Toward a Trustworthy Android Ecosystem

Yan Chen
Lab of Internet and Security Technology (LIST)
Northwestern University, USA
Smartphone Security

• Ubiquity - Smartphones and mobile devices
  - Smartphone sales already exceed PC sales
  - The growth will continue

• Performance better than PCs of last decade
  - Samsung Galaxy S6 Edge: 1,440x2,560 pixel display, 2.0 GHz 64bit octa-core, 3GB memory

---

### Worldwide smart phone and client PC shipments

<table>
<thead>
<tr>
<th>Category</th>
<th>Q4 2011 shipments (millions)</th>
<th>Growth Q4’11/Q4’10</th>
<th>Full year 2011 shipments (millions)</th>
<th>Growth 2011/2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart phones</td>
<td>158.5</td>
<td>56.6%</td>
<td>487.7</td>
<td>62.7%</td>
</tr>
<tr>
<td>Total client PCs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pads</td>
<td>26.5</td>
<td>186.2%</td>
<td>63.2</td>
<td>274.2%</td>
</tr>
<tr>
<td>- Netbooks</td>
<td>6.7</td>
<td>-32.4%</td>
<td>29.4</td>
<td>-25.3%</td>
</tr>
<tr>
<td>- Notebooks</td>
<td>57.9</td>
<td>7.3%</td>
<td>209.6</td>
<td>7.5%</td>
</tr>
<tr>
<td>- Desktops</td>
<td>29.1</td>
<td>-3.6%</td>
<td>112.4</td>
<td>2.3%</td>
</tr>
</tbody>
</table>

Source: Canalys estimates © Canalys 2012
Mobile Devices (apps) Dominate

Number of Global Users (Millions)

Apps Continue to Dominate the Mobile Web

Percentage of time spent

2013 2014

80% 86%

Source: Morgan Stanley Research

Source: Flurry Analytics
Android is Leading the Pack

WORLDWIDE SMARTPHONE SHIPMENTS FORECAST BY OS

- **Android**: 1164.3 million (2015), 1541.9 million (2019), 7.80% CAGR over 5 years
- **iOS (iPhone)**: 223.7 million (2015), 269.6 million (2019), 7% CAGR
- **Windows Phone**: 36.9 million (2015), 67.8 million (2019)
- **Others**: 11.5 million (2015), 23 million (2019)

SOURCE: IDC TRACKER, AUGUST 2015
NOTE: FIGURES IN MILLIONS
Android Ecosystem

Carriers
- AT&T
- Verizon
- Sprint
- T-Mobile

Vendors
- Samsung
- LG
- Motorola

Application Stores
- Google Play

Devices and OS

Applications

Security Vendors
- Kaspersky
- Norton
- AVG
- McAfee
- ESET
- NOD32

Developers

Users
Android Threats

• Malware and vulnerabilities
  – The numbers are increasing consistently
  – Anti-malware ineffective at catching zero-day and polymorphic malware

• Information Leakage
  – Users have no way to know when and what info is being leaked out of their device to whom
  – Even legitimate apps leak private info though the user may not be aware

• Fraud activities (esp. for mobile payment)
Privacy Leakage

- Android permissions are insufficient
  - User still does not know if some private information will be leaked
- Information leakage is more dangerous than information access
  - Example 1: popular apps (e.g., Angry Birds) leak location info with its developer, advertisers and analytics services
    - Even doesn’t need it for its functionality!
  - Example 2: malware apps may steal private data
    - A camera app trojan send video recordings out of the phone
123,255 eCommerce customers with >$0 fraud in March 2014, by LexisNexis
New Challenges & Opportunities

• Centralized control
  – Vet applications before they enter store
  – Carriers may have more complete pictures of users and traffic

• Apps are much easier to analyze statically
  – Use of Dalvik bytecode instead of x86

• Constrained environment
  – CPU, memory, battery
  – User perception
Problems and Our Solutions

• Issues for existing mobile anti-virus systems
  – Easy to evade [DroidChamelon]
  – Unable to detect native malware [DroidNative]
  – Unable to detect malware in ads or dynamically loaded content [AdShield]

• Privacy leakage detection and prevention
  – How to find questionable sensitive permissions [AutoCog]
  – Real time tracking & preventing privacy leakage on phone
    • Consumer [PrivacyShield]
    • Enterprise Mobility Management (EMM) [AppShield]

• Fraud detection mostly with app-level risk management [DroidCog]
  – Duplicate detection
  – Privacy infringement
Systems Developed

• **AppsPlayground** [ACM CODASPY’13]
  – Automatic, large-scale dynamic analysis of Android apps
  – System released with hundreds of download

• **DroidChamelon** [ACM ASIACCS’13, IEEE Transaction on Information Forensics and Security 14]
  – Evaluation of latest Android anti-malware tools
  – All can be evaded with transformed malware
  – System released upon wide interest from media and industry
Impact of DroidChameleon

Interest from vendors
System Developed: AutoCog

Check whether sensitive permissions requested by apps are consistent with its natural-language description.

<table>
<thead>
<tr>
<th>App Name</th>
<th>Description (permissions)</th>
</tr>
</thead>
</table>
| Memory                                  | 1. The pro version also has no adds  
2. Please send us an email or leave a review if you have any suggestions  
3. Math Helper is the perfect app to help you with daily math tasks that your calculator can’t help you with  
4. It has a n-th root calculator, a percent calculator when you need to check if the sale at your nearest store really is as good as they say  
5. The app includes a prime number checker with bonus features, a calculator for circle, sphere and |
Systems Developed: PrivacyShield

- Real-time information-flow tracking for privacy leakage detection
- App instrumentation, with zero platform modification
- App released in Google play and Baidu stores
ARE THESE ADS SAFE: DETECTING HIDDEN ATTACKS THROUGH MOBILE APP-WEB INTERFACES
Consider This...

Faked threat report

Click on the button(s)

Downloaded phishing app
The Problem

- Enormous effort toward analyzing malicious applications
- App may itself be benign
  - But may lead to malicious content through links
- **App-web interface**
  - Links inside the app leading to web-content
  - Not well-explored
- Types
  - Advertisements
  - Other links in app
App-Web Interface

Characteristics

- Can be highly dynamic
- A link may recursively redirect to another before leading to a final web page
- Links embedded in apps
  - Can be dynamically generated
  - Can lead to dynamic websites
- Advertisements
  - Ad libraries create links dynamically
  - Ad economics can lead to complex redirection chains
Advertising Overview

AdColony
admob
millennialmedia
友盟
inMobi

Advertisers  Ad networks  Apps / Developers Users
Ad Networks

• Ad libraries act as the interface between apps and ad network servers

• Ad networks may interface with each other
  – Syndication – One network asks another to fill ad space
  – Ad exchange – Real-time auction of ad space

• App or original ad network may not have control on ads served
Solution Components

- **Triggering**: Interact with app to launch web links

- **Detection**: Process the results to identify malicious content

- **Provenance**: Identify the origin of a detected malicious activity
  - Attribute malicious content to domains and ad networks
Triggering

- Use AppsPlayground
  - A gray box tool for app UI exploration
  - Extracts features from displayed UI and iteratively generates a UI model

- A novel computer graphics-based algorithm for identifying buttons
  - See widgets and buttons as a human would

---

Detection

- Automatically download content from landing pages
- Use VirusTotal for detecting malicious files and URLs
Provenance

• How did the user come across an attack?
• Code-level attribution
  – App code
  – Ad libraries
    • Identified 201 ad libraries
• Redirection chain-level attribution
  – Which URLs led to attack page or content
Results

- Deployments in US and China
- 600 K apps from Google Play and Chinese stores
  - 91, Anzhi (安智), AppChina (应用汇), Mumayi (木蚂蚁)
- 1.4 M app-web links triggered
- 2,423 malicious URLs
- 706 malicious files
Case Study: Fake AV Scam

- Multiple apps, one ad network: Tapcontext
- Ad network solely serving this scam campaign
- Phishing webpages detected by Google and other URL blacklists about 20 days after we detected first instance
Case Study: Free iPad Scam

- Asked to give personal information without any return
- New email address receiving spam ever since
- Origins at Mobclix and Tapfortap
  - Ad exchanges
  - Neither developers nor the primary ad networks likely aware of this
Case Study: iPad Scam from static link

- Another Scam, this time through a static link embedded in app
- Link target opens in browser and redirects to scam
- Not affiliated with Facebook
Case Study: SMS Trojan Video Player

- Ad from nobot.co.jp leads to download a movie player
- Player sends SMS messages to a premium number without user consent
Limitations

• Incomplete detection
  – Antiviruses and URL blacklists are not perfect
  – Our work DroidChameleon\textsuperscript{2} shows this

• Incomplete triggering
  – App UI can be very complex
  – May still be sufficient to capture advertisements

Conclusion and Ongoing Work

• Benign apps can lead to malicious content
• First large scale study to detect malicious ads on Android
• Making it a 24 * 7 service
• Working with ad network providers (e.g., Baidu and Google) and CNCERT for defense
• Only the tip of iceberg, security issues on dynamic code loading (DCL)
  – Detected malware and vulnerabilities that Google Bouncer missed
DROIDCOG: DEVICE-LEVEL MOBILE RISK MANAGEMENT
Motivations

- The growing popularity of mobile payment
- Attack surface of smartphone user's financial
- Countermeasures:
  - G1: authentication, explicit
  - G2: risk management, implicit/explicit
- Heavy usage of user privacy (e.g., !A#"-4/3
  - Application-level: duplicate data, redundant detection
A learning-based mechanism for user fraud detection

- Least user privacy required, high detection accuracy
- Device-level approach: only one copy of data is uploaded
- Robust, hard to evade
Goal

Decision

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>...</th>
<th>Classifiers</th>
</tr>
</thead>
</table>

Training

Data preprocess

Periodical upload

Mobile payment vendors

Server

Data collection
Motion sensors
U1

Data collection
Motion sensors
U2

Client
Problem Statement

Fingerprinting Bob’s usage manner

Verify based on classification results
Challenges

- Lack of features
- Data availability
- Imbalanced dataset
- Noise surrounding
- Unlabeled data
Challenges

• Lack of features
  – Only based on acceleration sensor and gyroscope sensor
  – Feature selection (6 values $\rightarrow$ 64 features)

• Data availability

• Imbalanced dataset

• Noise surrounding

• Unlabeled data
Challenges

- Lack of features
- Data availability
  - Periodical data collection
  - User activity detection
- Imbalanced dataset
- Noise surrounding
- Unlabeled data
Challenges

- Lack of features
- Data availability
- Imbalanced (classification) dataset
  - Control of distribution of training set
  - Random selection & stratified sampling
- Noise surrounding
- Unlabeled data
Challenges

- Lack of features
- Data availability
- Imbalanced dataset
- Noise surrounding
  - Calibrate sensor data based on gravity direction
  - Identify user motion state: static or in motion?
- Unlabeled data
Challenges

- Lack of features
- Data availability
- Imbalanced dataset
- Noise surrounding
- Unlabeled data
  - Semi-supervised online learning
Data Preprocessing

- Filter useless data on client side
  - The device is put on a flat plane
- Identify motion state on server

Each motion state has one corresponding

![Graphs showing acceleration on different axes for different motion states.](image-url)
Training Set Construction

- Resolve the issues of imbalanced dataset
- The data samples from other users are representative
- Preserve the temporal continuity
- Random selection v.s. stratified sampling
  - Similar performance
  - No cost of grouping user data for random selection
ML Algorithm Selection

- Expectation Maximization (EM): slow
- J48 decision tree: training set over fit, extra cost of tree pruning
- Logistic regression: cannot handle non-linear boundary
- SVM – Fast, handle non-linear classification boundary
Semi-supervised Online Learning

Old classifier

Online learning

New classifier

Training

Test

New data

Validation

Performance drop?

Yes

No

Rollback

Commit
Preliminary Evaluation

• Data
  – Collected with “Phone manager” (手机管家) by Tencent
  – 1st batch dataset: 210 users
  – 2nd batch dataset: 1516 users

• Metrics
  – Accuracy
    • True positive: owner is correctly identified
    • False positive: other is incorrectly identified as owner
    • False negative: owner is incorrectly identified as other
    • True negative: other is correctly identified
    • $R_{owner} = \frac{TP}{TP+FN}$, $R_{other} = \frac{TN}{TN+FP}$
    • ROC curve
  – Overhead
  – Robustness
• 80 users with full data; each user has 4K samples in training set and 1.2K samples in test set.

\[ R_{\text{other}}: 86.44\% \]

\[ R_{\text{owner}}: 75.50\% \]
Overhead

• Upload traffic
  – Around 300KB each time, compressed to 90KB.

• Latency (average over 210 users)

<table>
<thead>
<tr>
<th>#Samples in training set</th>
<th>Training time (s)</th>
<th>#Samples in test set</th>
<th>Test time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13203</td>
<td>18.415</td>
<td>52065</td>
<td>0.639</td>
</tr>
</tbody>
</table>
Robustness

- Brute-force attack
  - The classifier model for each authorized owner is pre-trained
  - A set of 500K randomly generated samples
  - Percentage of samples detected as non-owner: 94.01%
Robustness

- Human attack
  - A pre-trained classifier for the owner
  - 3 participants handle the phone with various gestures
  - Each participant lunches 10 attacks
  - Each attack lasts for 10 seconds
  - Percentage of samples detected as not owner: 93.84%
ROC Curve

- True positive rate v.s. False positive rate
  - $TPR = \frac{TP}{TP+FN}$, $FPR = \frac{FP}{FP+TN}$
  - Changes the classification threshold (0-1)
Conclusion and Ongoing Work

• DroidCog: The first device level user identification system with wild collected sensor data

• Deploy detection system on the phone

• Improve the classification accuracy
  – Explore more usable but privacy insensitive features (e.g. widely used IP address)

• Combine with existing risk management

• Theme of RSA 2016: Connect to Protect
Issues for existing mobile anti-virus systems
- Easy to evade [DroidChameleon]
- Unable to detect native malware [DroidNative]
- Unable to detect malware in ads or dynamically loaded content [AdShield]

Privacy leakage detection and prevention
- How to find questionable sensitive permissions [AutoCog]
- Real time tracking & preventing privacy leakage on phone
  - Consumer [PrivacyShield]
  - Enterprise Mobility Management (EMM) [AppShield]

Fraud detection mostly with app-level risk management [DroidCog]
- Duplicate detection
- Privacy infringement