RSA CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.

Capitalizing on
Collective Intelligence

Mobile Analysis Kung Fu, Santoku Style

SESSION ID: ANF-W03

Andrew Hoog

CEO/Co-founder
viaForensics
@ahoog42

Sebastián Guerrero

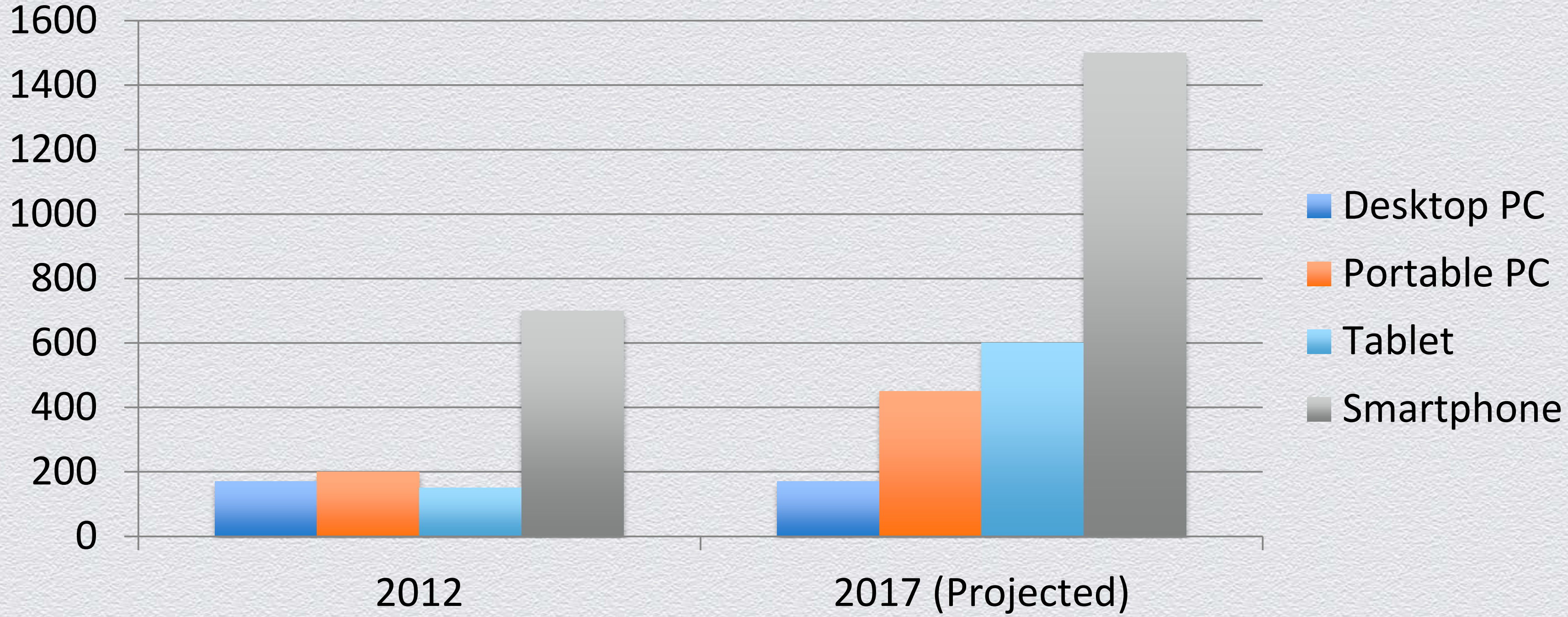
Mobile Security Analyst
viaForensics
@0xroot

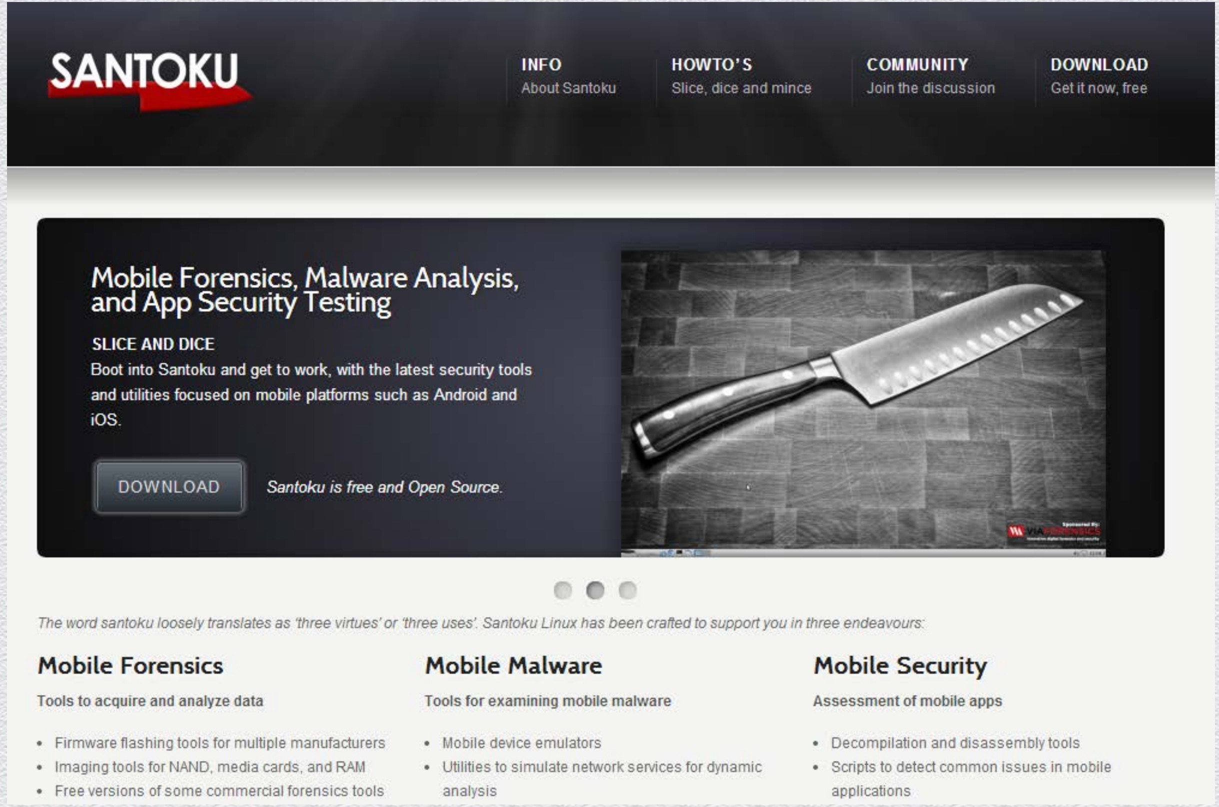
Agenda

- Santoku Intro
- Mobile Forensics Kung Fu
- Mobile Security Kung Fu
- Mobile Malware Analysis Kung Fu

Santoku Intro

Santoku – Why?


1600
1400
1200
1000
800
600
400
200
0
2012
2017 (Projected)
Desktop PC
Portable PC
Tablet
Smartphone



SANTOKU

INFO
About Santoku

HOWTO'S
Slice, dice and mince

COMMUNITY
Join the discussion

DOWNLOAD
Get it now, free

Mobile Forensics, Malware Analysis, and App Security Testing

SLICE AND DICE

Boot into Santoku and get to work, with the latest security tools and utilities focused on mobile platforms such as Android and iOS.

DOWNLOAD

Santoku is free and Open Source.

The word santoku loosely translates as 'three virtues' or 'three uses'. Santoku Linux has been crafted to support you in three endeavours:

Mobile Forensics

Tools to acquire and analyze data

- Firmware flashing tools for multiple manufacturers
- Imaging tools for NAND, media cards, and RAM
- Free versions of some commercial forensics tools

Mobile Malware

Tools for examining mobile malware

- Mobile device emulators
- Utilities to simulate network services for dynamic analysis

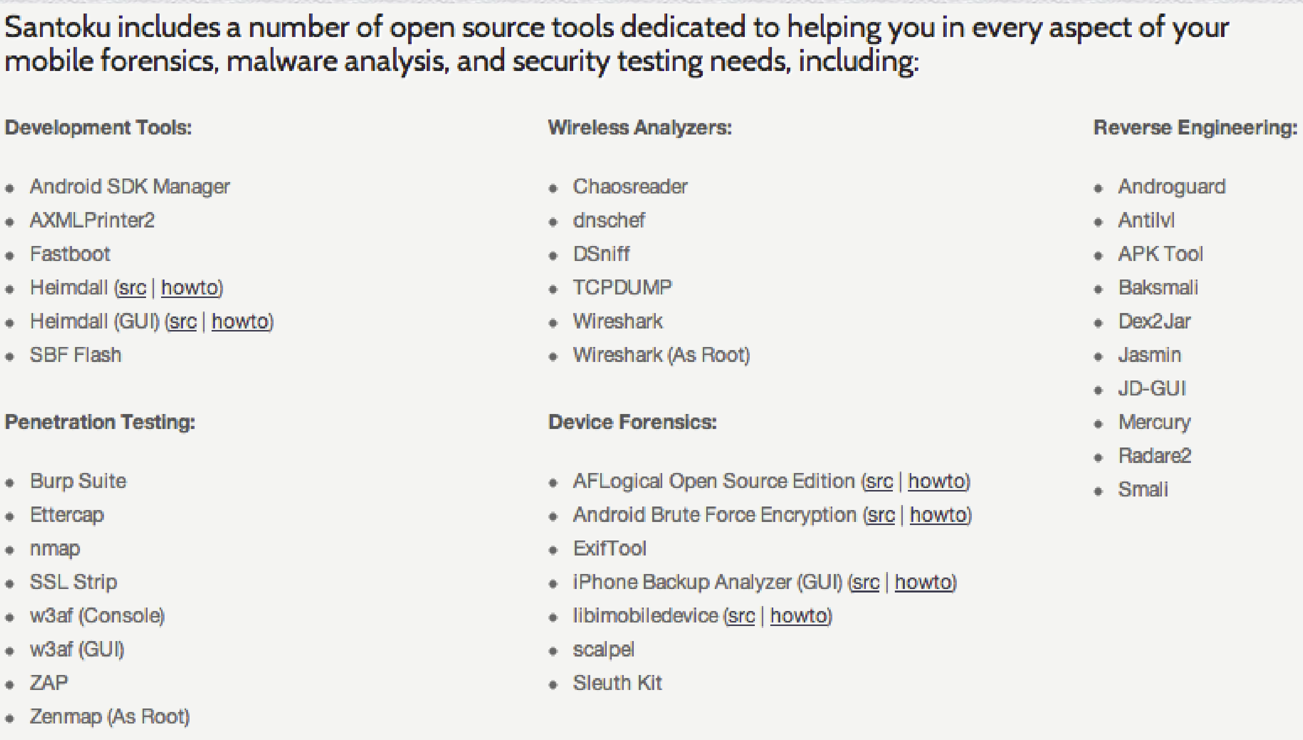
Mobile Security

Assessment of mobile apps

- Decompilation and disassembly tools
- Scripts to detect common issues in mobile applications

https://santoku-linux.com/ - It's Free!

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:

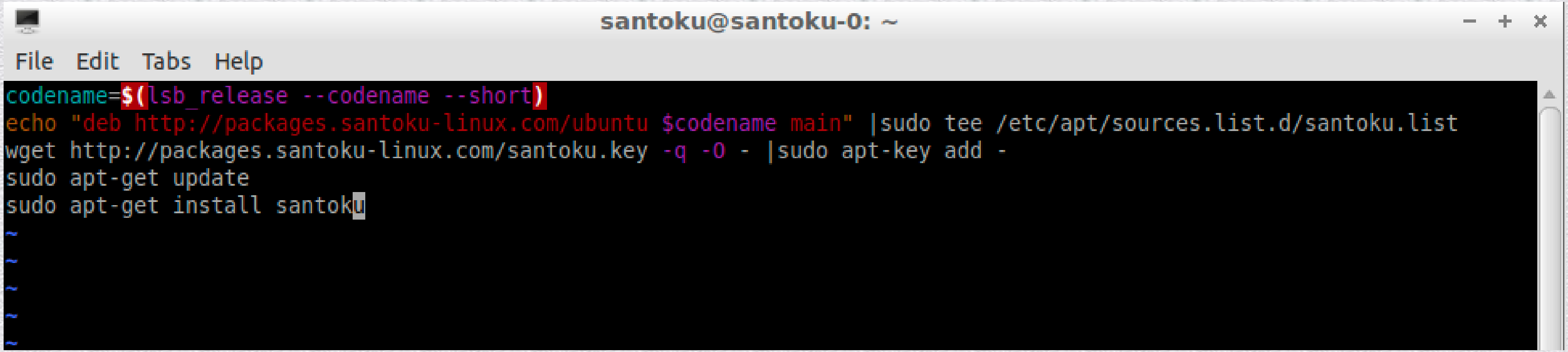
- AFLogical Open Source Edition (src | howto)
- Android Brute Force Encryption (src | howto)
- ExifTool
- iPhone Backup Analyzer (GUI) (src | howto)
- libimobiledevice (src | howto)
- scalpel
- Sleuth Kit

Reverse Engineering:

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

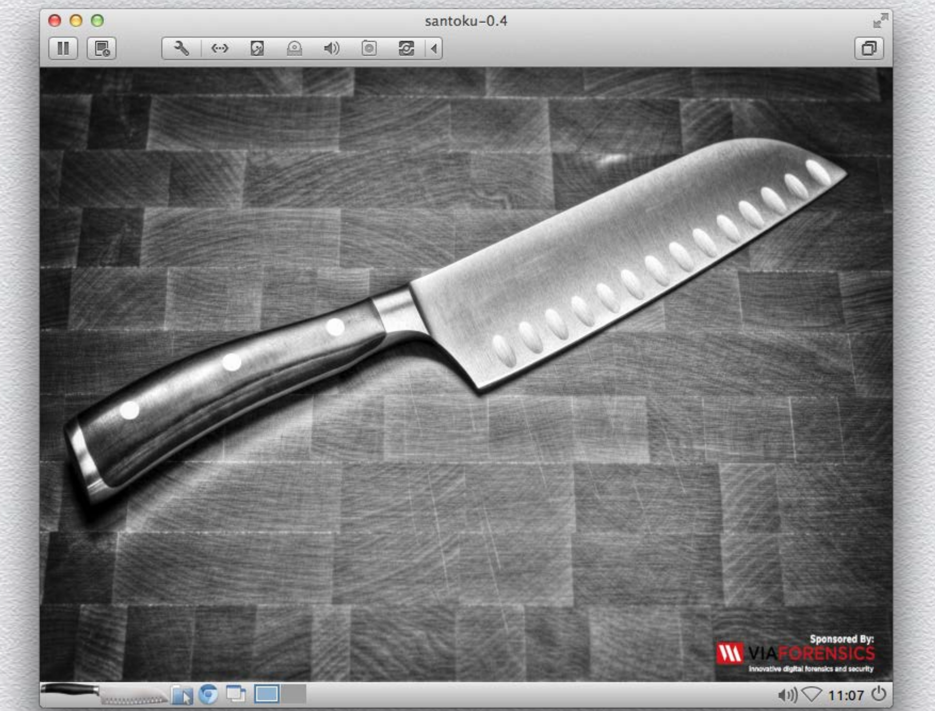
Santoku – How?

- Install Lubuntu 12.04 (precise) x86_64
- Santoku-ize it



```
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-get add -
sudo apt-get update
sudo apt-get install santoku
~
~
~
~
~
```

You should get (after reboot)

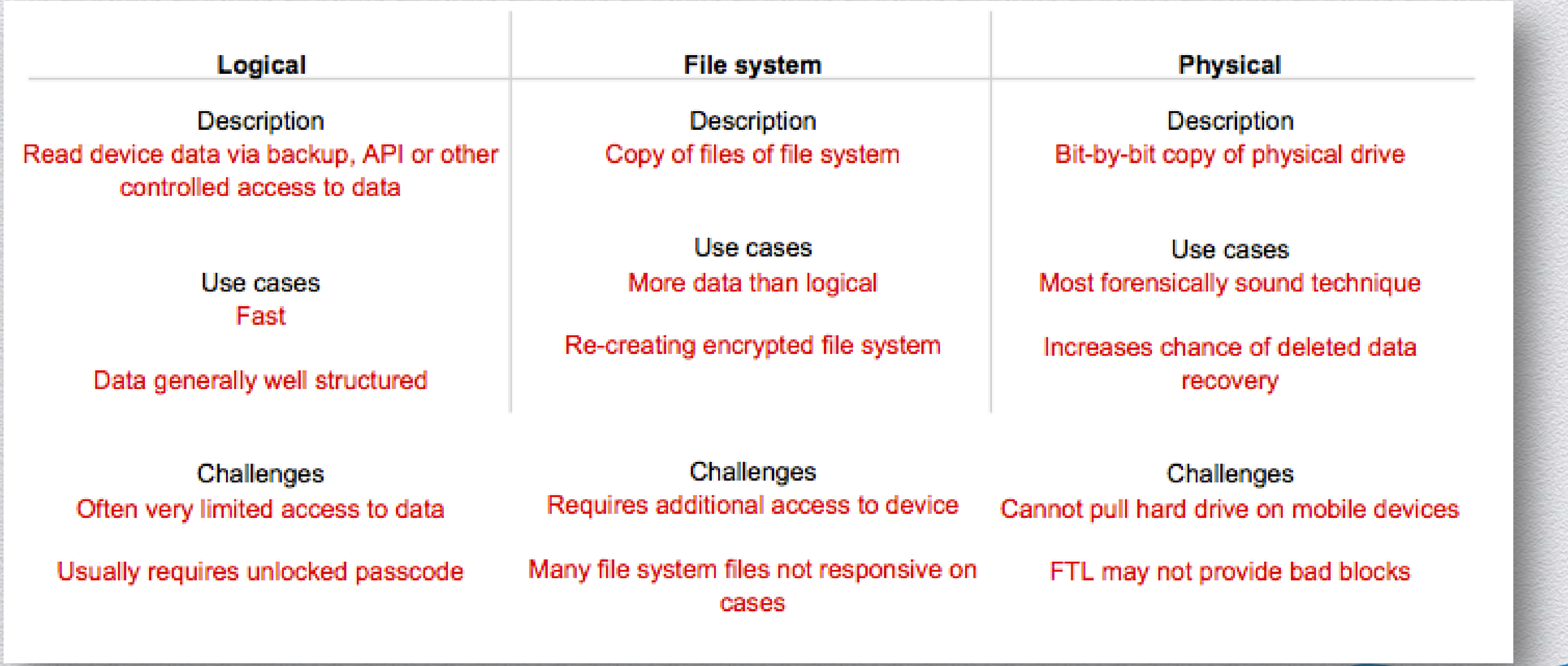


RSA CONFERENCE 2014

FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Mobile Forensics Kung Fu

Forensic Acquisition Types

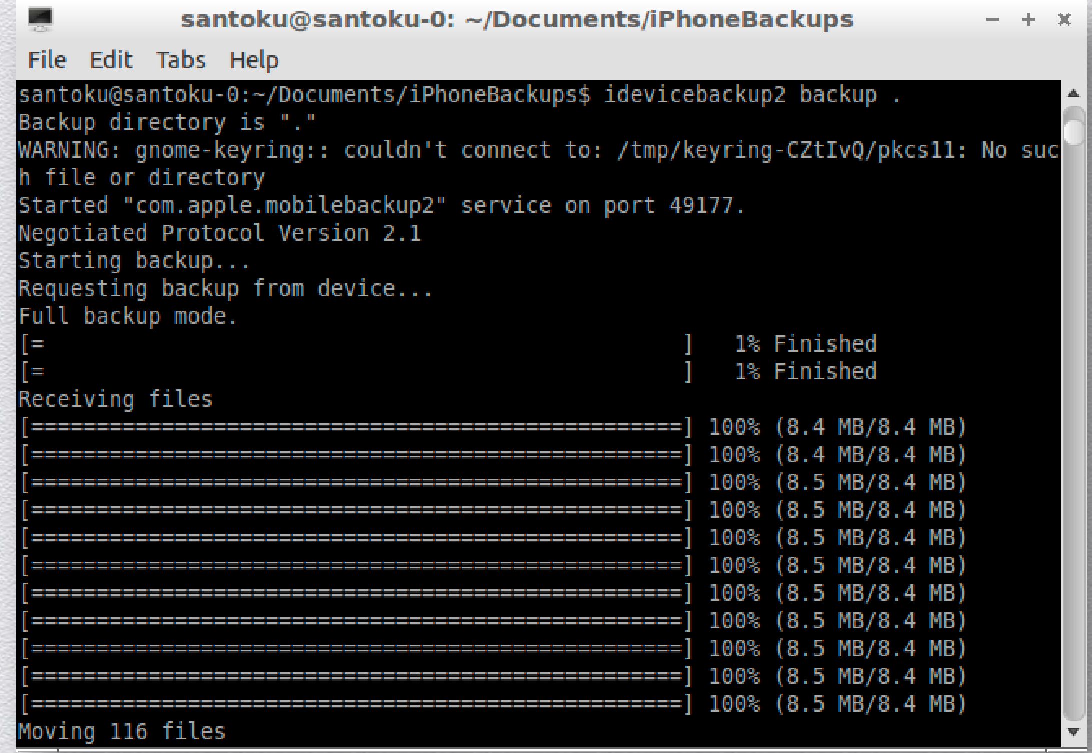


Logical	File system	Physical
Description Read device data via backup, API or other controlled access to data	Description Copy of files of file system	Description Bit-by-bit copy of physical drive
Use cases Fast Data generally well structured	Use cases More data than logical Re-creating encrypted file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

iOS Logical

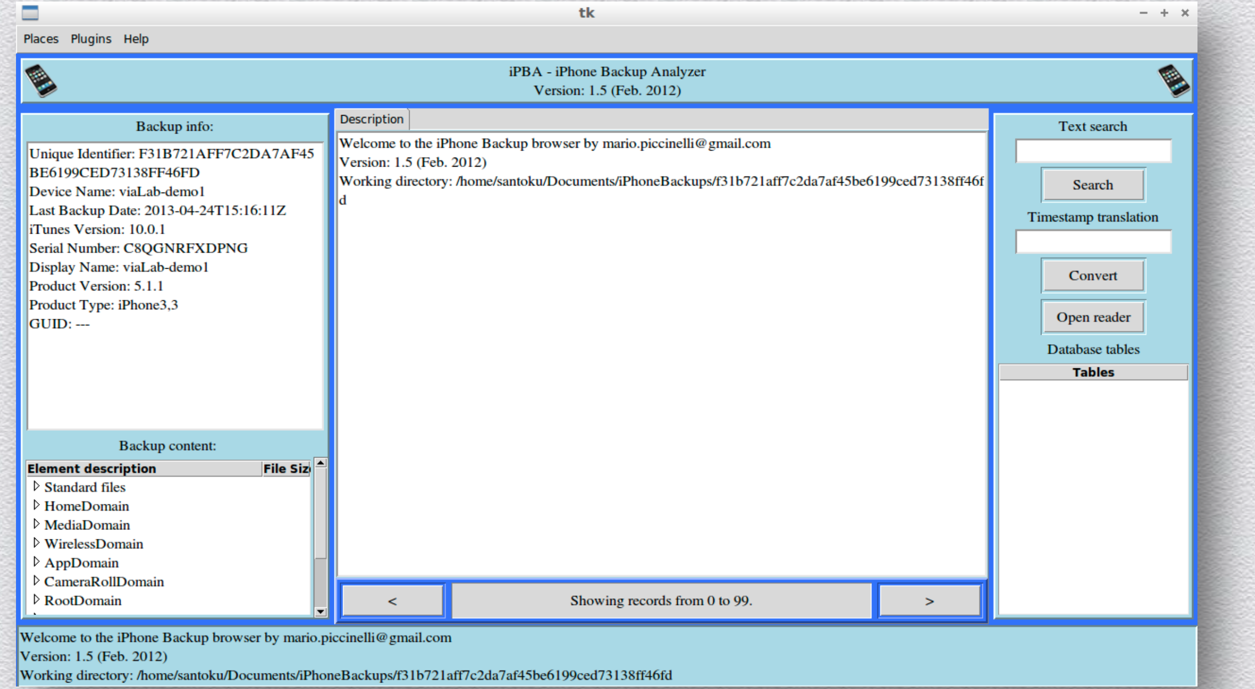
- Connect device (Enter PIN if needed)
- Ideviceback2 backup <backup dir>
- Ideviceback2 unback <backup dir>
- View backup|unpacked backup

iOS Logical



```
santoku@santoku-0:~/Documents/iPhoneBackups$ idevicebackup2 backup .
Backup directory is "."
WARNING: gnome-keyring:: couldn't connect to: /tmp/keyring-CZtIvQ/pkcs11: No suc
h 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)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
[==================================================] 100% (8.5 MB/8.4 MB)
Moving 116 files
```

iPhone Backup Analyzer



iPBA - iPhone Backup Analyzer
Version: 1.5 (Feb. 2012)

Backup info:

Unique Identifier: F31B721AFF7C2DA7AF45BE6199CED73138FF46FD
Device Name: viaLab-demo1
Last Backup Date: 2013-04-24T15:16:11Z
iTunes Version: 10.0.1
Serial Number: C8QGNRFXDPNG
Display Name: viaLab-demo1
Product Version: 5.1.1
Product Type: iPhone3,3
GUID: ---

Backup content:

- Standard files
- HomeDomain
- MediaDomain
- WirelessDomain
- AppDomain
- CameraRollDomain
- RootDomain

Welcome to the iPhone Backup browser by mario.piccinelli@gmail.com
Version: 1.5 (Feb. 2012)
Working directory: /home/santoku/Documents/iPhoneBackups/f31b721aff7c2da7af45be6199ced73138ff46fd

Showing records from 0 to 99.

Text search

Timestamp translation

Database tables

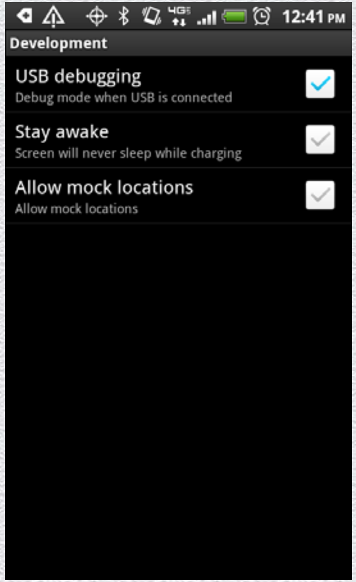
Tables

Welcome to the iPhone Backup browser by mario.piccinelli@gmail.com
Version: 1.5 (Feb. 2012)
Working directory: /home/santoku/Documents/iPhoneBackups/f31b721aff7c2da7af45be6199ced73138ff46fd

Android Logical

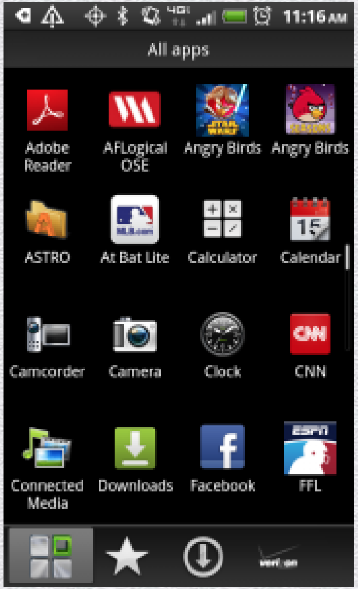
- AFLogical OSE (https://github.com/viaforensics/android-forensics)
- Reads Content-Providers
- Push to phone, run, store on SD-Card
- Pull CSVs to Santoku for review

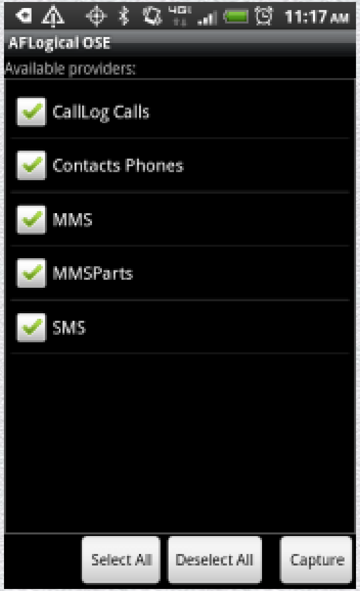
AFLogical OSE



Development

- USB debugging ☑
  Debug mode when USB is connected
- Stay awake ☐
  Screen will never sleep while charging
- Allow mock locations ☐
  Allow mock locations

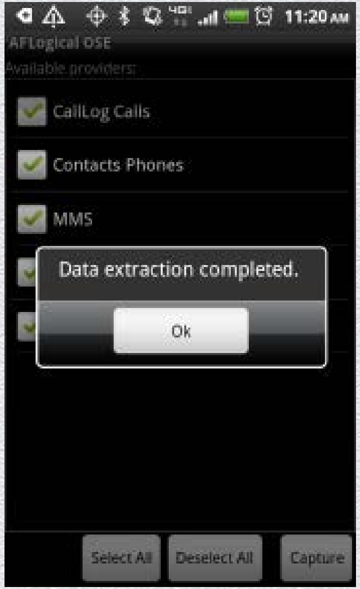




AFLogical OSE

Available providers:

- ☑ CallLog Calls
- ☑ Contacts Phones
- ☑ MMS
- ☑ MMSParts
- ☑ SMS



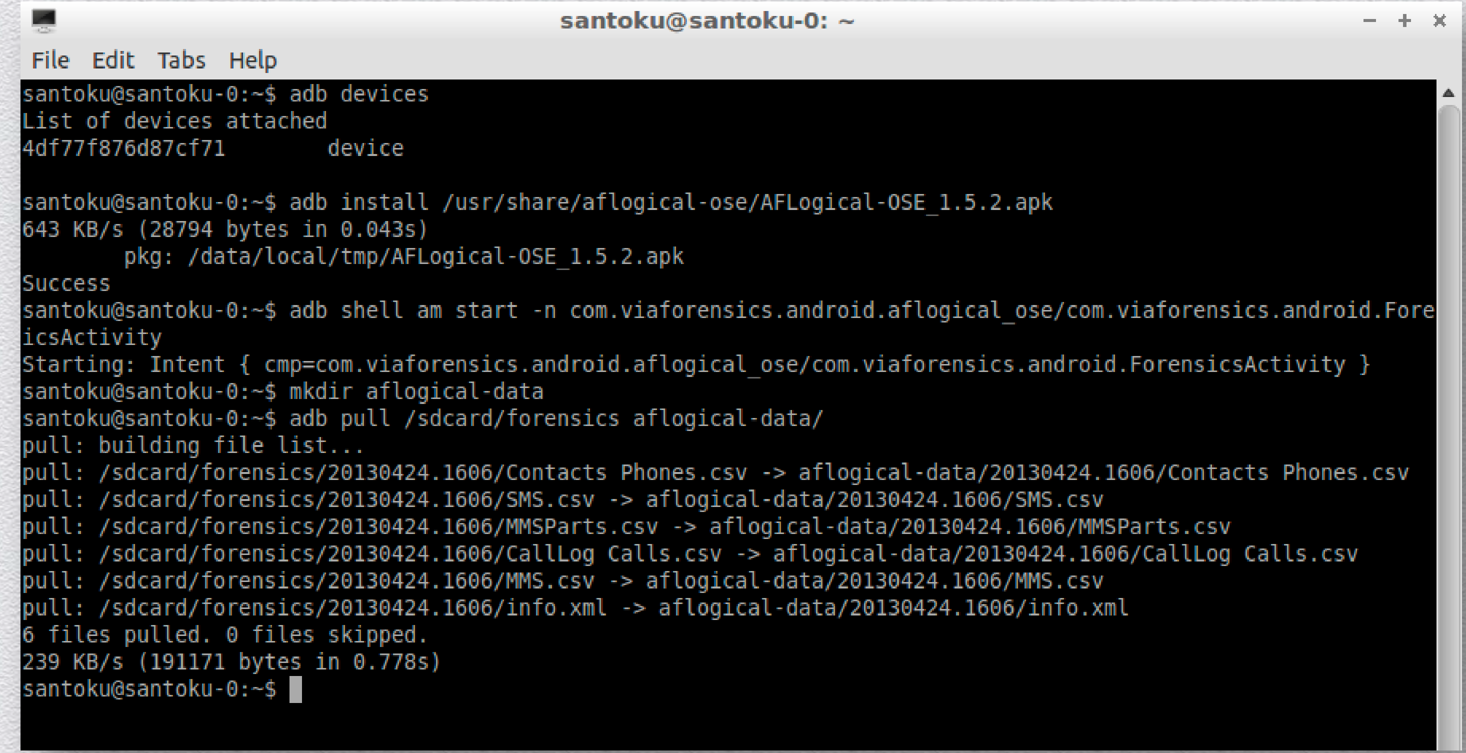
AFLogical OSE

Available providers:

- ☑ CallLog Calls
- ☑ Contacts Phones
- ☑ MMS

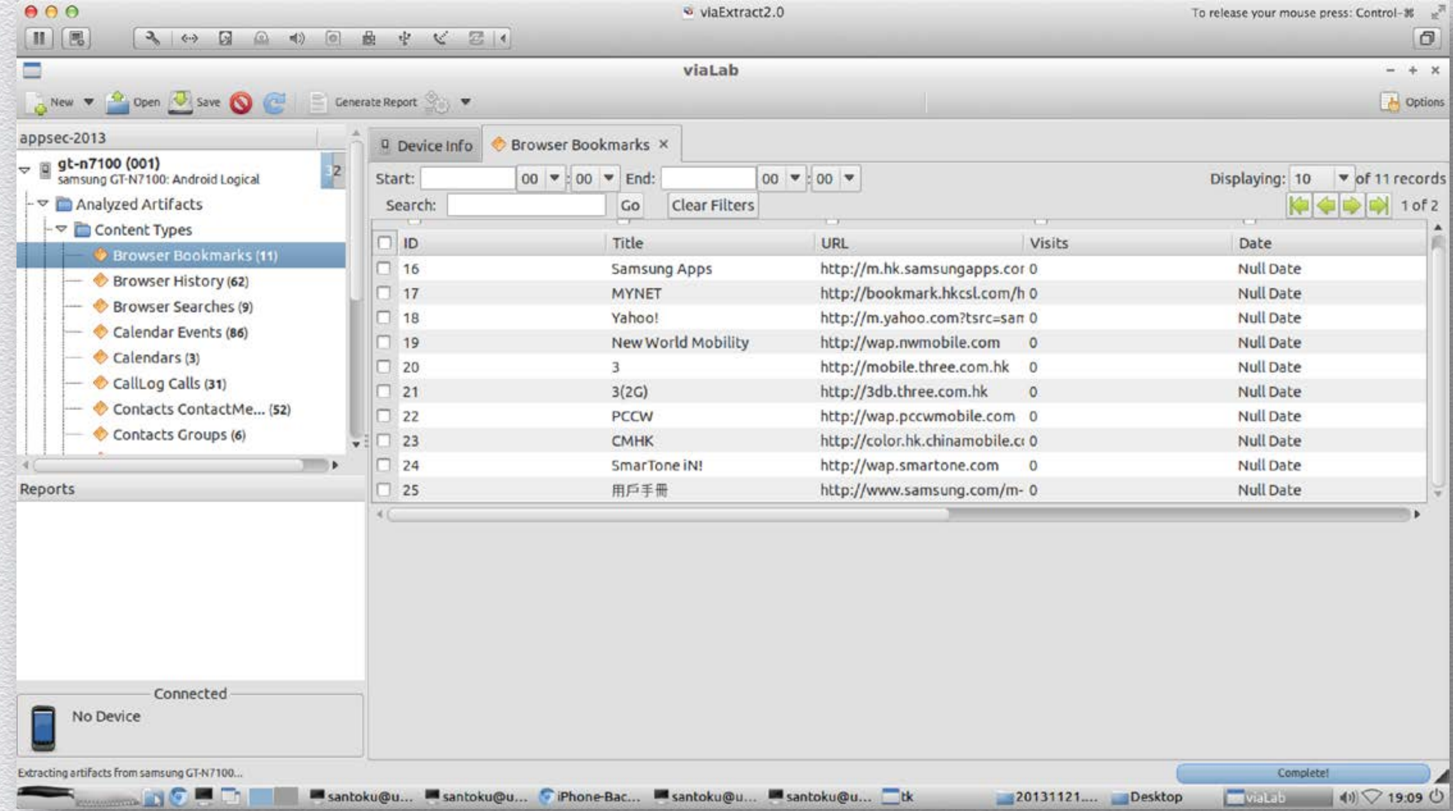
Data extraction completed.

Install, run, extract



```
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.Fore
icsActivity
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:~$
```

viaExtract



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)

Start: 00 : 00 End: 00 : 00 Displaying: 10 of 11 records

Search: Go Clear Filters 1 of 2

□	ID	Title	URL	Visits	Date
□	16	Samsung Apps	http://m.hk.samsungapps.cor	0	Null Date
□	17	MYNET	http://bookmark.hkcsl.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

Connected

No Device

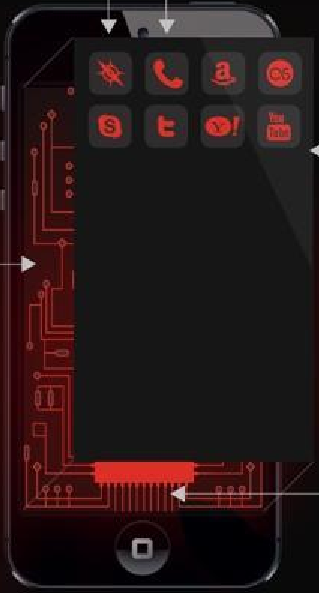
Extracting artifacts from samsung GT-N7100...

santoku@u... santoku@u... iPhone-Bac... santoku@u... santoku@u... tk 20131121.... Desktop viaLab 19:09

Mobile Security Kung Fu

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 Rootng
OS Data Caching
Passwords & Data Accessible
Carrier-Loaded Software
No Encryption/Weak Encryption
User-Initiated Code

MALWARE

THE INTERNET

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

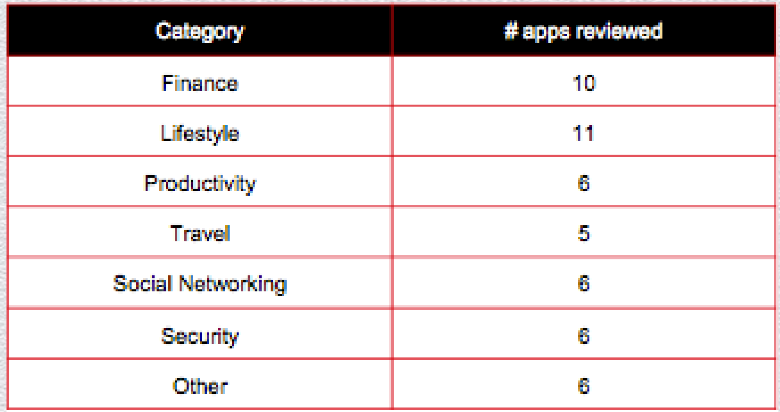
DATABASE

SQL Injection
Privilege Escalation
Data Dumping
OS Command Execution

VIAFORENSICS

App Selection

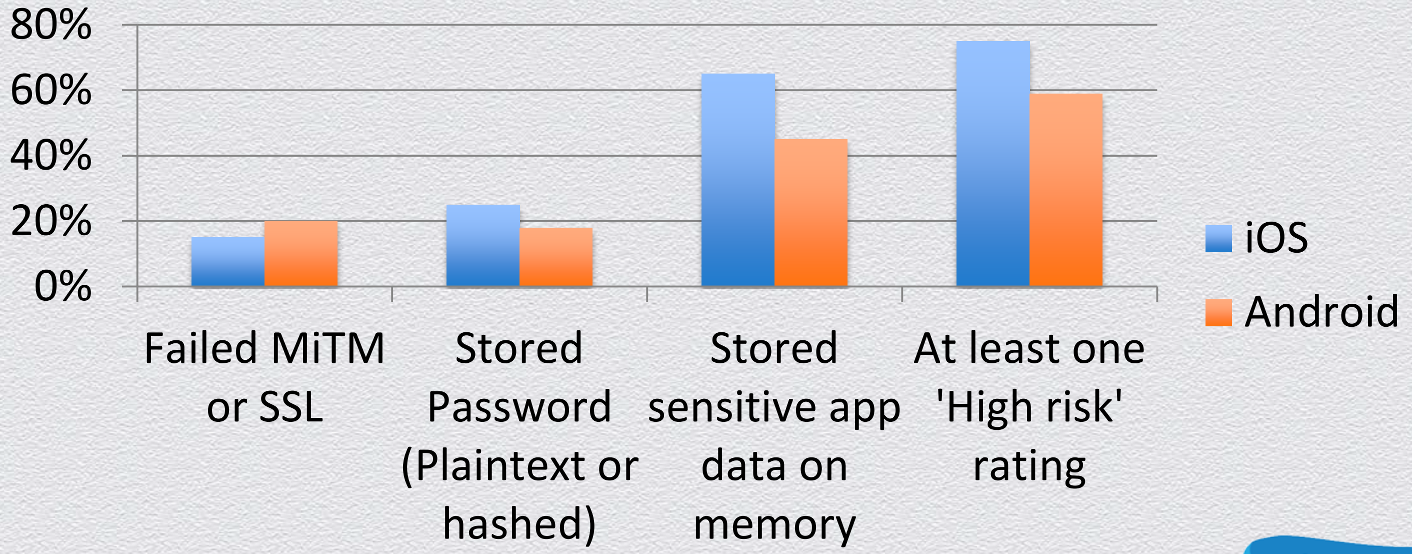
- Apps were selected based on popularity, number of downloads, or potential sensitivity of data
- Approximately 80 apps have been reviewed and organized into categories



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

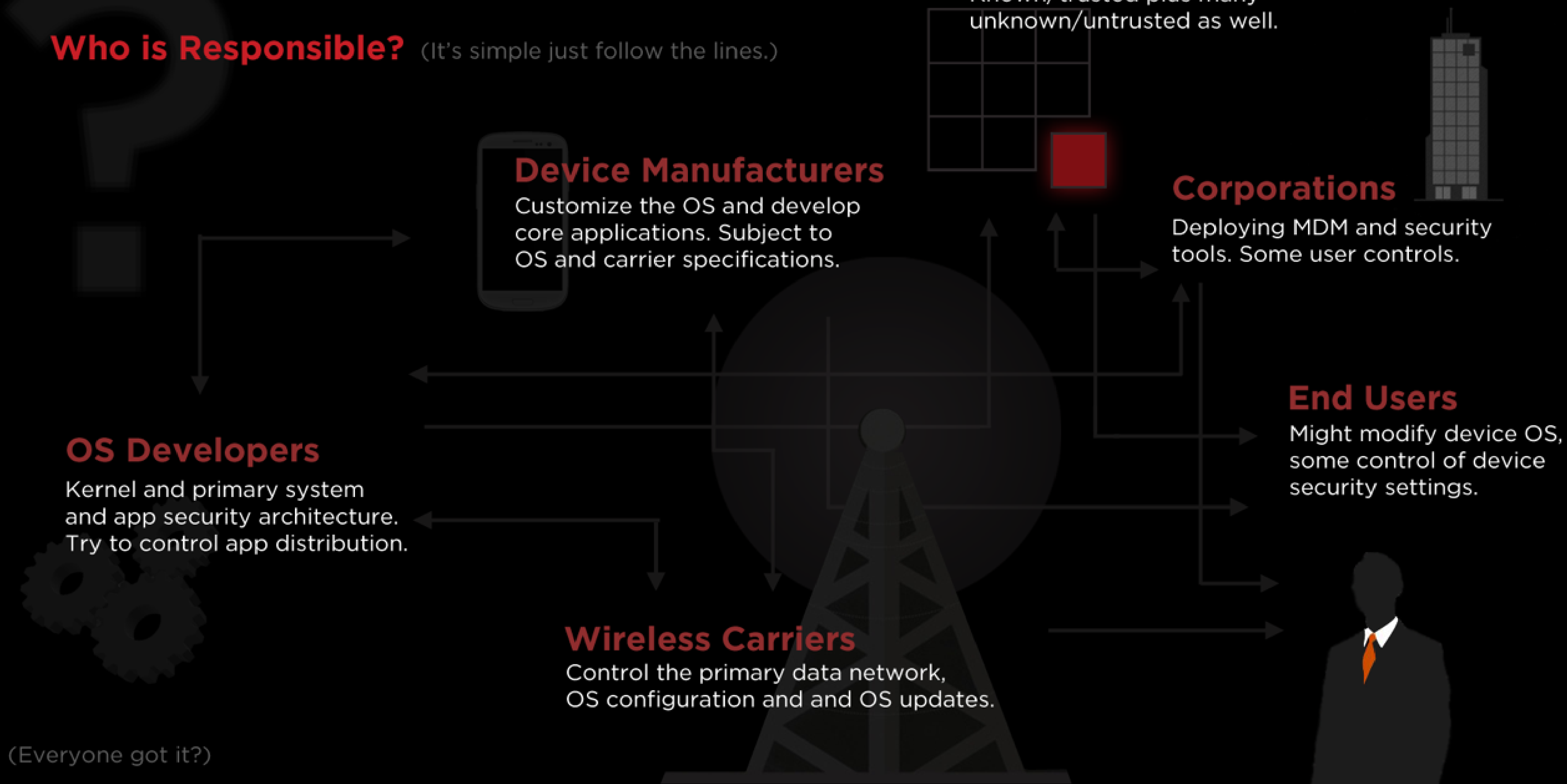
2013 App testing result

- 81 tested apps, 32 iOS, 49 Android



Mobile Device Security

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



App Developers

Known/trusted plus many unknown/untrusted as well.

Device Manufacturers

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

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 and OS updates.

(Everyone got it?)

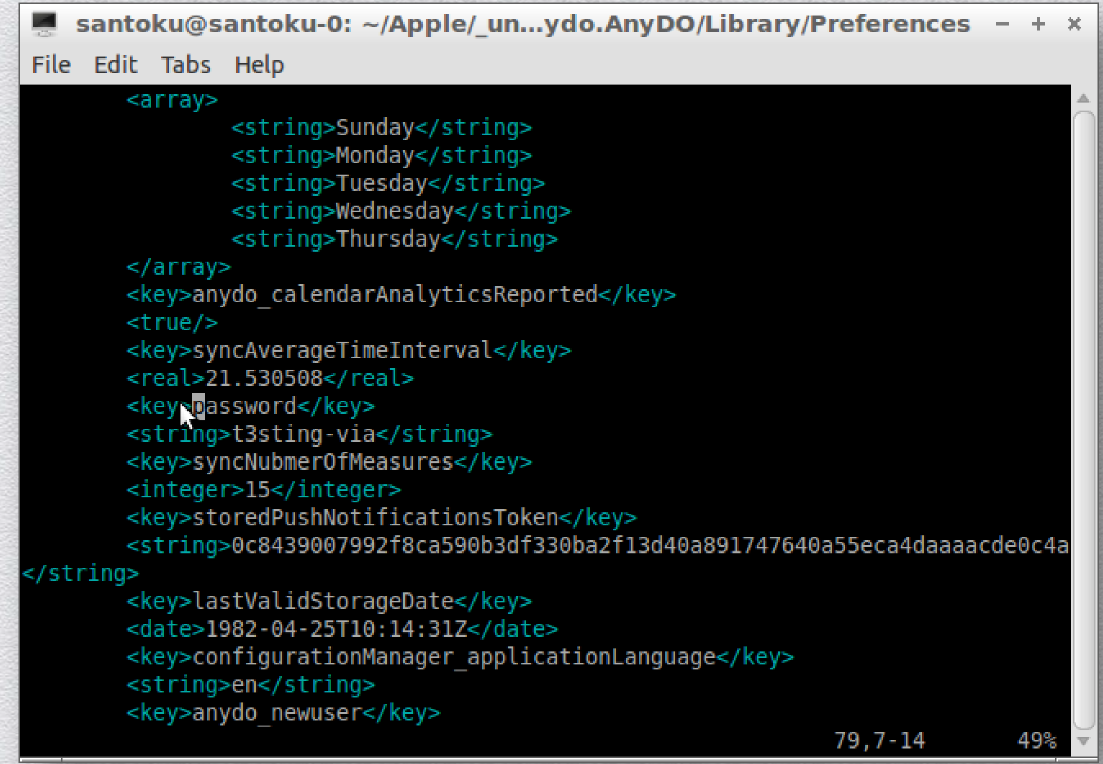
Any.Do

- Business and personal task management app iOS and Android
- Millions of users
- Many vulnerabilities, no response from company
- https://viaforensics.com/mobile-security/security-vulnerabilities-anydo-android.html

Any.Do Analysis - Forensics

- Locat Any.DO app directory
- Adb pull /data/data/com.anydo
- Examine database/binary files
- Capture network traffic

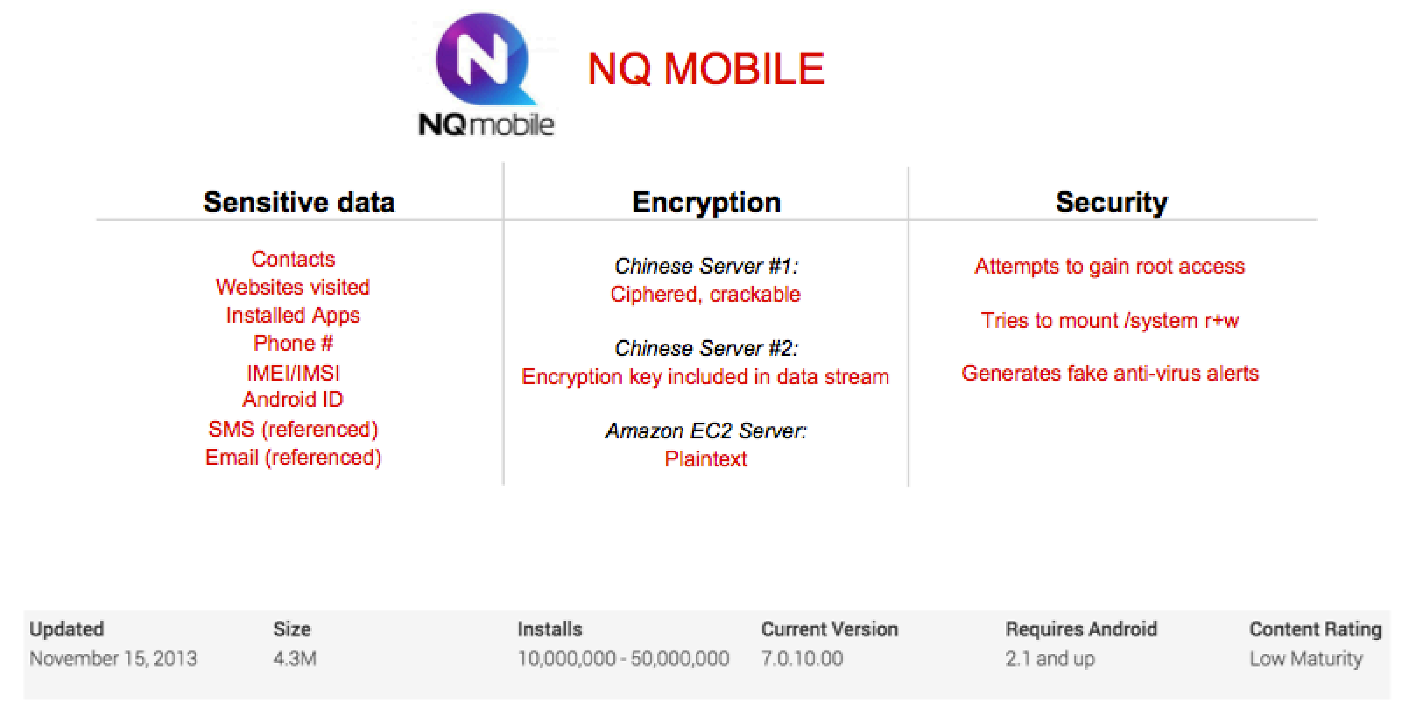
Any.Do Analysis - Forensics


santoku@santoku-0: ~/Apple/_un...ydo.AnyDO/Library/Preferences
File
Edit
Tabs
Help

```
        <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>syncNubmerOfMeasures</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>
                                                                     79,7-14       49%
```

Mobile Malware Analysis Kung Fu

NQ Mobile



NQmobile

NQ MOBILE

Sensitive data	Encryption	Security
Contacts Websites visited Installed Apps Phone # IMEI/IMSI Android ID SMS (referenced) Email (referenced)	*Chinese Server #1:* Ciphered, crackable *Chinese Server #2:* Encryption key included in data stream *Amazon EC2 Server:* Plaintext	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

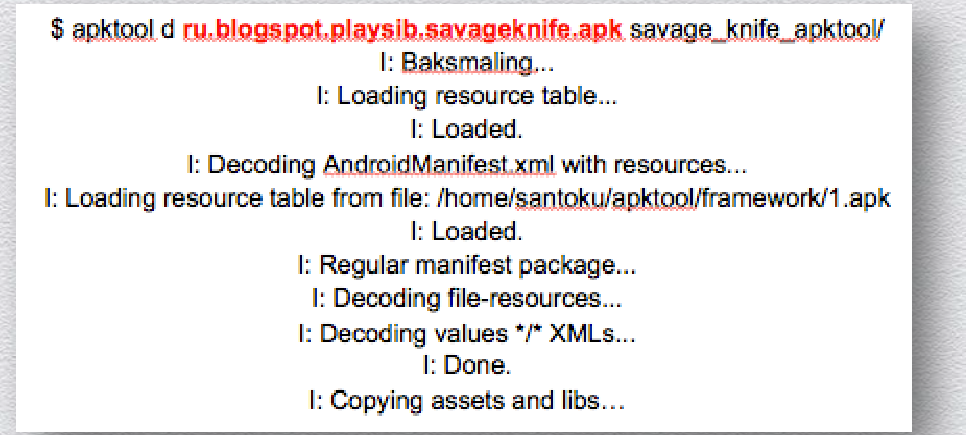
- Android Malware, masquerades as an innocent advertising network
- Packaged in many legitimate apps, usually targeting Russian market
- Has ability to download additional apps, and propmts the user to install them, posing as “Critical Updates”. Uses this mechanism to spread known malware, typically 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

- Tool for reverse engineering Android apk
- Dissasembles code to smali files, also decodes resources contained into the apk.
- It can also repackage the applications after you have modified them
- We can run it on Badnews

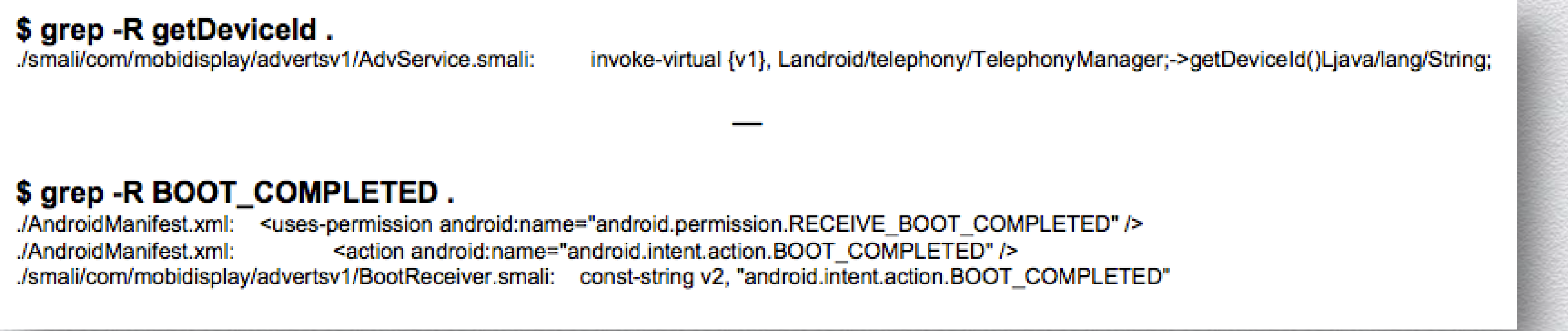
Badnews Sample



```
$ 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...
```

From apktool to smali

- We can grep for known sensible method calls and strings

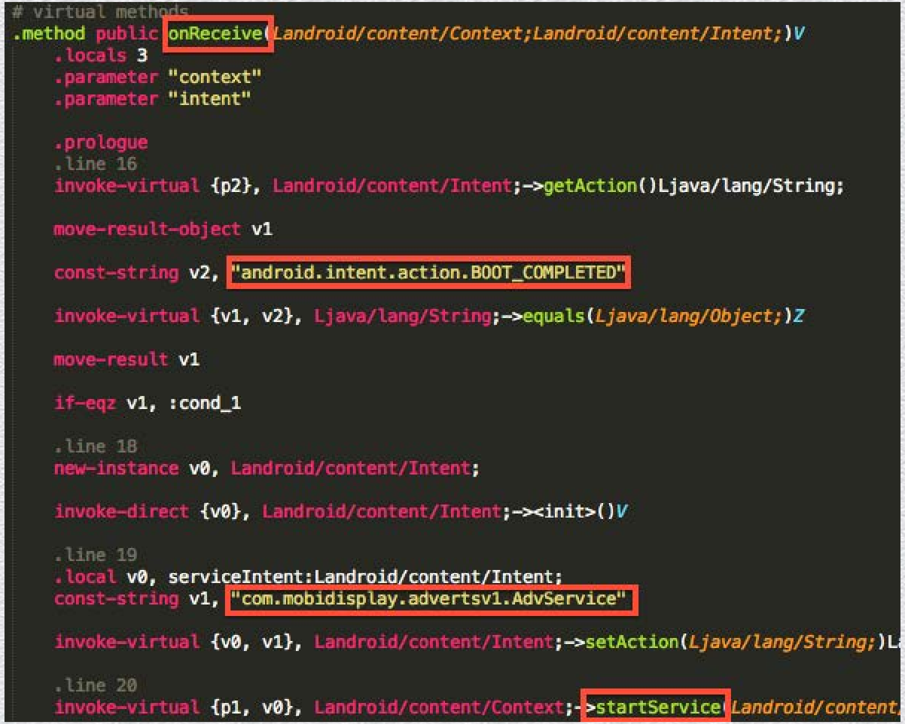


```
$ grep -R getDeviceId .
./smali/com/mobidisplay/advertsv1/AdvService.smali:        invoke-virtual {v1}, Landroid/telephony/TelephonyManager;->getDeviceId()Ljava/lang/String;
```

```
$ 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"
```

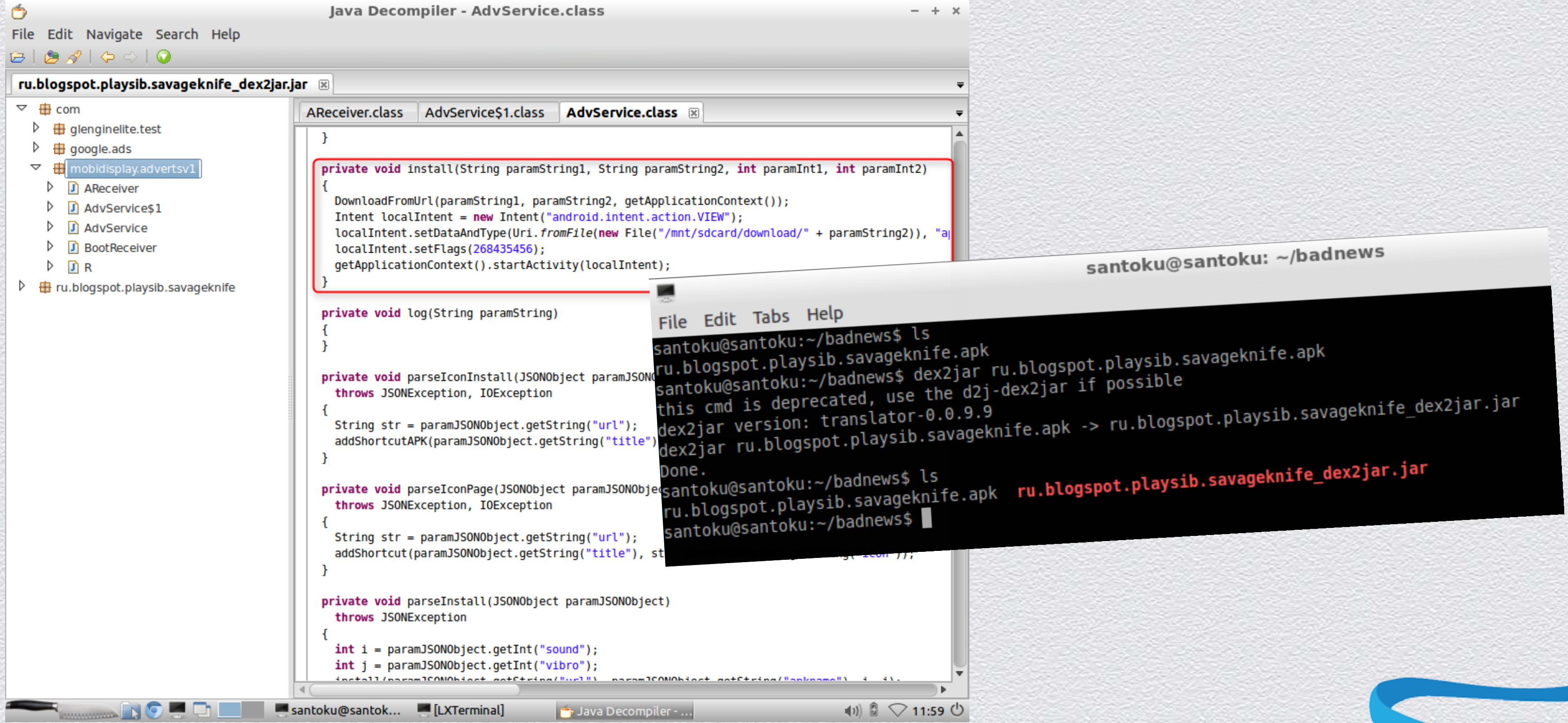
From apktool to smali

- We can manually analyze the disassembled smali coded 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;)L

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

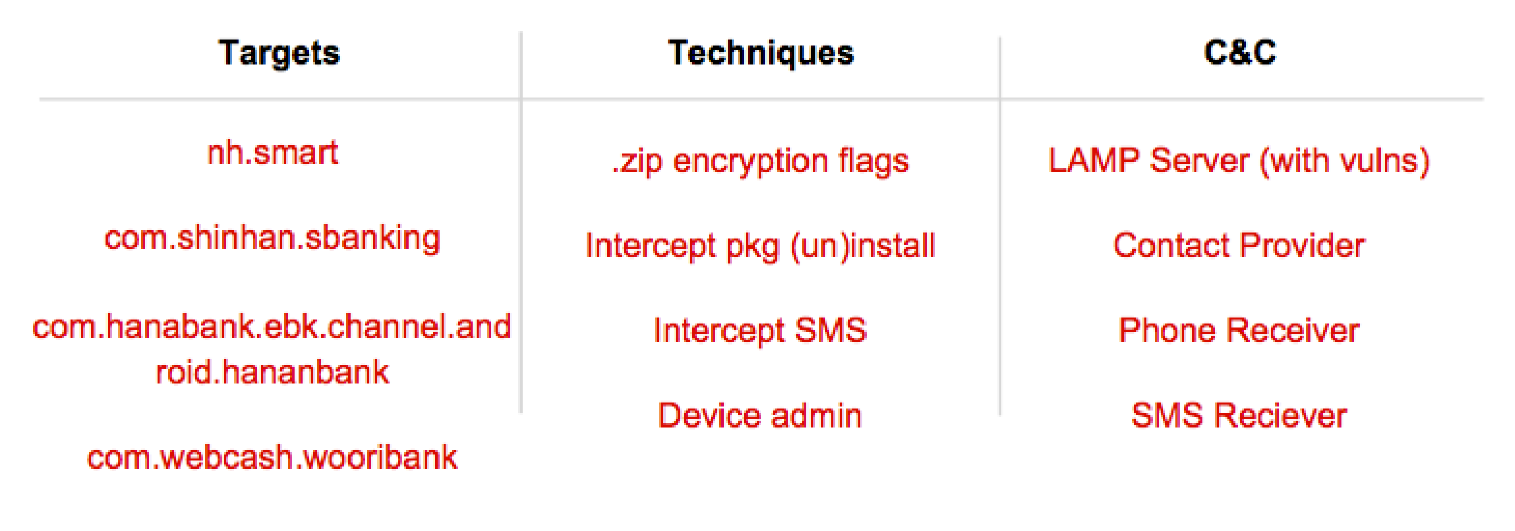
Badnews sample – Dex2Jar - JDGui



```
  }

  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)), "a
    localIntent.setFlags(268435456);
    getApplicationContext().startActivity(localIntent);
  }

  private void log(String paramString)
  {
  }

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

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

  private void parseInstall(JSONObject paramJSONObject)
    throws JSONException
  {
    int i = paramJSONObject.getInt("sound");
    int j = paramJSONObject.getInt("vibro");
```

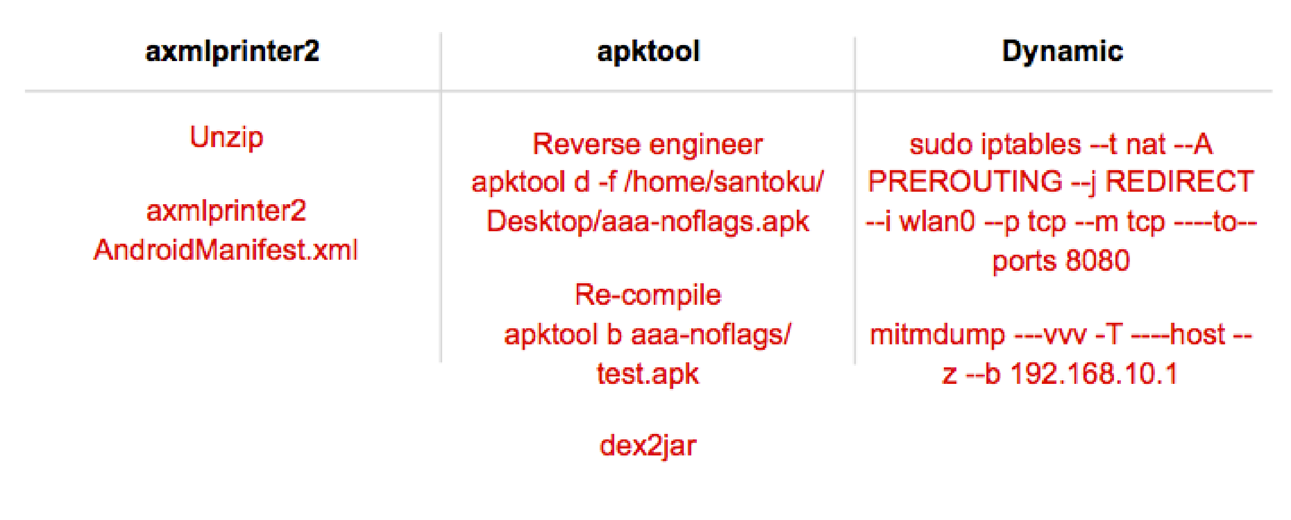
```
santoku@santoku:~/badnews$ ls
ru.blogspot.playsib.savageknife.apk
santoku@santoku:~/badnews$ dex2jar ru.blogspot.playsib.savageknife.apk
this cmd is deprecated, use the d2j-dex2jar if possible
dex2jar version: translator-0.0.9.9
dex2jar ru.blogspot.playsib.savageknife.apk -> ru.blogspot.playsib.savageknife_dex2jar.jar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

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 Re-compile apktool b aaa-noflags/test.apk dex2jar	sudo iptables --t nat --A PREROUTING --j REDIRECT --i wlan0 --p tcp --m tcp ----to--ports 8080 mitmdump ---vvv -T ----host --z --b 192.168.10.1

A little help fu, please

- HOWTOs
- New/existing tool development
- .deb package maintenance
- Forums, spreading the word

Q&A | Contact | Feedback

- Thanks for listening…

@0xroot / @ahoog42

github/0xroot / github/viaforensics

sguerrero@viaforensics.com / ahoog@viaforensics.com