idevice_id
santoku@santoku-virtual-machine:~$
```

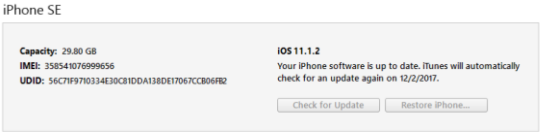
Figure 6
Libimobiledevice Commands

The next step is to see if we can determine the UDID (Unique Device Identifier). This UDID is a sequence of 40 letters and numbers specific to an iOS device by issuing the following command:

santoku@santoku-virtual-machine:~$ **idevice_id -l**

56C71F9710334E30C81DDA138DE17067CCB06FB2

This can be verified by reconnecting the device to the laptop (shutting down the VM) and displaying the UDID in iTunes as shown in Figure 7: UDID displayed in iTunes (you will need to click where it says "Serial Number").



iPhone SE

Capacity: 29.80 GB
IMEI: 358541076999656
UDID: 56C71F9710334E30C81DDA138DE17067CCB06FB2

iOS 11.1.2
Your iPhone software is up to date. iTunes will automatically check for an update again on 12/2/2017.

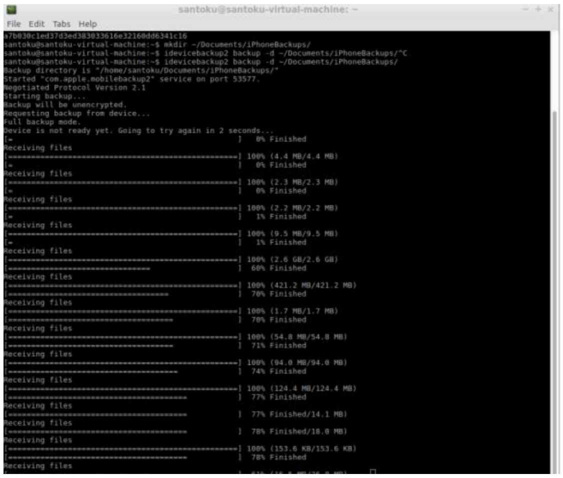
Figure 7
UDID Displayed in iTunes

Once we have findout the UDID we proceed with doing a backup. The HOWTO instructs us to ensure the iOS device is unlocked (libimobiledevice cannot perform a backup if the device is pin locked or password lock) and also instructs us to change our auto-lock settings to "Never" to ensure the iOS device does not lock us out for inactivity while performing a backup.

We then reconnect the device to the VM (or start the VM) and if prompted, press "trust" (this will appear in the iPhone or iPad) the computer we are connecting to and are now ready to back up our device. You will need to open libimobiledevice again or a terminal window and type the following:

santoku@santoku-virtual-machine:~$ **mkdir ~/Documents/iPhoneBackups**

santoku@santoku-virtual-machine:~$ **idevicebackup2 backup ~/Documents/iPhoneBackups/**

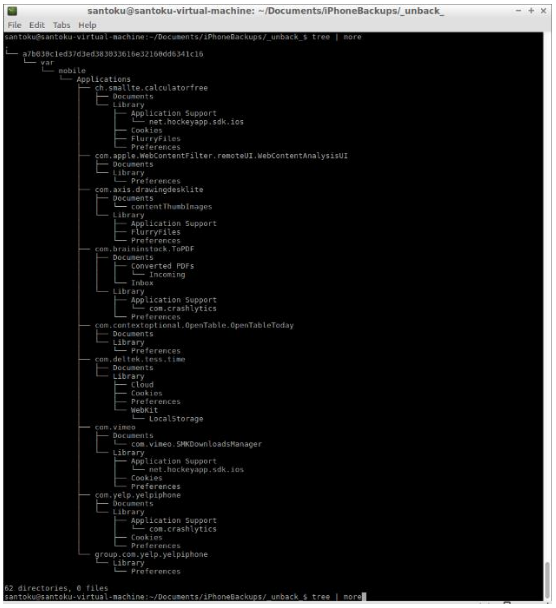


```
a7b030c1ed37d3ed383033616e32160dd6341c16
santoku@santoku-virtual-machine:~$ mkdir ~/Documents/iPhoneBackups/
santoku@santoku-virtual-machine:~$ idevicebackup2 backup -d ~/Documents/iPhoneBackups/^C
santoku@santoku-virtual-machine:~$ idevicebackup2 backup -d ~/Documents/iPhoneBackups/
Backup directory is "/home/santoku/Documents/iPhoneBackups/"
Started "com.apple.mobilebackup2" service on port 53577.
Negotiated Protocol Version 2.1
Starting backup...
Backup will be unencrypted.
Requesting backup from device...
Full backup mode.
Device is not ready yet. Going to try again in 2 seconds...
[=                                                 ]   0% Finished
Receiving files
[==================================================] 100% (4.4 MB/4.4 MB)
[=                                                 ]   0% Finished
Receiving files
[==================================================] 100% (2.3 MB/2.3 MB)
[=                                                 ]   0% Finished
Receiving files
[==================================================] 100% (2.2 MB/2.2 MB)
[=                                                 ]   1% Finished
Receiving files
[==================================================] 100% (9.5 MB/9.5 MB)
[=                                                 ]   1% Finished
Receiving files
[==================================================] 100% (2.6 GB/2.6 GB)
[==============================                    ]  60% Finished
Receiving files
[==================================================] 100% (421.2 MB/421.2 MB)
[==================================                ]  70% Finished
Receiving files
[==================================================] 100% (1.7 MB/1.7 MB)
[===================================               ]  70% Finished
Receiving files
[==================================================] 100% (54.8 MB/54.8 MB)
[===================================               ]  71% Finished
Receiving files
[==================================================] 100% (94.0 MB/94.0 MB)
[====================================              ]  74% Finished
Receiving files
[==================================================] 100% (124.4 MB/124.4 MB)
[=======================================           ]  77% Finished
Receiving files
[=======================================           ]  77% Finished/14.1 MB)
Receiving files
[=======================================           ]  78% Finished/18.0 MB)
Receiving files
[==================================================] 100% (153.6 KB/153.6 KB)
[=======================================           ]  78% Finished
Receiving files
```

Figure 8
Idevicebackup2

The backup is stored in a folder named after the UDID of your device in **~/Documents/iPhoneBackups/**.

Once we have done the backup of the device we will need to unback it. When attempting to perform the next step in the HOWTO to "unback" or extract the file and make it easily browsable, it is discovered that the unback functionality in libimobiledevice was partially broken for relatively newer iOS devices (apparently iOS 9+) and retrieved only partial results. Since my iPhone is running iOS 11.1.2, I could only unback half of the files and it was stuck on "Device seems to be busy" (this comes back to apple being secure in their phones) and exited the command. The only way that we can see what was unbacked was to install the tree view and go to the unback folder and see it. Once you do this it will look something like Figure 9: Partial unback functionality.



santoku@santoku-virtual-machine: ~/Documents/iPhoneBackups/_unback_
File Edit Tabs Help

```
santoku@santoku-virtual-machine:~/Documents/iPhoneBackups/_unback_$ tree | more
.
└── a7b030c1ed37d3ed383033616e32160dd6341c16
    └── var
        └── mobile
            └── Applications
                ├── ch.smallte.calculatorfree
                │   ├── Documents
                │   └── Library
                │       ├── Application Support
                │       │   └── net.hockeyapp.sdk.ios
                │       ├── Cookies
                │       ├── FlurryFiles
                │       └── Preferences
                ├── com.apple.WebContentFilter.remoteUI.WebContentAnalysisUI
                │   ├── Documents
                │   └── Library
                │       └── Preferences
                ├── com.axis.drawingdesklite
                │   ├── Documents
                │   │   └── contentThumbImages
                │   └── Library
                │       ├── Application Support
                │       ├── FlurryFiles
                │       └── Preferences
                ├── com.braininstock.ToPDF
                │   ├── Documents
                │   │   ├── Converted PDFs
                │   │   │   └── Incoming
                │   │   └── Inbox
                │   └── Library
                │       ├── Application Support
                │       │   └── com.crashlytics
                │       └── Preferences
                ├── com.contextoptional.OpenTable.OpenTableToday
                │   ├── Documents
                │   └── Library
                │       └── Preferences
                ├── com.deltek.tess.time
                │   ├── Documents
                │   └── Library
                │       ├── Cloud
                │       ├── Cookies
                │       ├── Preferences
                │       └── WebKit
                │           └── LocalStorage
                ├── com.vimeo
                │   ├── Documents
                │   │   └── com.vimeo.SMKDownloadsManager
                │   └── Library
                │       ├── Application Support
                │       │   └── net.hockeyapp.sdk.ios
                │       ├── Cookies
                │       └── Preferences
                ├── com.yelp.yelpiphone
                │   ├── Documents
                │   └── Library
                │       ├── Application Support
                │       │   └── com.crashlytics
                │       ├── Cookies
                │       └── Preferences
                └── group.com.yelp.yelpiphone
                    └── Library
                        └── Preferences

62 directories, 0 files
santoku@santoku-virtual-machine:~/Documents/iPhoneBackups/_unback_$ tree | more
```

Figure 9
Partial Unback Functionality

Now that we've created our backup with libimobiledevice, we'd like to analyze that backup using the iPhone Backup Analyzer 2 (iPBA2). Unfortunately, the link to the HOWTO for using iPBA2 advertised on the Santoku Linux webpage points to a different HOWTO. Additionally when trying to open the backup file using the tool it kept foreclosing. I couldn't find a way to prevent this and could not find the right instruction to reinstall the tool. Once the developers fix this, the tool GUI will be intuitive enough to facilitate selfexploration devoid of specific tutorials. Thankfully an ex-coworker add gone through the tool and provided me with some information and screenshots of how the tool works, Figure 10: iPBA2 shows this. Steps taken to use the tool will be, from the GUI, we selected the following to open iPBA2: **Santoku > Device Forensics > iOS Backup Analyzer.** Once the application opened, we selected: **File > Open Archive** then navigated to the iPhone backup in the **Open Directory** window and selected **Open** to open the backup for viewing.

In the **main.py** window launches within the application displaying a status bar and ultimately opens the backup image for browsing. Two windows on the left hand-side of the iPBA2 application window *Backup Info* and *Backup Filesystem* provide at-a-glance information with the Backup Info window providing general information about the iPad including Serial Number, UDID, Device Name, Product Version and Product Type. The Backup filesystem window provides a logical view of the filesystem in directory tree fashion for browsing, while in the main menu, the **Plugins** dropdown can be selected to run one of several useful plugins. We selected the Messages and Note Browsers, and the Call History plugin as well with useful results.

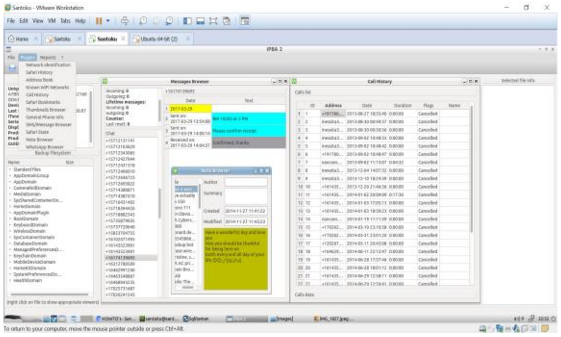


Figure 10
iPBA2

The examiner will be able to browse through the Backup filesystem window results and use built-in iPBA2 functionality to open several different types of file appropriate viewers to view stored data as well as in some cases being able to export attachments, images, other media, ".plist" files and database files to a folder for viewing with other Santoku tools or for transferring to another device for additional analysis if necessary. In the right most pane in the application window "Selected file info" provides a meaningful and easy to read metadata for the currently viewed file, as shown in Figure 11: plist files viewed in plist viewer, Hex, and ASCII viewers.

iPBA2 natively offers users the ability to view key system files in plist viewer, and ASCII or Hex viewer. One such file is the **lsdidentifiers.plist** file stored at: **/private/var/db/lsd/com.apple.lsdidentifiers.plist** and managed by the Launch Services Daemon (lsd). This file is important to mobile forensics in that it will allow a forensic investigator to provide non-repudiation that a service or App was launched on a mobile device even if the App and corresponding service were subsequently removed from the device.

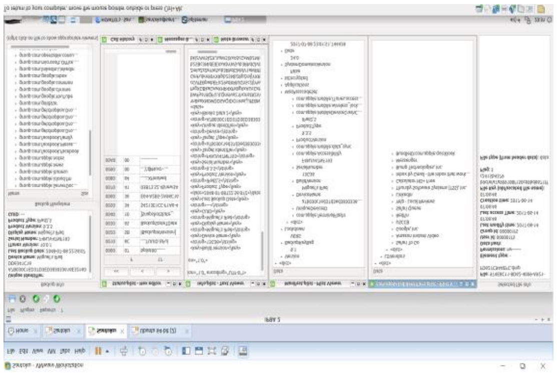


Figure 11
.plist Files Viewed in Plist Viewer, Hex, and ASCII Viewers

Another functionality that the tool could demonstrate was the ability to view browsing history stored in:

App Domain > com.apple.mobilesafari > Library/Safari > SearchDescriptions.plist | History.db | History.plist | SuspendedState.plist

With this information, an examiner will be able to provide non-repudiation report for sites a user has surfed to. An example is shown in Figure 12: Web History.

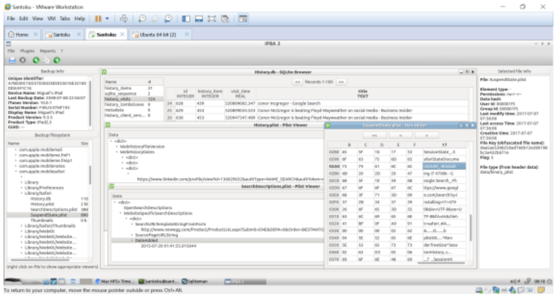

Santoku - VMware Workstation
File Edit View VM Tabs Help
Home
Santoku
Ubuntu 64-bit (2)
iPBA 2
File Plugins Reports ?
Backup Info
Backup Filesystem
History.db - SQLite Browser
History.plist - Plist Viewer
SearchDescriptions.plist - Plist Viewer
Selected File Info
conor mcgregor - Google Search
Conor McGregor is beating Floyd Mayweather on social media - Business Insider
(right click on file to show appropriate viewers)
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Figure 12
Web History

Another guidance besides the HOWTO presented at beginning was HackMag [4], which contains almost the same HOWTO but with more knowledgeable information on each of the steps.

Android Mobile Device Forensic

Santoku Linux also provides tools for extracting data from Android devices, as well as emulating an android device. The first tool we'll use to perform the extraction of data from Android devices is Android Forensics Logical Open Source Edition or AFLogical OSE. AFLogical OSE pulls MMS, SMS, Contacts and Call Logs from Android devices.

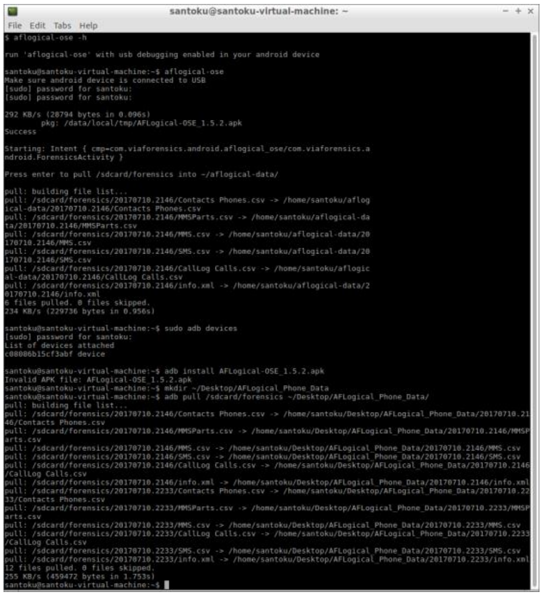
Using a Samsung Galaxy 6 and following the directions provided in the HOWTO of AFLogical OSE [5] posted on the Santoku site:

1. Ensure that the Android device is connected to the Santoku VM.
2. Enable USB debugging on your device.
   a. **Apps > Settings > About phone or tablet** then tap the build number seven times (until a message appears)
   b. Go to **Apps > Settings > Developer Options** and Ensure USB Debugging is selected
   c. Trust the Santoku VM on the tablet or phone and select always trust this device
3. In the VM, select: **Santoku > Device Forensics > AF Logical OSE**
4. In the Terminal window that opens, enter as follows: $ **aflogical-ose**

   (Note: this pushes the AFLogical-OSE_1.5.2.apk to your device)
5. Press enter in the terminal window to pull /sdcard/forensics into ~/aflogical-data/
6. $ **sudo adb devices**

   (Note: this will display an ID for the attached device)
7. On your Android device, open the AFLogical OSE application, select all checkmarks and extract the data. (Note: a progress bar and a "Data extraction completed." message will appear when done.
8. Pull the data from your card or phone to the Santoku VM:
   a. $**mkdir ~/Desktop/AFLogical_Phone_Data**
   b. $ **adb pull /sdcard/forensics/ ~/Desktop/AFLogical_Phone_Data**

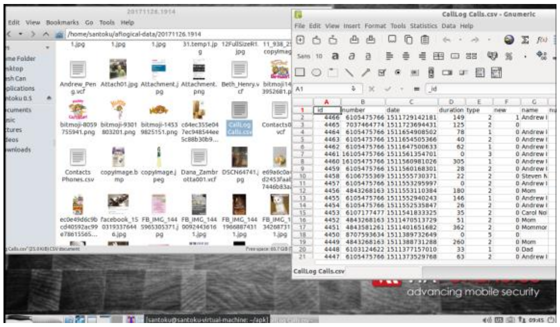
Your extracted data is now stored in the following directory: **~/Desktop/AFLogical_Phone_Data.**



```
$ aflogical-ose -h

run 'aflogical-ose' with usb debugging enabled in your android device

santoku@santoku-virtual-machine:~$ aflogical-ose
Make sure android device is connected to USB
[sudo] password for santoku:
[sudo] password for santoku:

292 KB/s (28794 bytes in 0.096s)
        pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success

Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaforensics.a
ndroid.ForensicsActivity }

Press enter to pull /sdcard/forensics into ~/aflogical-data/

pull: building file list...
pull: /sdcard/forensics/20170710.2146/Contacts Phones.csv -> /home/santoku/aflog
ical-data/20170710.2146/Contacts Phones.csv
pull: /sdcard/forensics/20170710.2146/MMSParts.csv -> /home/santoku/aflogical-da
ta/20170710.2146/MMSParts.csv
pull: /sdcard/forensics/20170710.2146/MMS.csv -> /home/santoku/aflogical-data/20
170710.2146/MMS.csv
pull: /sdcard/forensics/20170710.2146/SMS.csv -> /home/santoku/aflogical-data/20
170710.2146/SMS.csv
pull: /sdcard/forensics/20170710.2146/CallLog Calls.csv -> /home/santoku/aflogic
al-data/20170710.2146/CallLog Calls.csv
pull: /sdcard/forensics/20170710.2146/info.xml -> /home/santoku/aflogical-data/2
0170710.2146/info.xml
6 files pulled. 0 files skipped.
234 KB/s (229736 bytes in 0.956s)

santoku@santoku-virtual-machine:~$ sudo adb devices
[sudo] password for santoku:
List of devices attached
c08086b15cf3abf device

santoku@santoku-virtual-machine:~$ adb install AFLogical-OSE_1.5.2.apk
Invalid APK file: AFLogical-OSE_1.5.2.apk
santoku@santoku-virtual-machine:~$ mkdir ~/Desktop/AFLogical_Phone_Data
santoku@santoku-virtual-machine:~$ adb pull /sdcard/forensics ~/Desktop/AFLogical_Phone_Data/
pull: building file list...
pull: /sdcard/forensics/20170710.2146/Contacts Phones.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.21
46/Contacts Phones.csv
pull: /sdcard/forensics/20170710.2146/MMSParts.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2146/MMSP
arts.csv
pull: /sdcard/forensics/20170710.2146/MMS.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2146/MMS.csv
pull: /sdcard/forensics/20170710.2146/SMS.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2146/SMS.csv
pull: /sdcard/forensics/20170710.2146/CallLog Calls.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2146
/CallLog Calls.csv
pull: /sdcard/forensics/20170710.2146/info.xml -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2146/info.xml
pull: /sdcard/forensics/20170710.2233/Contacts Phones.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.22
33/Contacts Phones.csv
pull: /sdcard/forensics/20170710.2233/MMSParts.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2233/MMSP
arts.csv
pull: /sdcard/forensics/20170710.2233/MMS.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2233/MMS.csv
pull: /sdcard/forensics/20170710.2233/CallLog Calls.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2233
/CallLog Calls.csv
pull: /sdcard/forensics/20170710.2233/SMS.csv -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2233/SMS.csv
pull: /sdcard/forensics/20170710.2233/info.xml -> /home/santoku/Desktop/AFLogical_Phone_Data/20170710.2233/info.xml
12 files pulled. 0 files skipped.
255 KB/s (459472 bytes in 1.753s)
santoku@santoku-virtual-machine:~$
```

Figure 13
AFLogical OSE Data Extraction

In this case, although it appears that the AFLogical OSE data pull was successful as shown in Figure 12: AFLogical OSE Data Extraction, the created files yielded relatively little information in comparison to the results using libimobiledevice and iPBA2 when examining the iOS device. The files retrieved from our Android device were retrieved in duplicate and stored in two separate timestamped folders 20170710.2146 and 20170710.2333 respectively. CallLogCalls.csv, MMS.csv, MMSParts.csv, SMS.csv, and info.xml yielded no useable data (only field headers) and ContactsPhones.csv was able to pull phone contacts that were linked from the owner's phone to his tablet device.


/home/santoku/aflogical-data/20171126.1914
CallLog Calls.csv - Gnumeric
File Edit View Insert Format Tools Statistics Data Help
A1
_id
CallLog Calls.csv
advancing mobile security

	A	B	C	D	E	F	G
1	_id	number	date	duration	type	new	name
2	4466	6105475766	1511729142181	149	2	1	Andrew I
3	4465	7037464774	1511723694431	125	2	0	
4	4464	6105475766	1511654908502	78	1	0	Andrew I
5	4463	6105475766	1511654505366	40	2	0	Andrew I
6	4462	6105475766	1511647500633	62	1	0	Andrew I
7	4461	16105475766	1511561354701	0	3	0	Andrew I
8	4460	16105475766	1511560981026	305	1	0	Andrew I
9	4459	6105475766	1511560168301	28	2	0	Andrew I
10	4458	6106755369	1511555730371	22	2	0	Steven N
11	4457	6105475766	1511553295997	0	2	0	Andrew I
12	4456	4843268163	1511553110384	180	2	0	Mom
13	4455	6105475766	1511552940243	146	1	0	Andrew I
14	4454	6105475766	1511552535847	26	2	0	Andrew I
15	4453	6107177477	1511541833325	35	2	0	Carol No
16	4452	4843268163	1511470513729	51	2	0	Mom
17	4451	4843581261	1511401651682	362	2	0	Mommor
18	4450	8707593634	1511389732649	0	5	0	
19	4449	4843268163	1511388731288	260	2	0	Mom
20	4448	6103124622	1511377157010	33	1	0	Dad
21	4447	6105475766	1511373529768	63	2	0	Andrew I

Figure 14
AFLogical OSE Files Pulled

For a more robust guidance on what tools can be used in Santoku with Android devices, I would recommend "INFOSEC INSTITUE" about "Android Forensics Labs" [6]. Another way that students can benefit quite a lot from the Santoku tool will be with rooting the device [7] and give you more administrative privilege to the Android smartphone. Every student or person that tries to root their device will need to do it at their own risk, because it will void the warranty on the phone and maybe it's usefulness if done incorrectly.

PROS AND CONS

To sum up this whole project we need to highlight the pros and cons. Let's start with the pros. The best pro that this tool has is that is an open source tool and it doesn't cost any money. Another good pro is that it can be used for persons who has a strong Linux background or persons with the desire to develop a strong Linux background, Santoku is a great out of the box starter project with enough remaining development work.

The cons of Santoku in some cases are clumsy and its tools are not *immediately* available to be used by all users. Sometimes when launching the tools from the GUI menu without "sudo'ing" to root privileges from the command line will provide users with limited permissions to the launched tools and potentially frustrating time-consuming results.

Additionally, the Santoku website has not been maintained over time so some of the tutorials are either missing (e.g. the HOWTO for iPhone Backup Analyzer 2 directs users to a HOWTO for compiling AFLogical OSE on Santoku Linux) and some haven't been updated in several years. All of this leads to an unpleasant but challenging user experience for Linux newbies and intermediate users that is slightly less than ideal.

CONCLUSION

In conclusion, Santoku Linux contains several outstanding free tools, collected in a single purpose-built OS that make it a meaningful addition to a digital or mobile forensics team. Santoku proves to be frustrating to maintain and extract useful results with for examiners with less than intermediate to advanced Linux skills. It is highly encouraged for anyone who wants to use this tool regularly to communicate directly with the owners of the tool, which is at NowSecure and volunteer to help them improve the tool.

All that being stated, I firmly believe that the Cons of Santoku far outweigh its Pros in regards to mobile forensic. In terms of a forensic and penetration testing tool the Pro's outweighs the cons because of all the tools it has and their capabilities. This project was an interesting and educational experience to examine the tool, its applications and some of its functionality. Additionally going to the forums [8] help quite a lot when dealing with the iOS logical extraction.

Hopefully the owners of Santoku see some benefit to this tool and update its applications and OS in conjunction with the HOWTOS for better experience and recognition. One last thing that helped me in understanding a little more about mobile Forensics was reading a book that was lent to me [9]. This also helped me to have an idea of what it was needed to be done and how to extract some of the data that resides on the phones.

References

[1] Oracle. (2017, 1 September). VirtualBox Download VirtualBox [Online]. Available: https://www.virtualbox.org/wiki/Downloads.

[2] NowSecure. (2017, 1 September). Santoku 0.5 – Packaged and Delivered [Online]. Available: https://santoku-linux.com

[3] NowSecure. (2017, 1 September). HOWTO create a logical iOS device backup using libimobiledevice on Santoku Linux [Online]. Available: https://santoku-linux.com/howto/mobile-forensics/howto-create-a-logical-backup-of-an-ios-device-using-libimobiledevice-on-santoku-linux/.

[4] Hackmag. (2017, 23 September). Apple Forensic: Advanced Look onto Apple Security [Online]. Available: https://hackmag.com/security/apple-forensic/.

[5] NowSecure (2017, 1 September), HOWTO: Use AFLogical OSE for Logical Forensics of an Android Device [Online] Available: https://santoku-linux.com/howto/howto-use-aflogical-ose-logical-forensics-android/.

[6] Sinrivas. (2017, 26 September). Android Forensics Labs [Online]. Available: http://resources.infosecinstitute.com/android-forensics-labs/#gref.

[7] J. I. James. (2017, 25 November). [Linux] Android Acquisition using ADB, root, netcat and DD [Online] Available: https://www.youtube.com/watch?v=UQYuaOC5v0I.

[8] S. Gleske (2017, 9 September). libimobiledevice_ifuse_Ubuntu.md – iOS 10+ to work with Lubuntu 16.04 [Online] Available: https://gist.github.com/samrocketman/70dff6ebb18004fc37dc5e33c259a0fc.

[9] L. Reiber, *Mobile Forensic Investigations A Guide to Evidence Collection, Analysis, and Presentations*. McGraw-Hill Education, 2016.