Mobile Security, Malwares, and App Stores

CSE 162 – Mobile Computing

Hua Huang
Department of Computer Science and Engineering
University of California, Merced

Mobile Security

Introduction

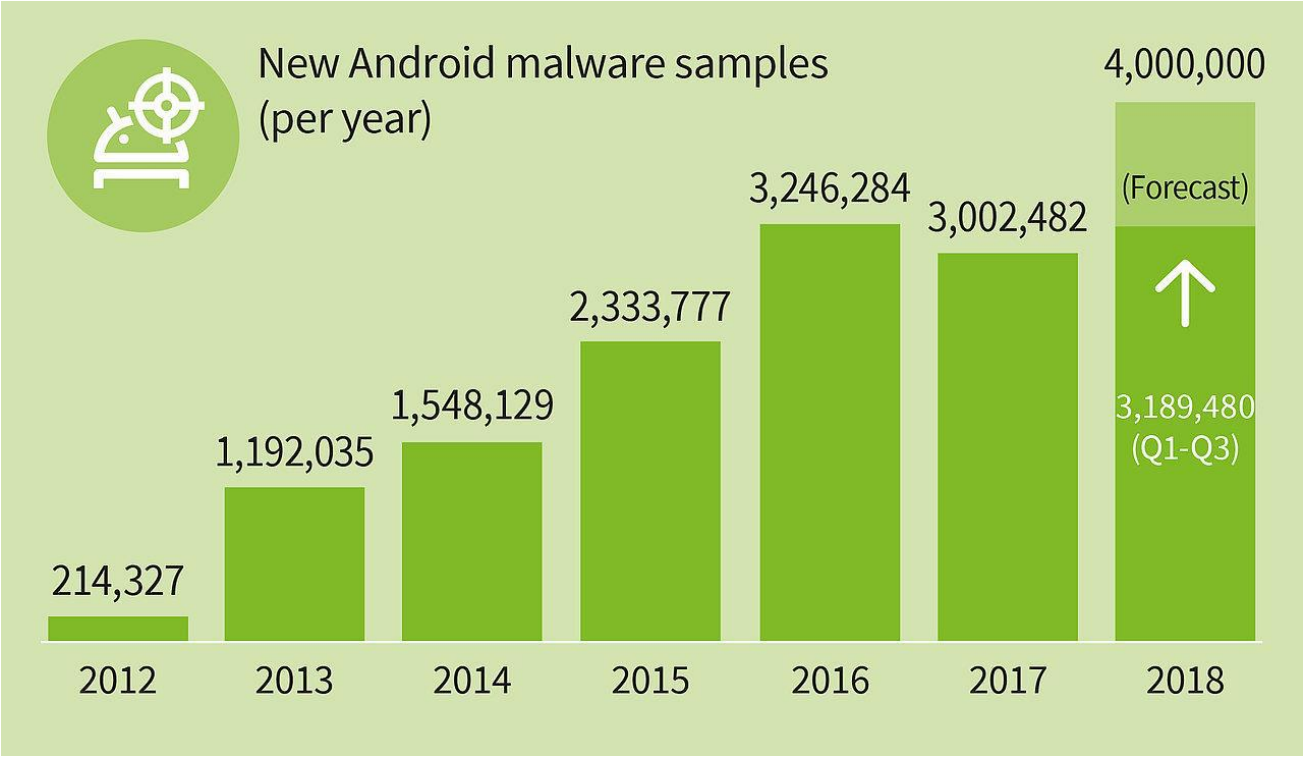
- So many cool mobile apps
- Access to web, personal information, social media, etc
- Security problems (not previously envisaged) have resulted
- Examples:
  - Malicious apps can steal your private information (credit card information, etc)
  - Jogging map generated from paths of Fitbit users can expose locations/behavioral habits of users. E.g. US soldiers at German base
  - Malware can lock your phone till you pay some money (ransomeware)
- Users/developers need better understanding of mobile security

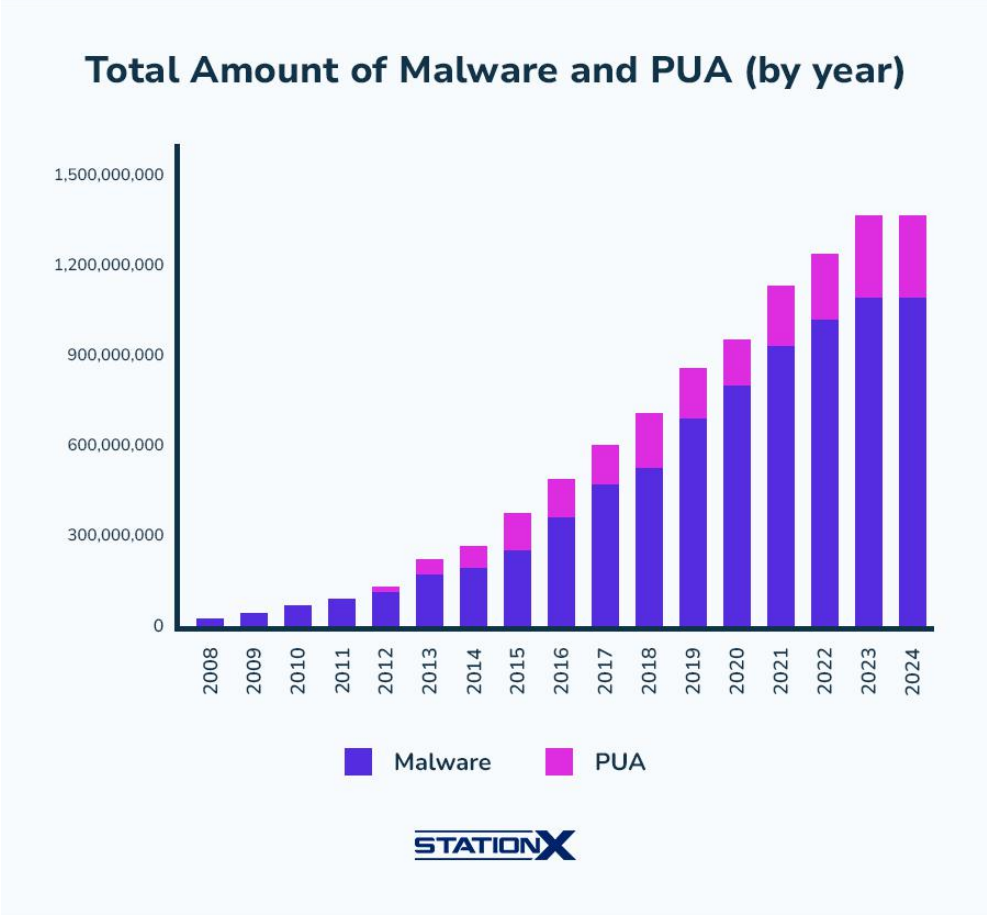
Malware Evolution

Threat Types: Malware, Grayware & Personal Spyware

- **Malware:**
  - Gains access to a mobile device in order to steal data, damage device, or annoying the user, etc. **Malicious!!**
- **Personal Spyware:**
  - Collects user's personal information over of time
  - Sends information to app **installer** instead of author
  - E.g. spouse may install personal spyware to get info
- **Grayware:**
  - Collect data on user, but with no intention to harm user
  - E.g. for marketing, user profiling by a company

Growth of Android Malware





Mobile Malware Survey (*Felt et al*)

Mobile Malware Study?

A survey of mobile malware in the wildAdrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner in Proc SPSM 2021

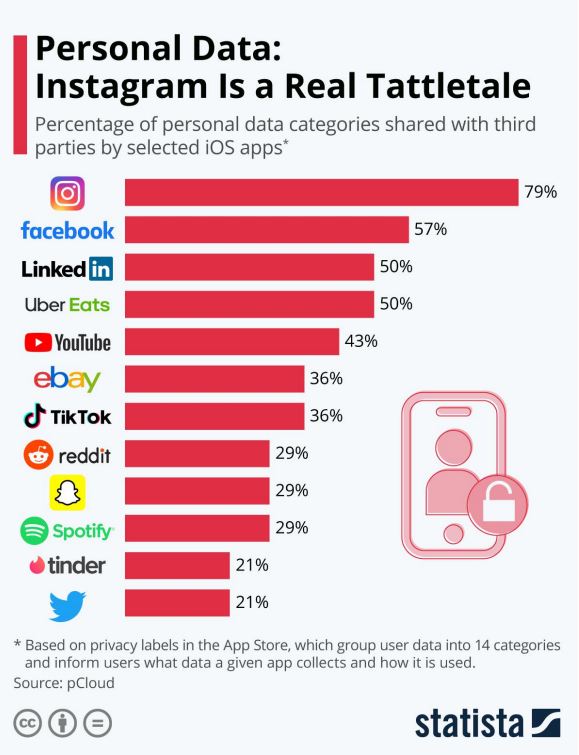
- First major mobile malware study in 2021 by Andrienne Porter Felt *et al*
  - Prior studies mostly focused on PC malware
- Analyzed 46 malwares that spread Jan. 2009 –June 2021
  - 18 –Android
  - 4 –iOS
  - 24 –Symbian (discontinued)
- Analyzed information:
  - in databases maintained by anti-virus companies
    - E.g., Symantec, F-Secure, Fortiguard, Lookout, and Panda Security
  - Discover malware based on mentions of malware in news sources
- Just analyzed malware. Did not analyze spyware and grayware

Categorized Apps based on Behaviors

1.Novelty and amusement

- Designed primarily for minor mischief rather than serious harm
- Typically alters non-critical user settings
- Example: Changing the user's wallpaper or interface theme

- **2. Selling user information**
  - Collects personal data through system/API access
    - Possible data harvested: location, contacts, app download history, browser activity, user preferences
  - Collected information is packaged and sold to third parties, often advertisers
    - Example: A coffee chain (e.g., Dunkin Donuts) purchasing data about users who frequent competitor locations
  - Typical price range: Approximately $1.90–$9.50 per user per month



Categorized Apps based on Behaviors

3. Stealing user credentials

- People use smartphones for activities that require them to input their passwords and payment information. E.g. shopping, banking, e-mail
- Malwares can log keys typed by user (keylogging), scan their documents for username + password
- User credentials can be sold
  - In 2018, black market price of:
    - Bank account credentials: $10 to $1, 000,
    - Credit card numbers: $.10 to $25,
    - E-mail account passwords: $4 to $30

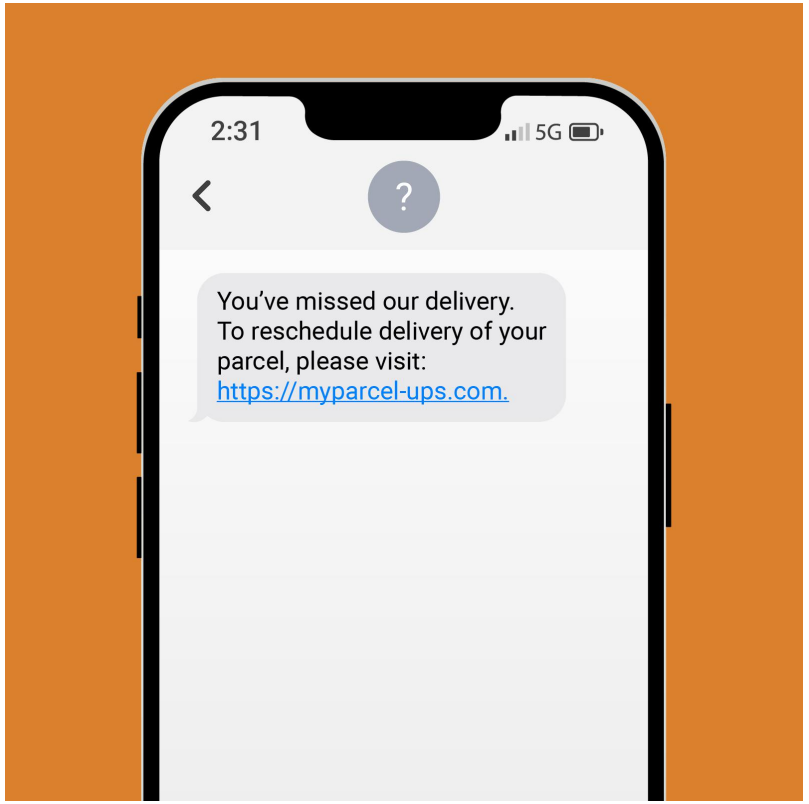
Categorized Apps based on Behaviors

4.Make premium-rate calls and SMS

- Premium rate texts to specific numbers are expensive (E.g. 1-900.. Numbers)
- Attacker can set up premium rate number, Malware sends SMS there
- User is billed by their cell carrier (e.g. sprint), attacker makes money

5.SMS spam

- Used for commercial advertising and phishing
- Sending spam email is illegal in most countries
- Attacker uses malware app on user's phone to send SPAM email
- Harder to track down senders



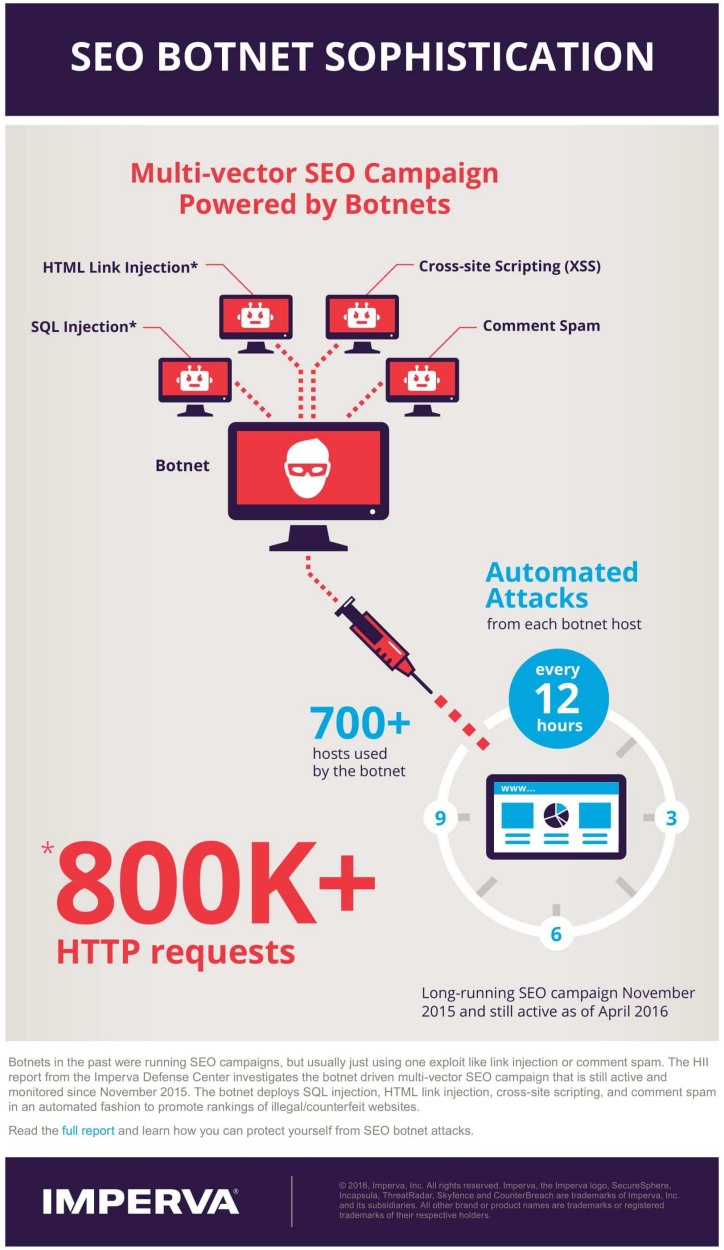
Categorized Apps based on Behaviors

6.Search Engine Optimization (SEO):

- Malware makes HTTP requests for specific pages to increase their search ranking (e.g. on Google)
- Increases popularity of requested websites

7.Ransomeware

- Possess device, e.g. lock screen till money is paid
- *Kenzero*–Japanese virus inserted into pornographic games distributed on P2P networks
  - Publishes user's browser history on public website
  - Asked **5800 Yen**(~$60) to delete information from website
  - About 12 % of users (661 out of 5510) actually paid



Recent High Profile Ransomware Attacks

- Colonial Pipeline (USA, 2021)
  - Ransom paid: $4.4 million
  - Impact: Shutdown of the largest US fuel pipeline, causing fuel shortages across the East Coast.
- JBS Foods (Global, 2021)
  - Ransom paid: $11 million
  - Impact: Forced shutdown of meat-processing plants in the US, Canada, and Australia.
- CNA Financial (USA, 2021)
  - Ransom paid: $40 million (one of the largest known payments)
  - Impact: Shutdown of internal systems for weeks.

Malware Detection based on Permissions

- Does malware request more permissions?
- Analyzed permissions of 11 Android malware

- **Findings: Yes!**
- 8 of 11 malware request SMS permission (73%)
  - Only 4% of non-malicious apps ask for this
- Dangerous permissions: requests for personal info (e.g. contacts), etc
- Malware requests 6.18 dangerous permissions
  - 3.46 for Non-malicious apps

Number of Dangerous permissions	Number of non-malicious applications		Number of malicious applications
0	75	(8%)	-
1	154	(16%)	1
2	182	(19%)	1
3	152	(16%)	-
4	140	(15%)	2
5	82	(9%)	1
6	65	(7%)	-
7	28	(3%)	2
8	19	(2%)	1
9	21	(2%)	1
10	10	(1%)	1
11	6	(0.6%)	1
12	7	(0.7%)	-
13	4	(0.4%)	-
14	4	(0.4%)	-
15	2	(0.2%)	-
16	1	(0.1%)	-
17	1	(0.1%)	-
18	-		-
19	-		-
20	1	(0.1%)	-
21	-		-
22	-		-
23	1	(0.1%)	-
24	-		-
25	-		-
26	1	(0.1%)	-

Table 2: The number of "Dangerous" Android permissions requested by 11 pieces of malware and 956 non-malicious applications [28].

Android Security Model

Android Security

- Android security goals are to
  - Protect user data, system resources (hardware, software)
  - Isolate applications (e.g. app 1 from app 2)

Foundations of Android Security

1.Application Isolation:

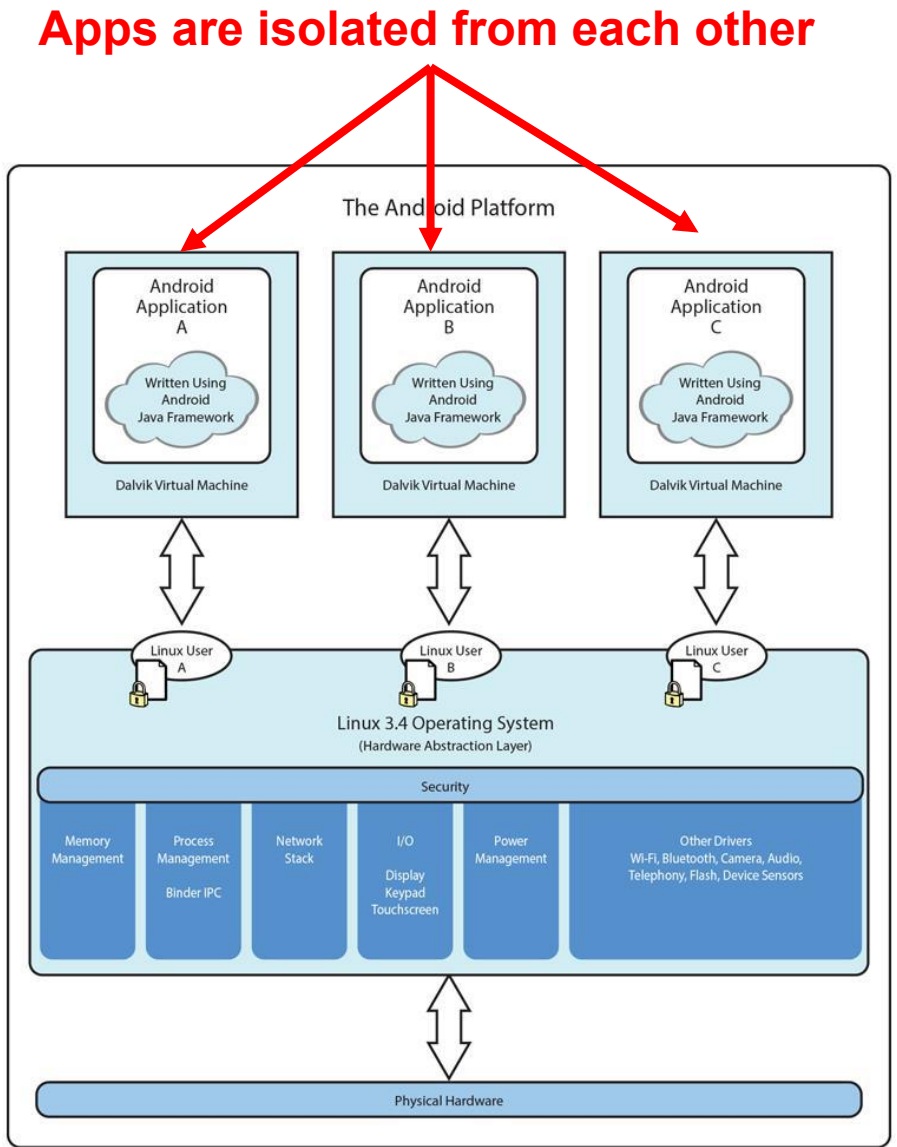
- Application sandboxing: App 1 cannot interact directly with app 2
- Apps can only communicate using secure inter-process communication

2.Permission Requirement:

- Supports default system, and user-defined permissions
- All apps must be signed: identifies author, ensures future updates are authentic

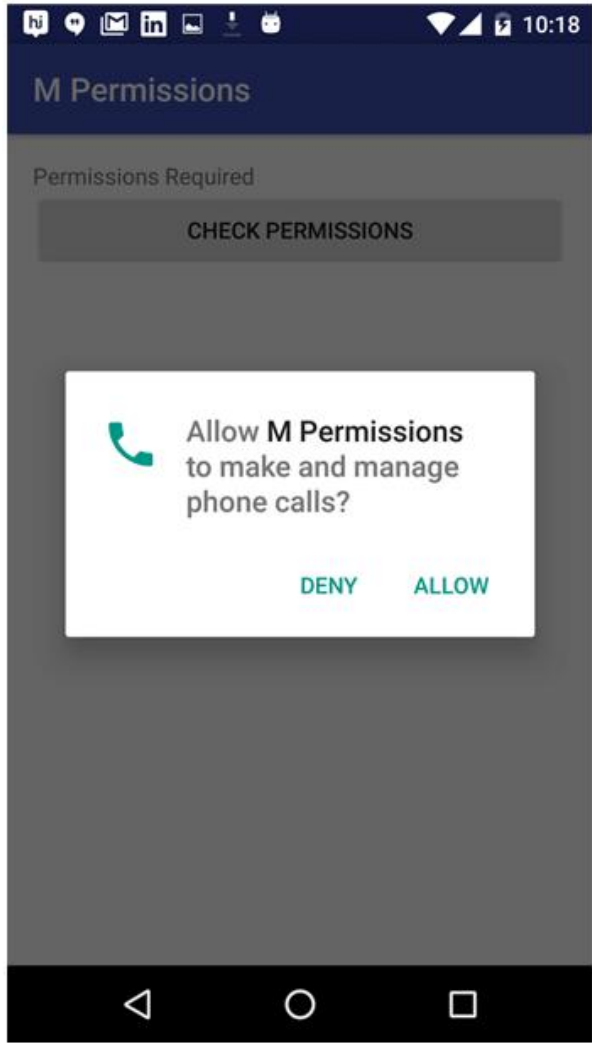
Recall: Android Software Framework

- Each Android app runs in its own security sandbox (VM, minimizes complete system crashes)
- Android OS multi-user Linux system
- Each app is a different user (assigned unique Linux ID)
- Access control: only process with the app's user ID can access its files
- Apps talk to each other only via intents, IPC or ContentProviders



Android Run-Time Permissions Changed in Marshmallow (Android 6.0)

- Pre Android 6.0: Permissions during install
- Android 6.0: Changes:
  - “Normal” permissions don’t require user consent
    - E.g. change timezone
    - Normal permissions can do very little to harm user
    - Automatically granted
  - Dangerous permissions (e.g. access to contacts can harm user
    - Android 6.0: Run-time permissions now required for “dangerous” permissions



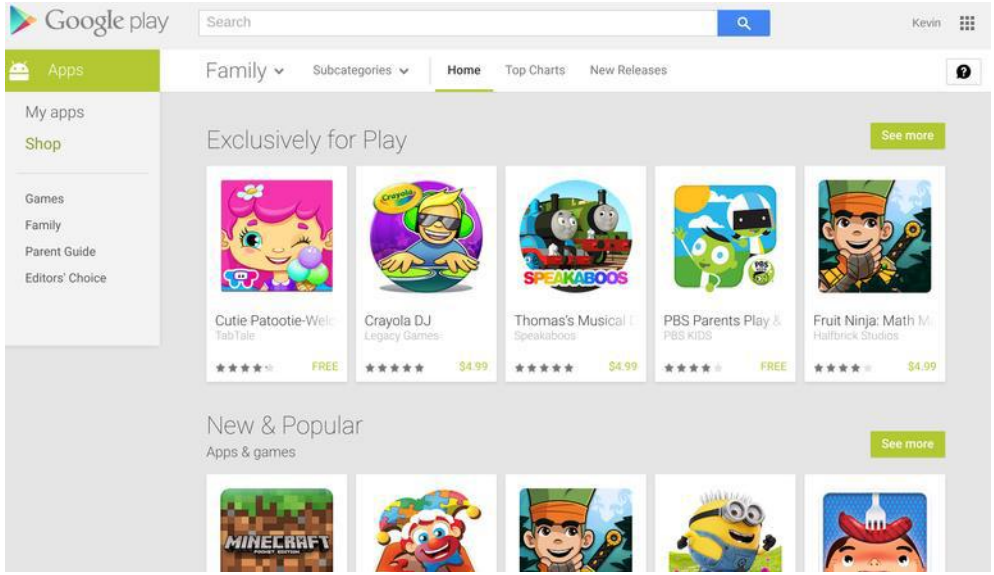
Android Encryption

- Encryption encodes data/information, unauthorized party cannot read it
- **Full-disk encryption:** Android 5.0+ supports full filesystem encryption
  - Single key used to encrypt all the user's data
  - User password needed to access files, even to boot device
- **File-based encryption:** Android 7.0+ allows specific files to be encrypted and unlocked independently
  - Different keys used to encrypt different files

App Markets and Security Scanning

App Markets & Distribution

- Major OS vendors manage their own markets for “certified” apps
  - Android: Google Play Store
  - iOS: App Store (only way to download iPhone apps)



App Market Scanning

- Google App Store: scanning called **Google Play Protect**
  - Antivirus scans apps on Google Play for threats, malware
  - New “peer grouping system:
  - similar apps (e.g. all calculators) are grouped on app market.
  - If an app requests more permissions than similar apps, human takes a look
  - Also scans apps already installed on device, warns user if app looks malicious

App Market Scanning

- Apple App Store
  - Highly regulated
  - All applications are reviewed by human
  - iOS devices can only obtain apps through official app store, unless jailbroken
- Many malware developers target third-party app stores (e.g. Amazon, getJar)
  - Weaker/no restrictions or analysis capabilities

Android Analysis Tools

Analyzing Android Apps

- Attacker can use analysis tools to get more information about an Android app
- **Source code recovery:** generate app source code from executable
- **Static analysis (binaries or source code):** Understand app design without running it.
  - Examine application logic, flow, APIs used
- **Dynamic analysis:** Observe how app executes
  - App memory usage, network usage, response time, performance, etc
- Many available (open source?) tools for all of the above!

Android Analysis Tools

- APKinspector
- Androguard
- AndroBugs
- Qark
- Epicc / IC3
- FlowDroid
- DidFail
- DroidBox
- MobSF

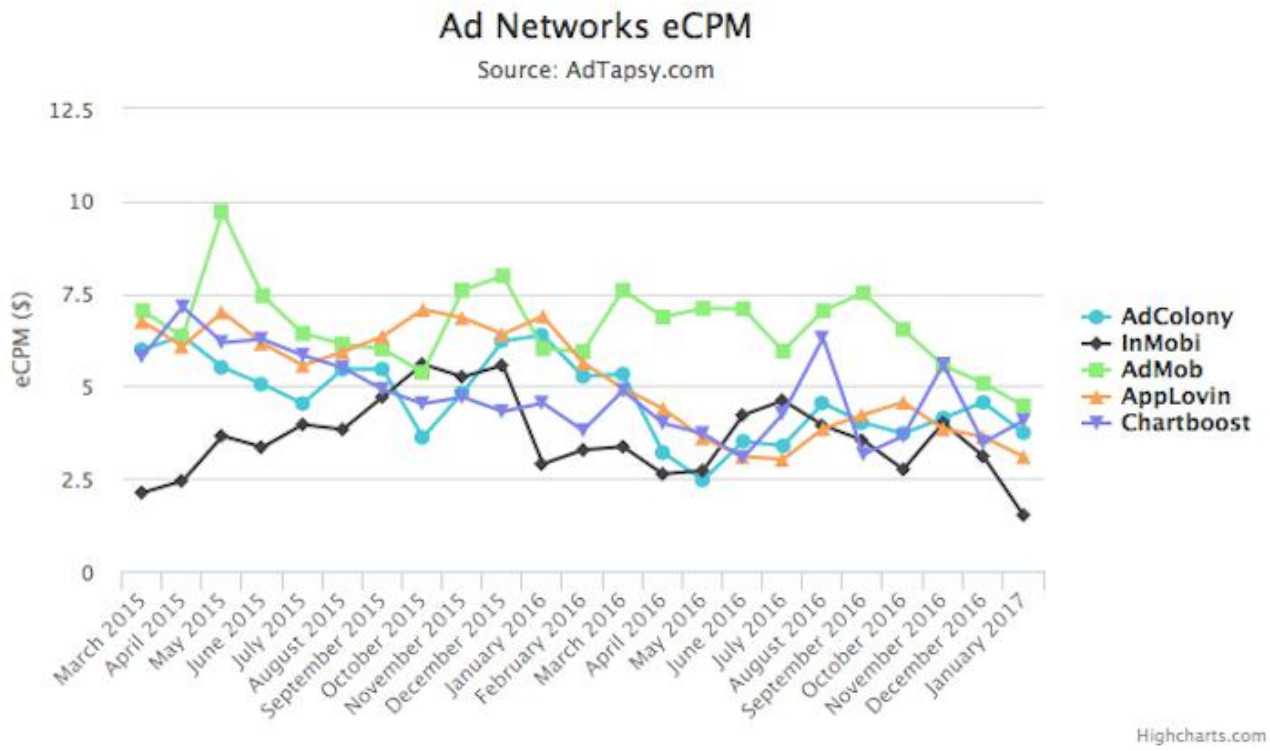
Mobile Ad: Monetization or Grayware?

Ad Services

- App developers make money from apps in 2 main ways:
  - Charge users fee for apps
  - Getting $$$ from advertisers to include ads in apps
- To make money from ads, app author integrates ad services into app
- Mobile ad company serves ads to device

AdMob

- AdMob: Most popular mobile ad company
  - Acquired by Google in 2009



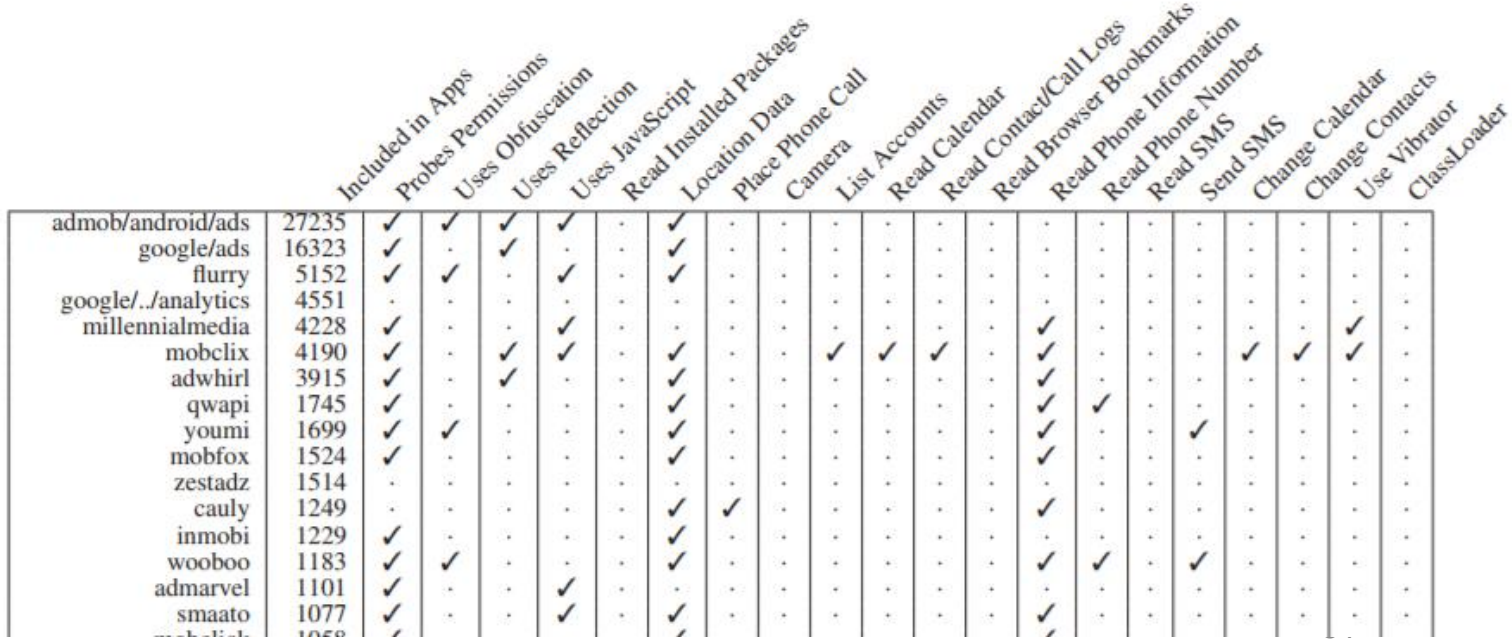
Permissions Requested by Ad Services

- Ad Services can also add requests to app's Android Manifest file
- Total permissions an app's AndroidManifest.xml

  = permissions requested by app **+ permissions requested by ad service**

Rogue? Ad Services

- Google is careful about permissions requested by AdMob
- Some other mobile ad libraries require more permissions:
  - Access location data, camera, account details, calendar, call logs, browser bookmarks, contact lists, phone information, phone number, SMS, etc
  - Make phone calls, send SMS messages, vibrate
  - Change calendar and contacts

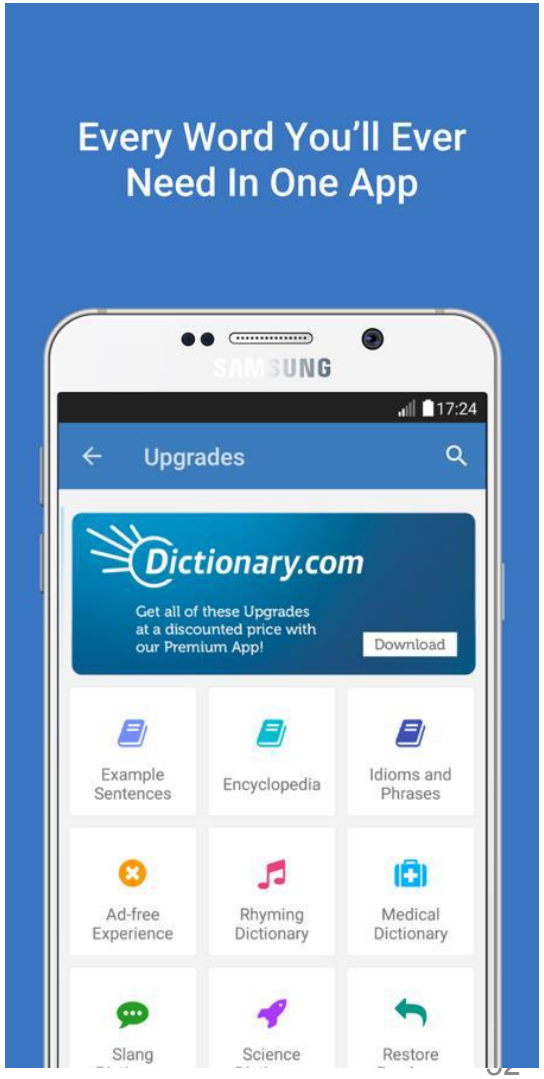


		Included in Apps	Probes Permissions	Uses Obfuscation	Uses Reflection	Uses JavaScript	Read Installed Packages	Location Data	Place Phone Call	Camera	List Accounts	Read Calendar	Read Contact/Call Logs	Read Browser Bookmarks	Read Phone Information	Read Phone Number	Read SMS	Send SMS	Change Calendar	Change Contacts	Use Vibrator	ClassLoader
admob/android/ads	27235	✓	✓	✓	✓	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
google/ads	16323	✓	·	✓	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
flurry	5152	✓	✓	·	✓	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
google/../analytics	4551	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
millennialmedia	4228	✓	·	·	✓	·	·	·	·	·	·	·	✓	·	·	·	·	·	·	✓	·	·
mobclix	4190	✓	·	✓	✓	·	✓	·	·	✓	✓	✓	✓	·	·	·	✓	✓	✓	·	·	·
adwhirl	3915	✓	·	✓	·	·	✓	·	·	·	·	·	✓	·	·	·	·	·	·	·	·	·
qwapi	1745	✓	·	·	·	·	✓	·	·	·	·	·	✓	✓	·	·	·	·	·	·	·	·
youmi	1699	✓	✓	·	·	·	✓	·	·	·	·	·	✓	·	·	✓	·	·	·	·	·	·
mobfox	1524	✓	·	·	·	·	✓	·	·	·	·	·	✓	·	·	·	·	·	·	·	·	·
zestadz	1514	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
cauly	1249	·	·	·	·	·	✓	✓	·	·	·	·	✓	·	·	·	·	·	·	·	·	·
inmobi	1229	✓	·	·	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
wooboo	1183	✓	✓	·	·	·	✓	·	·	·	·	·	✓	✓	·	✓	·	·	·	·	·	·
admarvel	1101	✓	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
smaato	1077	✓	·	·	✓	·	✓	·	·	·	·	·	✓	·	·	·	·	·	·	·	·	·

Final Words: Mobile Ad Services

- Many apps use multiple ad services
  - Angry Birds app (a game) includes 7+ ad services

- Example of rogue requests:
  - One version of the Dictionary.com app requests permissions to **monitor phone calls** and **access location**



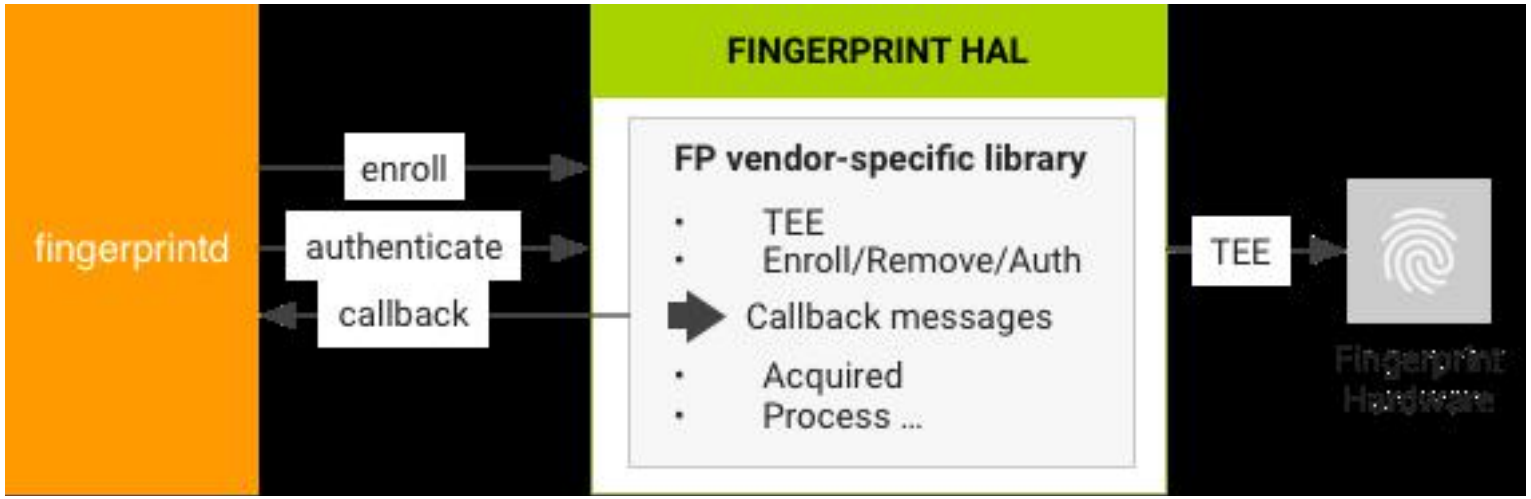
Authentication using Biometrics

Biometrics

- Passwords tough to remember, manage
- Many users have simple passwords (e.g. 1234) or do not change passwords
- Biometrics are unique physiological attributes of each person
  - Fingerprint, voice, face
- Can be used to replace passwords
  - No need to remember anything. Just be you. Cool!!

Android Biometric Authentication: Fingerprints

- **Fingerprint:** On devices with fingerprint sensor, users can enroll multiple fingerprints for unlocking device



Face ID

- Apple's Face ID is a facial-recognition technology that launched in 2017.
- The technology replaces Apple's Touch ID fingerprint scanning system
- Face ID uses a "TrueDepth camera system“
  - It consists of sensors, cameras, and a dot projector at the top of the iPhone display
  - It creates a detailed 3D map of your face

Continuous Passive Authentication using Behavioral Biometrics

User Behavior as a Biometric

- User behaviors patterns are unique personal features. E.g
  - Each person's daily location pattern (home, work, places) + times
  - Walk pattern
  - Phone tilt pattern

- **General idea:** Continuously authenticate user as long as they behave like themselves
- If we can measure user behavior reliably, this could enable **passive authentication**

BehavioMetrics

- Derived from Behavioral Biometrics
  - Behavioral: the way a human subject behaves
  - Biometrics: technologies and methods that measure and analyzes biological characteristics of the human body
    - Fingerprints, eye retina, voice patterns

- BehavioMetrics:
  - Measurable behavior to recognize or verify a human’s identity

Mobile Sensing → BehavioMetrics

- Accelerometer
  - Activity & movement pattern, hand trembling, driving style
  - sleeping pattern
  - Activity level, steps per day, calories burned
- Motion sensors, WiFi, Bluetooth
  - Indoor position and trajectory.
- GPS
  - outdoor location, geo-trace, commuting pattern
- Microphone, camera
  - From background noise: activity, type of location.
  - From voice: stress level, emotion
  - Video/audio: additional contexts
- Keyboard, taps, swipes
  - User interactions, tasks .....

BehavioMetrics → Security

- Track smartphone user behavior using sensors
- Continuously extract and classify features from sensors = Detect contexts, personal behavior features (pattern classification)
- Generate unique pattern for each user
- **Trust score:** How similar is today's behavior to user's typical behavior
- Trigger authentication schemes with different levels of authentication based on trust score

Using Hand Gestures to Curb Mobile Malware (*Shrestha et al*)

Malware Protection using Hand Movements

Curbing Mobile Malware Based on User-Transparent Hand MovementsBabins Shrestha,Manar Mohamed, Anders Borg, Nitesh Saxena and Sandeep Tamrakar in Proc IEEE Percom 2015

- **General idea:** Use real world hand movements to distinguish malware from real user
- Real user will make certain natural hand gestures when:
  - Making phone call
  - Taking a picture
  - Swiping to use NFC reader

- These hand gestures will be missing if activity is by malware
- **Main idea:** Check for these gestures (gesture recognition) to distinguish malware requests from valid user requests

Sensors used for Gesture Identification

- Gesture Identifier used sensors to detect natural hand movements associated with phone dialing, taking picture, NFC usage
  - **Motion Sensors:** Accelerometer and gyroscope
  - **Position Sensors:** Magnetometer and orientation sensors
  - **Environmental Sensors:** Temperature, pressure and illuminance

TABLE I. SENSORS UTILIZED FOR GESTURE DETECTION

Type	Sensor	Description
Motion	Accelerometer (A)	The acceleration force including gravity
Motion	Gyroscope (Gy)	The rate of rotation
Motion	Linear Acceleration (LA)	The acceleration force excluding gravity
Motion	Rotation Vector (R)	The orientation of a device
Motion	Gravity (G)	The gravity force on the device
Position	Game Rotation (GR)	Uncalibrated rotation vector
Position	Magnetic Field (M)	The ambient magnetic field
Position	Orientation (O)	The device orientation
Environment	Pressure (P)	The ambient air pressure

System Architecture

- 3 Entities
  - **Gesture Identifier:** classifier to identify gesture
  - **Permission Controller:** checks permissions granted by Android
  - **Gesture Manager:** compares gestures with permissions

- **Results:** > 85% accuracy (user gesture detection)