User Expectations in Mobile App Security

Tao Xie

Joint Work w/ Wesley Brooks, Wing Lam, Davis Li, David Yang, Carl Gunter, ChengXiang Zhai (Illinois)
Benjamin Andow, William Enck (NCSU)

UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

illinois.edu

NSA SoS Lablet, NSF Medium CNS-1513939,
Google Faculty Research Award

Collaborating SoS Lablet PIs:
Sean Smith (Dartmouth), Ross Koppel (U Penn),
Jim Blythe (USC)

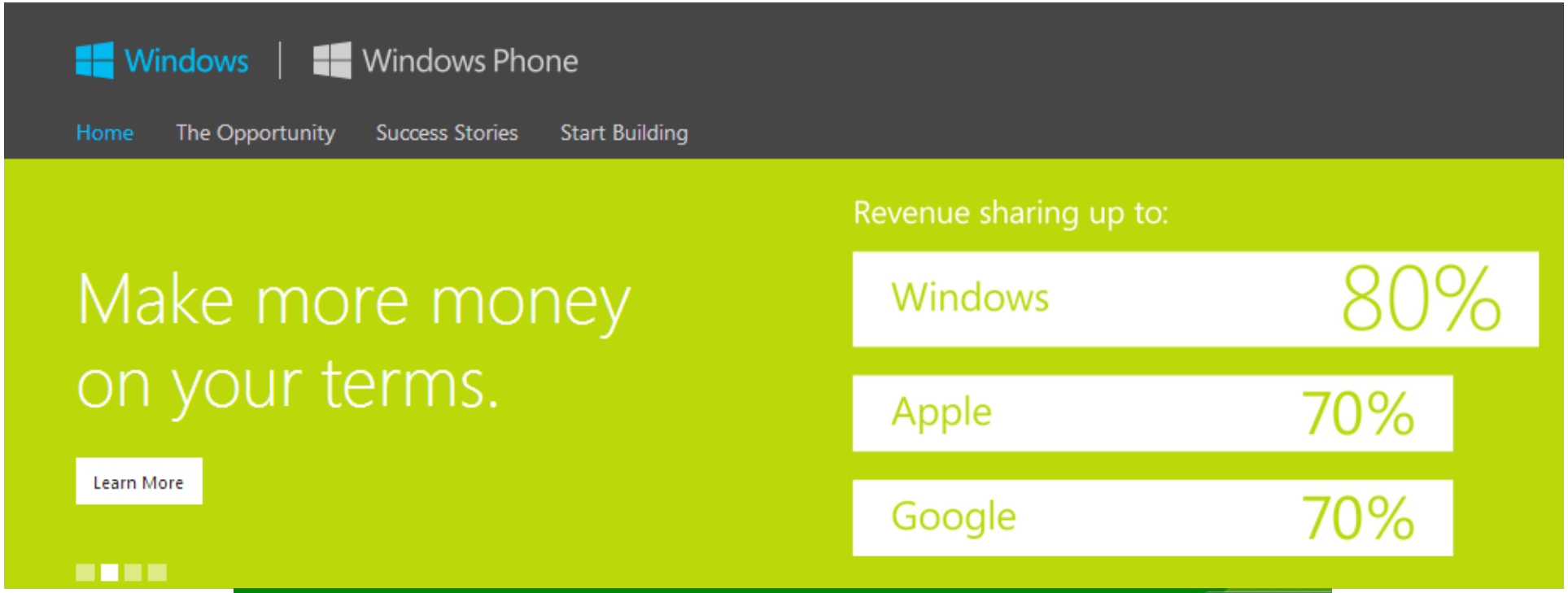
Mobile App Markets

Apple App Store

Google Play

Microsoft Windows Phone

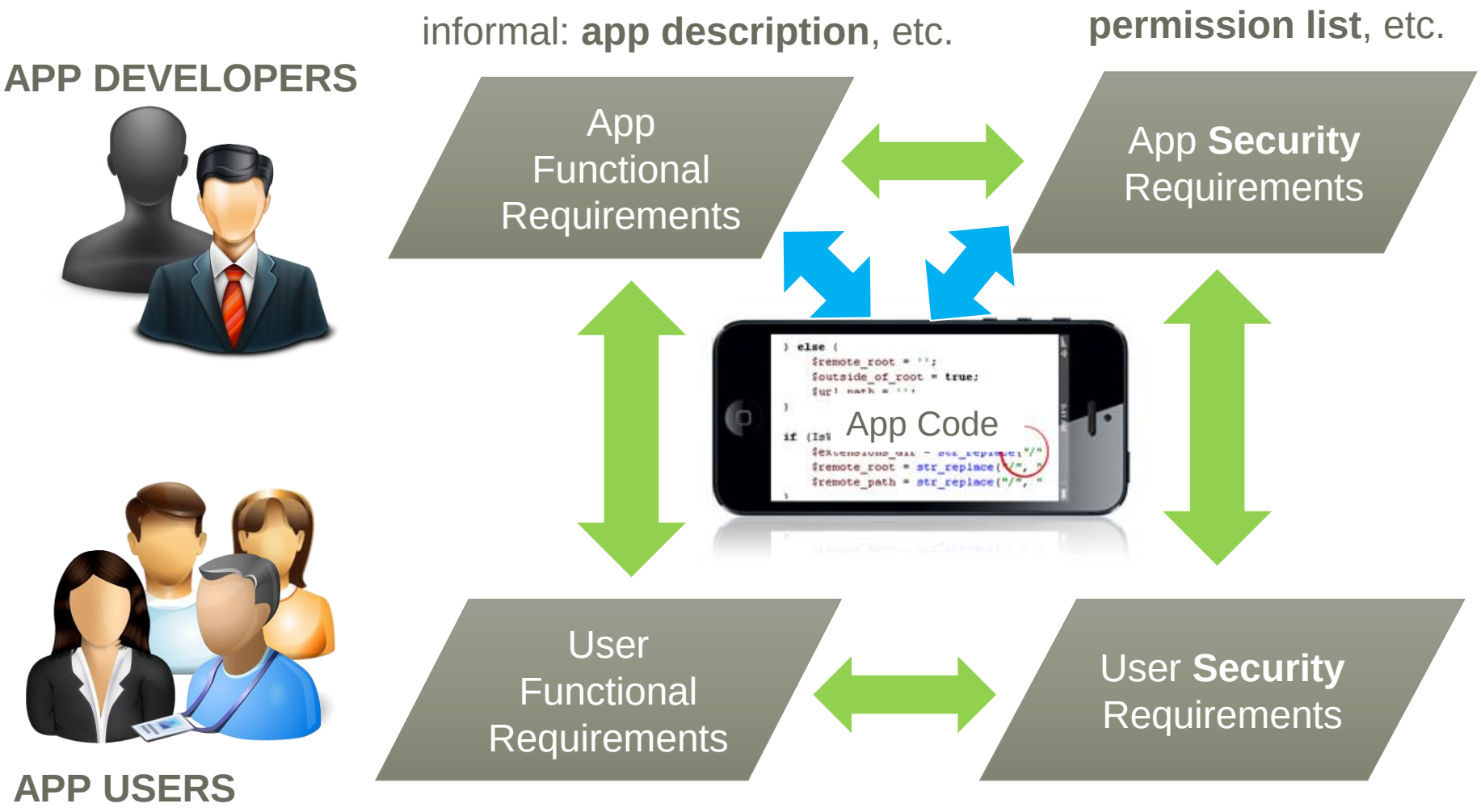
App Store beyond Mobile Apps!



Mobile apps can access a wealth of sensitive data and sensors

Acknowledgment: Slide adapted from Haoyu Wang's

“Conceptual” Model



Informal App Functional Requirements: App Description

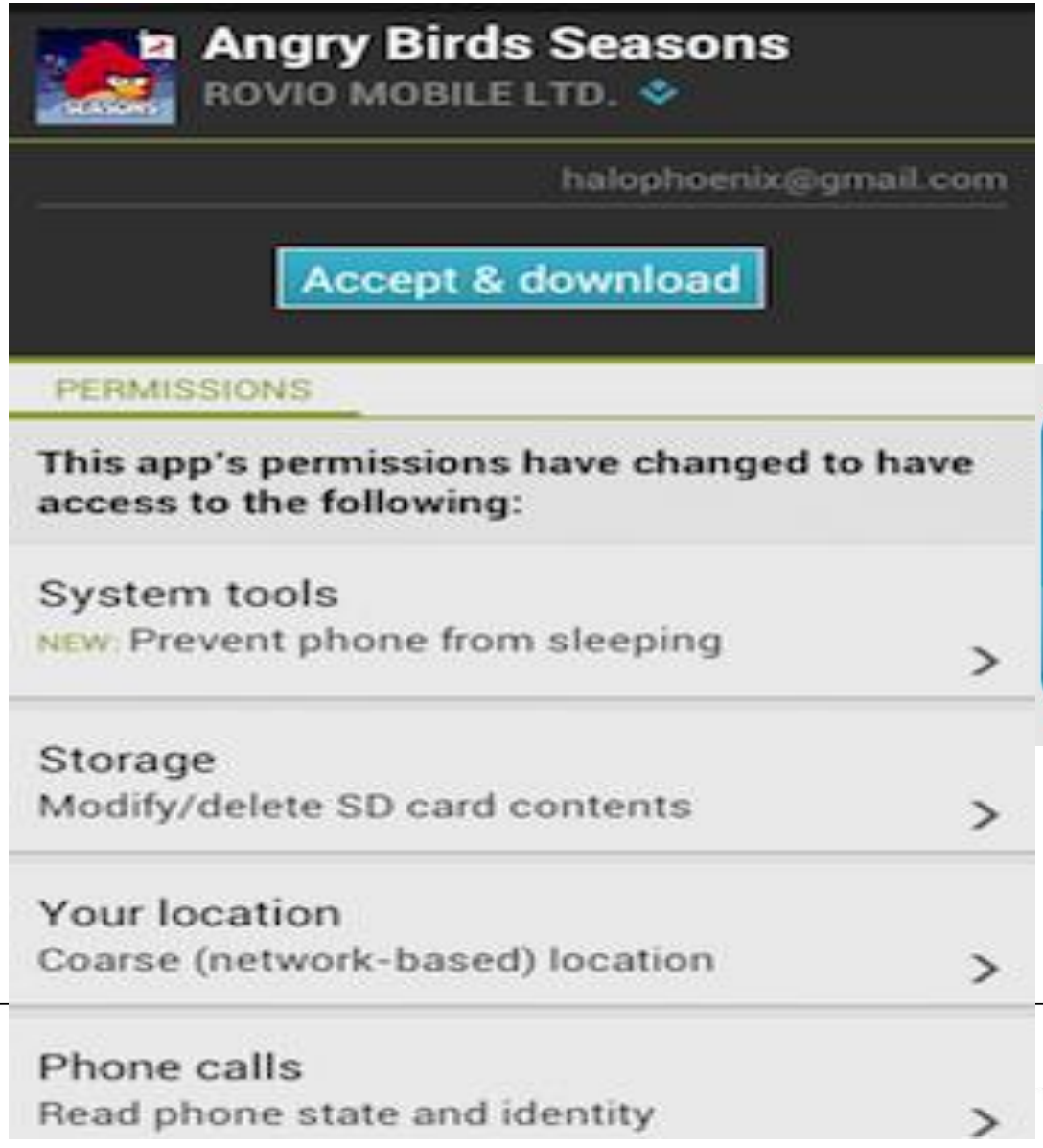
Description

The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs. Use the unique powers of each bird to destroy the pigs' defenses. Angry Birds features challenging physics-based gameplay and hours of replay value. Each level requires logic, skill, and force to solve.

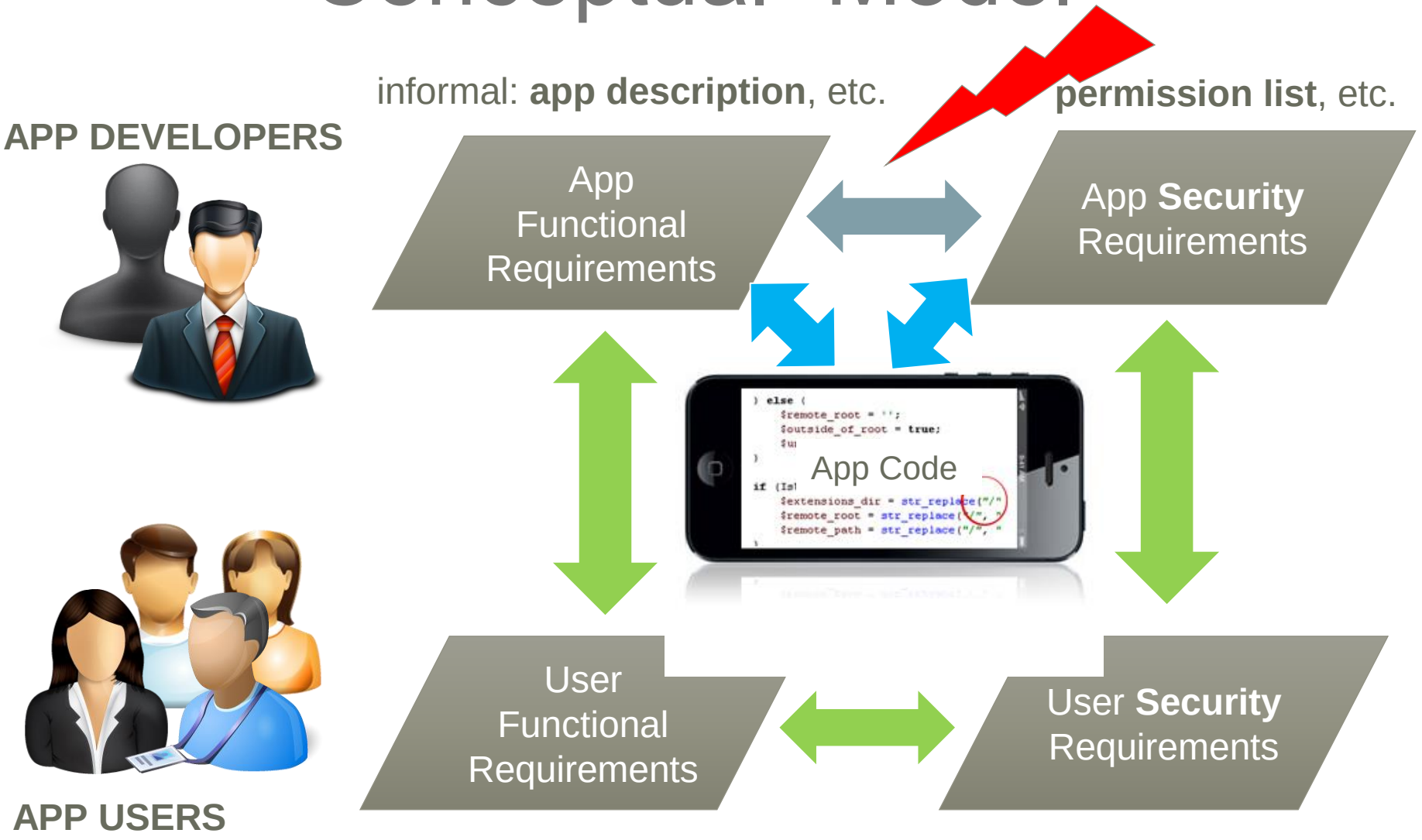
If you get stuck in the game, you can purchase the Mighty Eagle! Mighty Eagle is a one-time in-app purchase Angry Birds that gives unlimited use. This phenomenal creature will soar from the skies to wreak havoc and smash the pesky pigs into oblivion. There's just one catch: you can only use the aid of Mighty Eagle to pass a once per hour. Mighty Eagle also includes all new gameplay goals and achievements!

In addition to the Mighty Eagle, Angry Birds now has power-ups! Boost your birds' abilities and three-star leve

App Security Requirements: Permission List



“Conceptual” Model



Example Andriod App: Angry Birds

Strassia
Guest

Why does Angry Birds need to know who I call?

11-19-2012, 06:15 PM

DrDeth
Charter Member

Marketing.

Rovio explains the new permission that Angry Birds Seasons requires

Tweet ShareThis StumbleUpon Pinterest Reddit

Published on Friday, 02 December 2011 10:42

11-19-2012, 05:46 PM

Baron Greenback
Guest

Join

Angry Birds just needs to know the phone state so that it can handle an incoming call when you are playing.

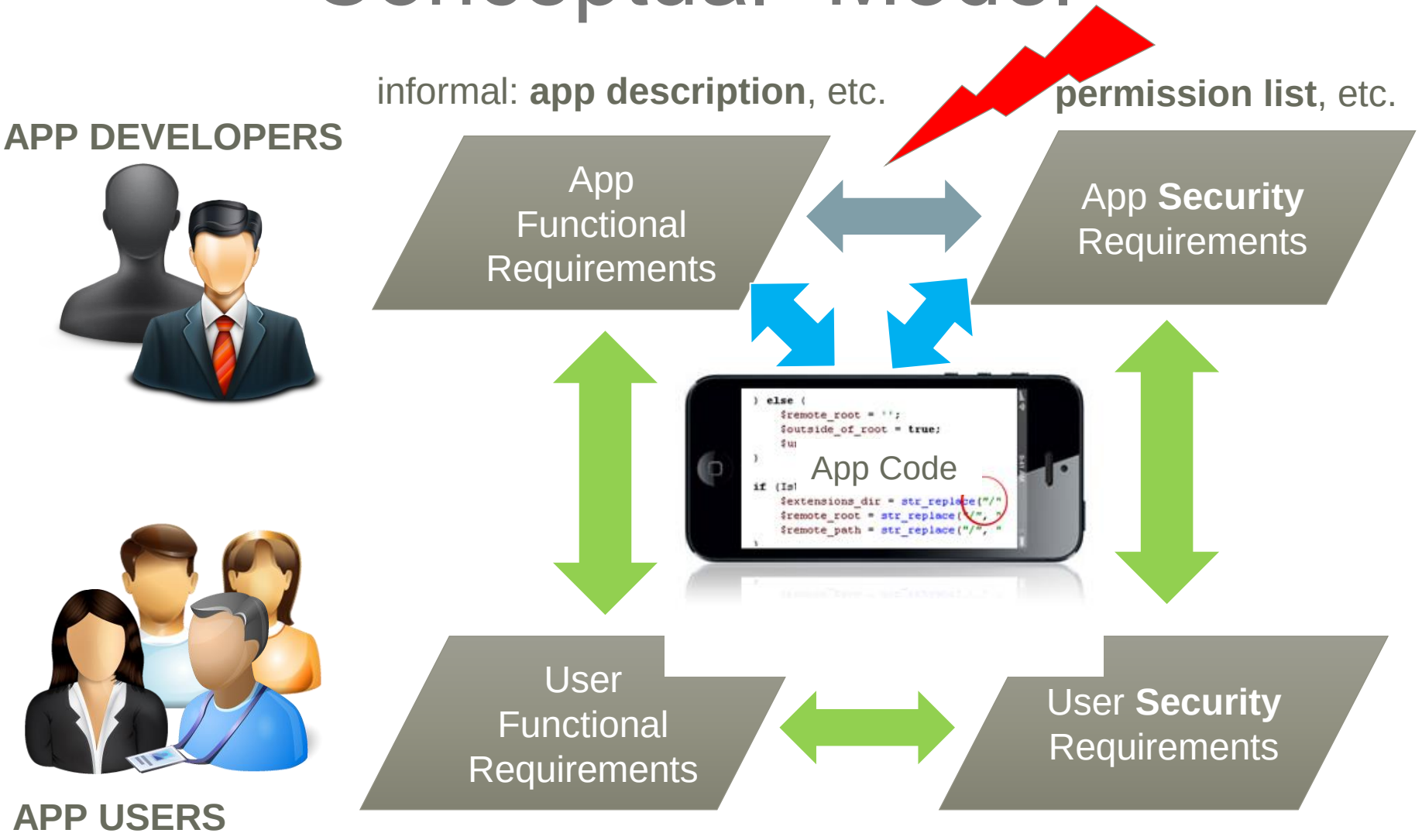
It is NOT that People Don’t Care

BUSINESS INSIDER | Tech | Finance | Politics | Strategy | Life | Sports | Video | All

People were asked to read aloud the terms and conditions for popular apps and were shocked by what they actually agreed to

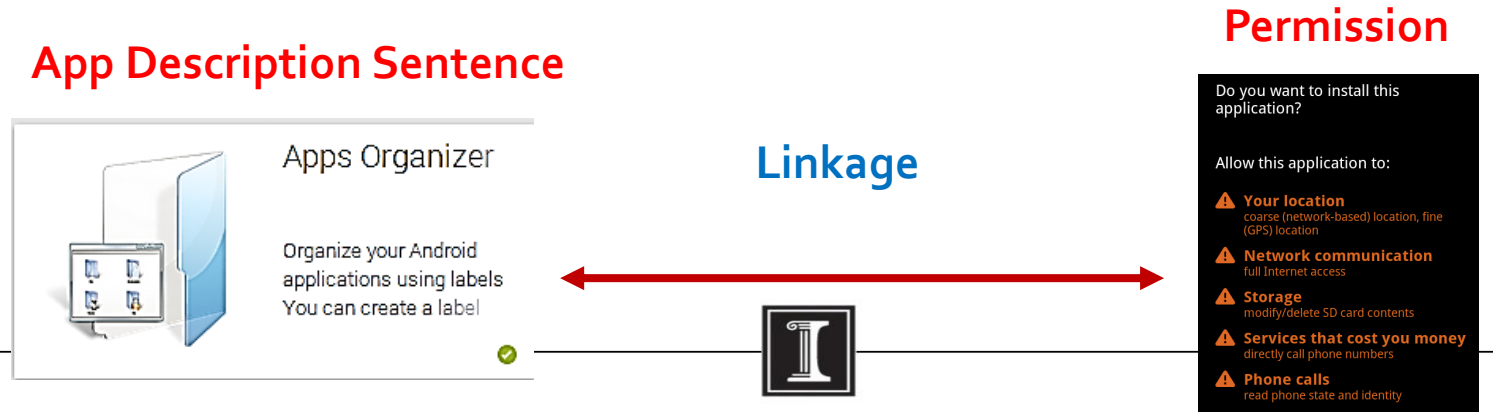
http://www.businessinsider.com/app-permission-agreements-privacy-video-2015-2

“Conceptual” Model



WHYPER: Text Analytics for Mobile Security

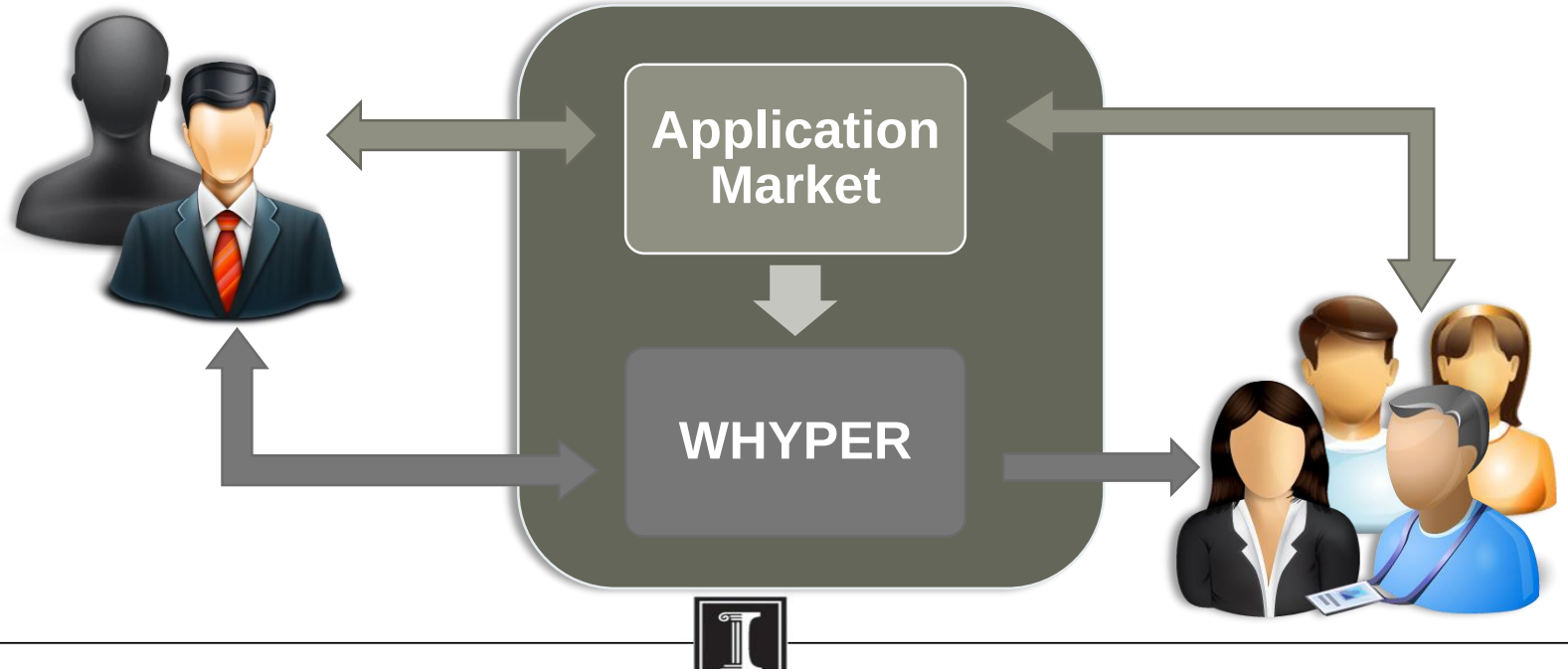
- Focus on permission ⟷ app descriptions
  - permissions (protecting user understandable resources) should be discussed
- ***What does the users expect (w.r.t. app functionalities)?***
  - **GPS Tracker:** record and send location
  - **Phone-Call Recorder:** record audio during phone call



Pandita et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications. *USENIX Security 2013*
http://web.engr.illinois.edu/~taoxie/publications/usenixsec13-whyper.pdf

WHYPER Overview

- Enhance **user** experience while installing apps
- Enforce functionality disclosure on **developers**
- Complement program analysis to ensure justifications



Pandita et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications. *USENIX Security 2013*
http://web.engr.illinois.edu/~taoxie/publications/usenixsec13-whyper.pdf

Natural Language Processing on App Description

"*Also you can share the yoga exercise to your friends via Email and SMS.*"

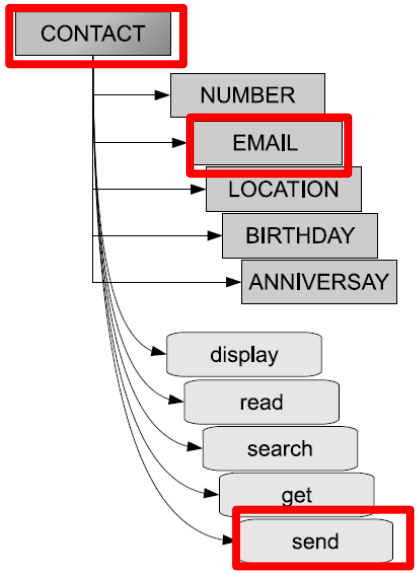
- Implication of using the contact permission
- Permission sentences

Confounding effects:

- Certain keywords such as "**contact**" have a confounding meaning
- E.g., "… displays user contacts, …" vs "… contact me at abc@xyz.com".

Semantic inference:

- Sentences describe a sensitive action w/o referring to keywords
- E.g., "share yoga exercises with your friends via Email and SMS"



*NLP + **Semantic Graphs/Ontologies** Derived from Android API Documents*

Challenges

- **Synonym analysis**
  - *Ex non-permission sentence: "You can now turn recordings into ringtones."*
    - *functionality that allows users to create ringtones from previously recorded sounds but NOT requiring permission to record audio*
    - *false positive due to using synonym: (turn, start)*

- **Limitations of Semantic Graphs**
  - *Ex. permission sentence: "blow into the mic to extinguish the flame like a real candle"*
    - *false negative due to failing to associate "blow into" with "record"*
  - Automatic mining from user comments and forums

Not All Malware Developers Are “Dumb” or “Lazy”

Security Threat Report 2014

Android Malware: Mutating and Getting Smarter

Android malware continues to grow and evolve, following paths first blazed by Windows. But there is progress to report in securing the platform.

Since we first detected Android malware in August 2010, we have recorded well over 300 malware families. And we have seen the Android malware ecosystem follow in many of the paths first established years ago by Windows malware.

Sophisticated at avoiding detection and removal

Recently, we have seen great innovation in how Android malware seeks to avoid and counter detection methods. Ginmaster is a case in point. First discovered in China in

In 2012, Ginmaster began resisting detection by obfuscating class names, encrypting URLs and C&C instructions, and moving towards the polymorphism techniques that have become commonplace in Windows malware. In 2013, Ginmaster’s developers implemented far more complex and subtle obfuscation and encryption, making this malware harder to detect or reverse engineer.[14] Meanwhile, with each quarter since early 2012, we have seen a steady growth in detections of Ginmaster, reaching more than 4,700 samples

Example Malicious App

I'm being charged for unwanted premium rate text messages

Are you paying for texts you don't want or didn't ask for?

It pays to read the small print before you sign up to a text service so you know exactly what it will cost.

The regulator for premium rate phone services - PhonepayPlus - handled almost 16,000 complaints in the year to 2014.

Of these, 80% related to SMS messages, with over 8,000 leading to enforcement action.

http://www.which.co.uk/consumer-rights/problem/im-being-charged-for-unwanted-premium-rate-text-messages

Example Malicious App

Unwanted premium rate texts

Also known as 'reverse billed' messages, premium rate texts come from four, five or six-digit numbers and are normally for subscription services such as games or weather updates.

The texts generally cost about £1.50 each for which you might not realise you're being charged, and can mean you end up with a shockingly high phone bill.

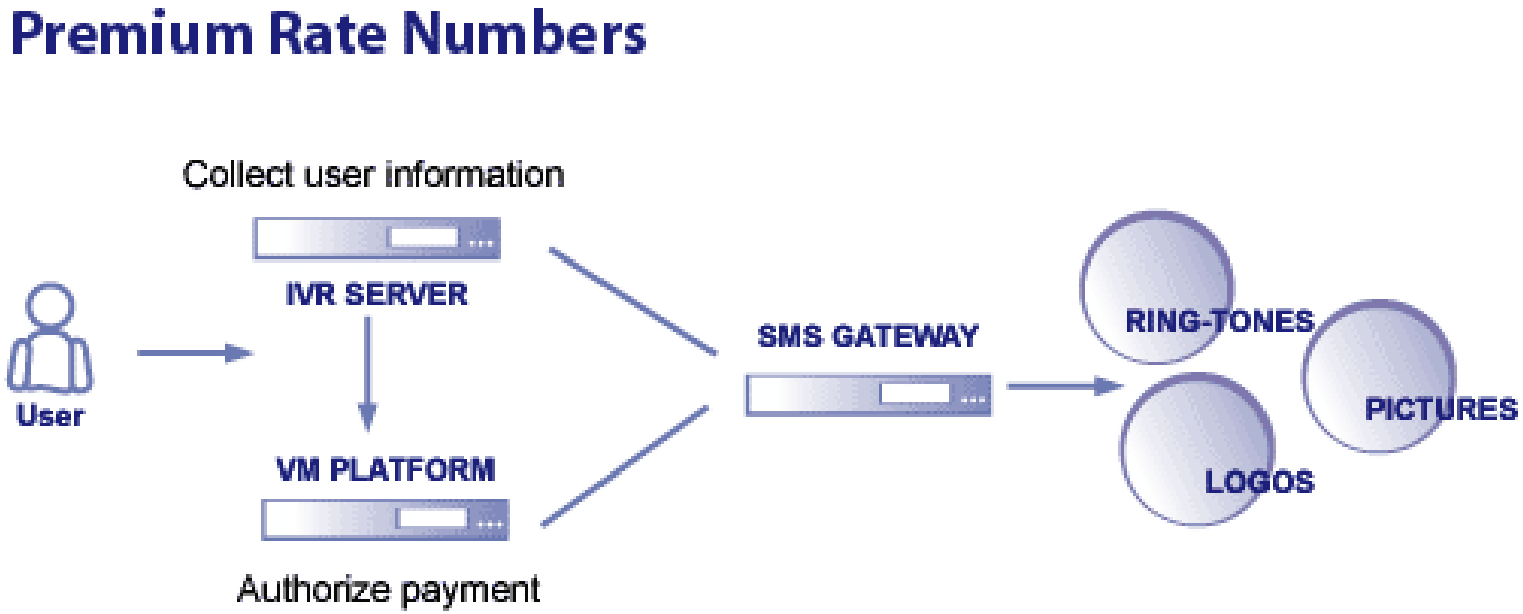
Texts of this kind can only be sent out if you sign up to the service.

There are several scenarios where you should check the small print before you sign up to a text service, so you know exactly what it will cost.

You can use our step by step guide to stop unwanted premium rate text messages.

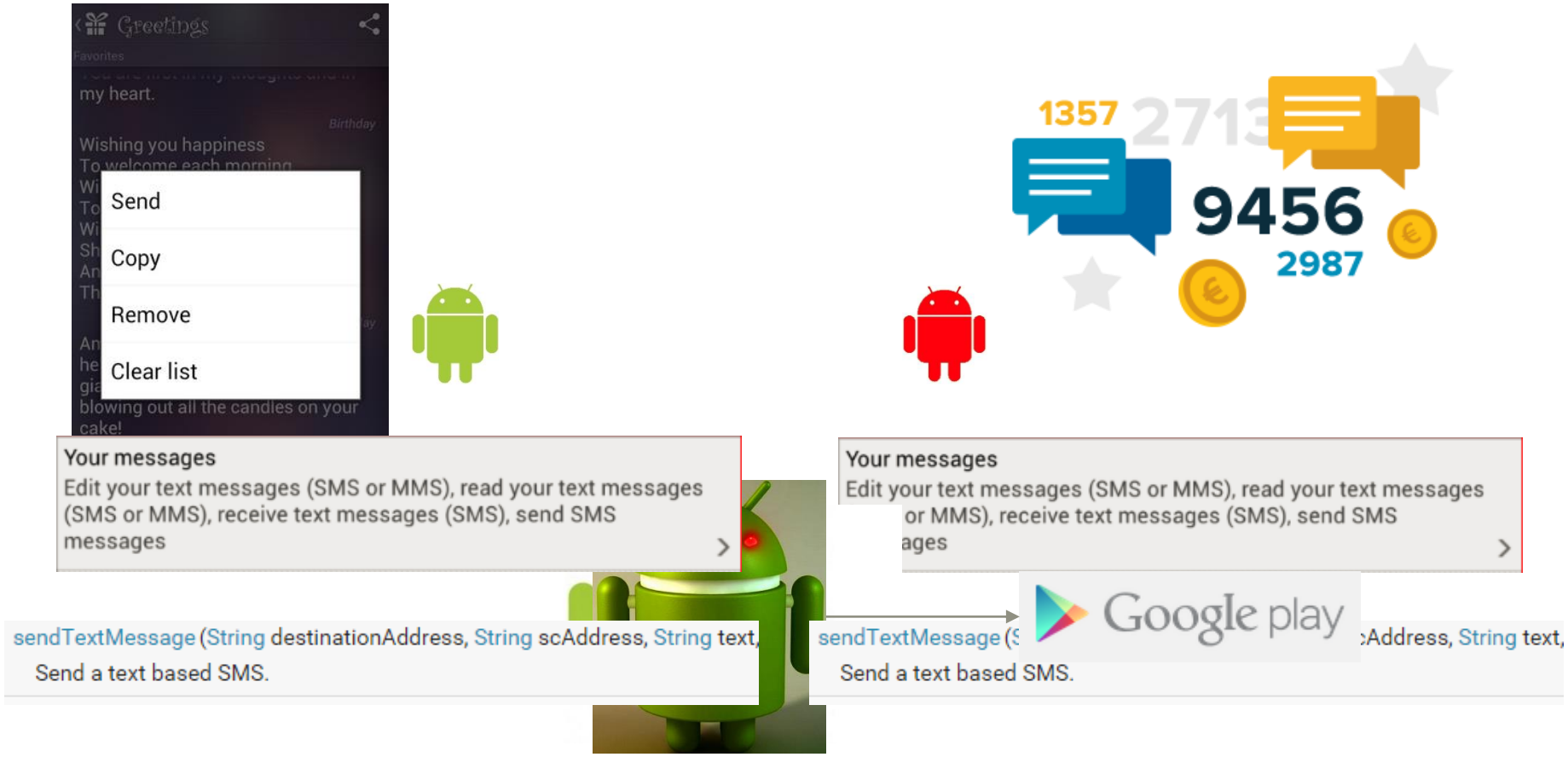
http://www.which.co.uk/consumer-rights/problem/im-being-charged-for-unwanted-premium-rate-text-messages

Example Malicious App



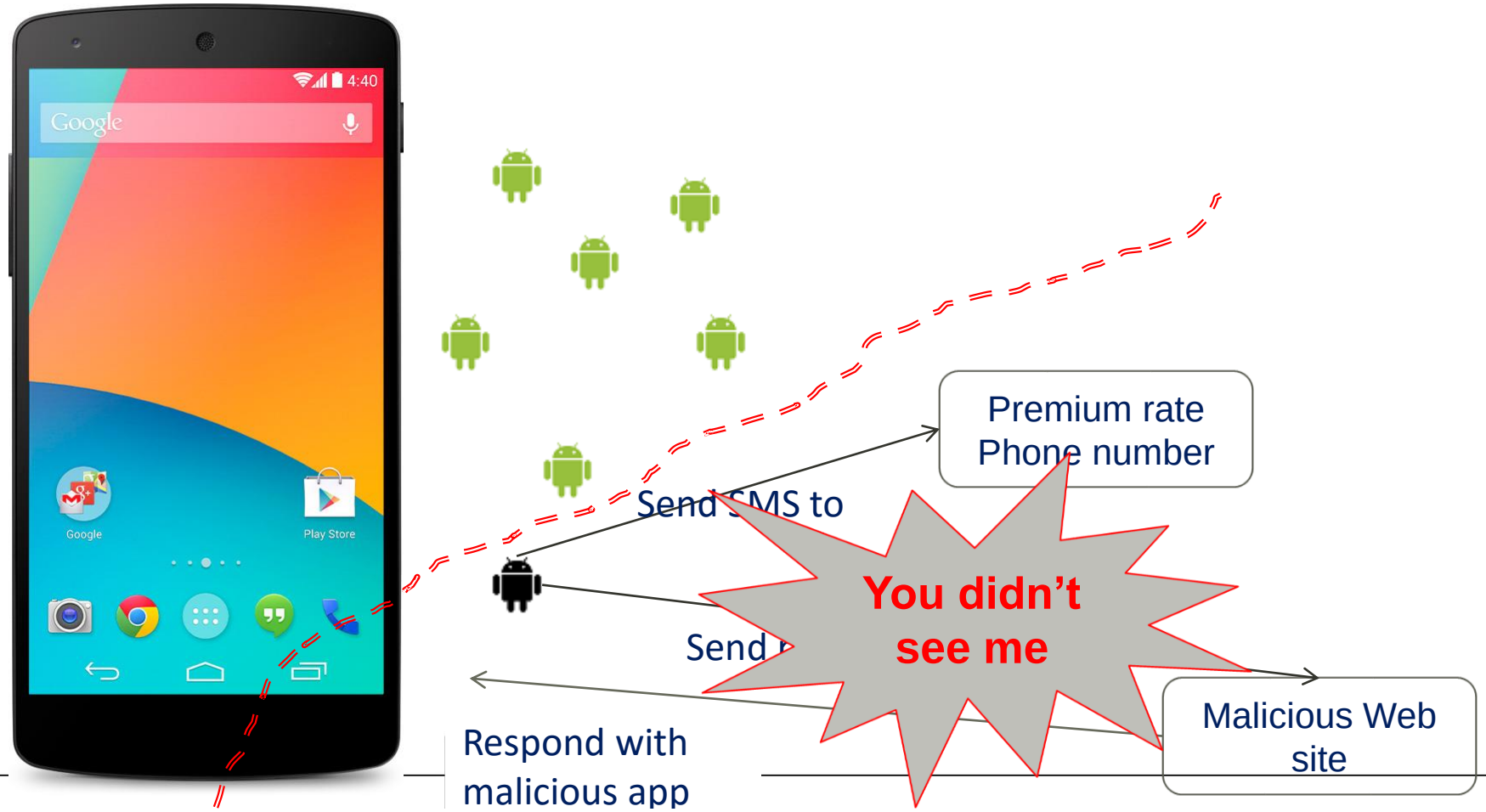
Not All Malware Developers Are “Dumb” or “Lazy”

Benign? Malicious?



Insight by Other Researchers

- Stealthy behaviors in Android apps



Huang et al. AsDroid: Detecting Stealthy Behaviors in Android Applications by User Interface and Program Behavior Contradiction. ICSE 2014.
https://www.cs.purdue.edu/homes/xyzhang/Comp/icse14_2.pdf

Acknowledgment: slide adapted from AsDroid authors'

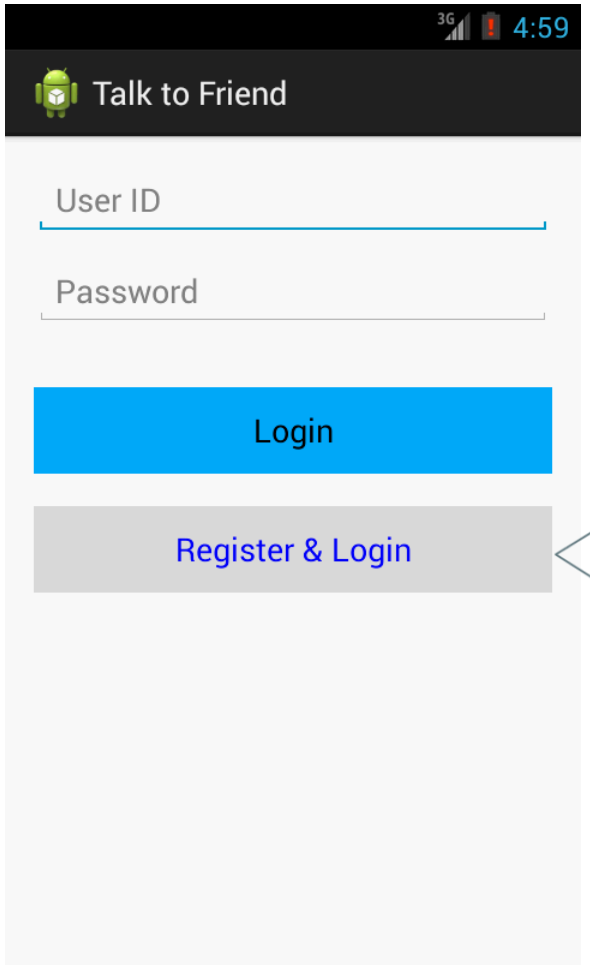
Motivation: Stealthy App Behaviors

- 52-64% of existing malwares send stealthy premium rate SMS messages or make phone calls [Felt et al. SPSM'11, Zhou et al. S&P'12]

- Stealthy HTTP requests are also very common undesirable behaviors in malware [Felt et al. SPSM'11]
  - A kind of malware making stealthy HTTP connections caused 8 million dollars loss in March 2010 in China [news in SINA.com]

Acknowledgment: slide adapted from AsDroid authors'

Motivating Example

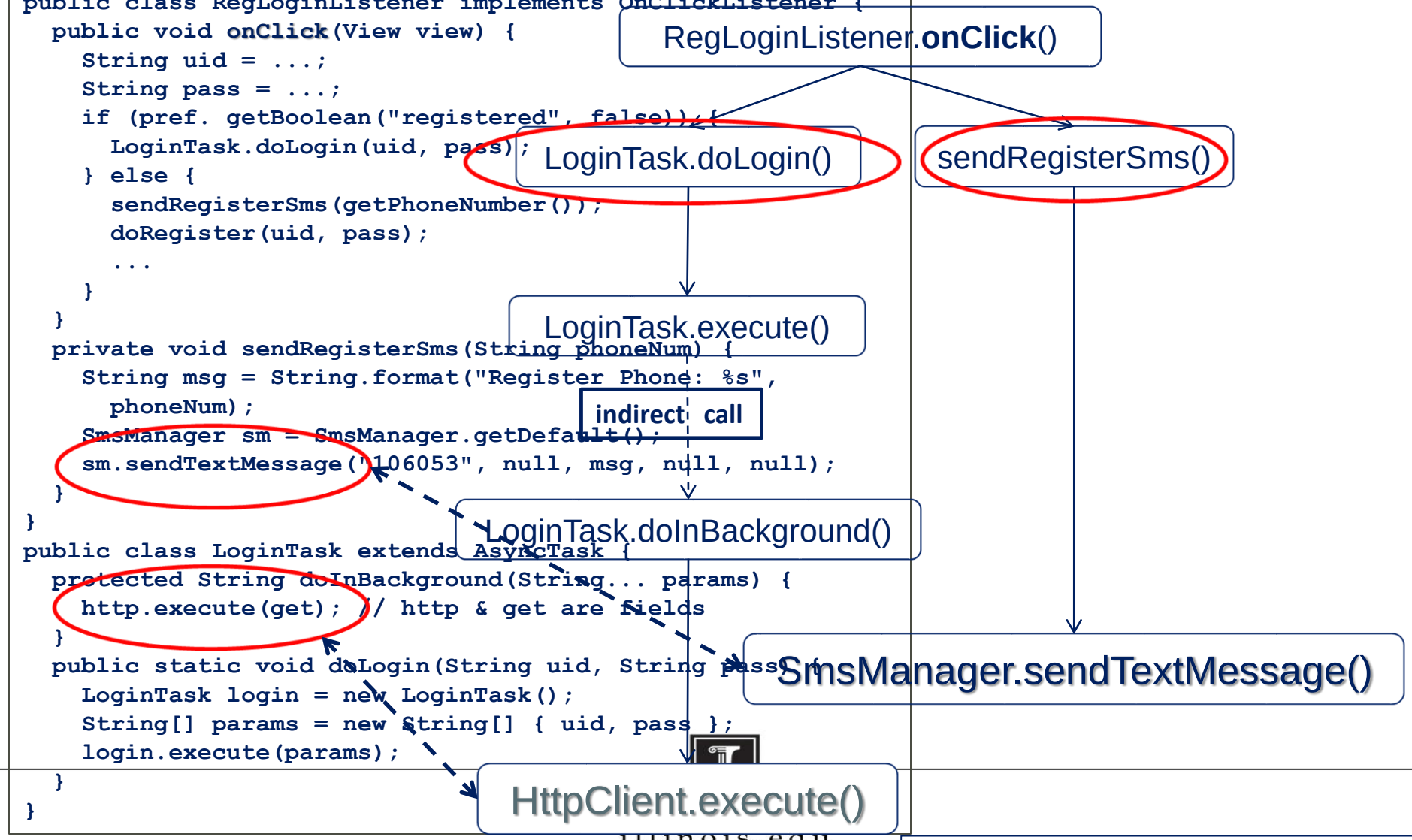



```
public class RegLoginListener implements OnClickListener {
  public void onClick(View view) {
    String uid = ...;
    String pass = ...;
    if (pref. getBoolean("registered", false)) {
      LoginTask.doLogin(uid, pass);
    } else {
      sendRegisterSms(getPhoneNumber());
      doRegister(uid, pass);
      ...
    }
  }
}
```


Acknowledgment: slide adapted from AsDroid authors'

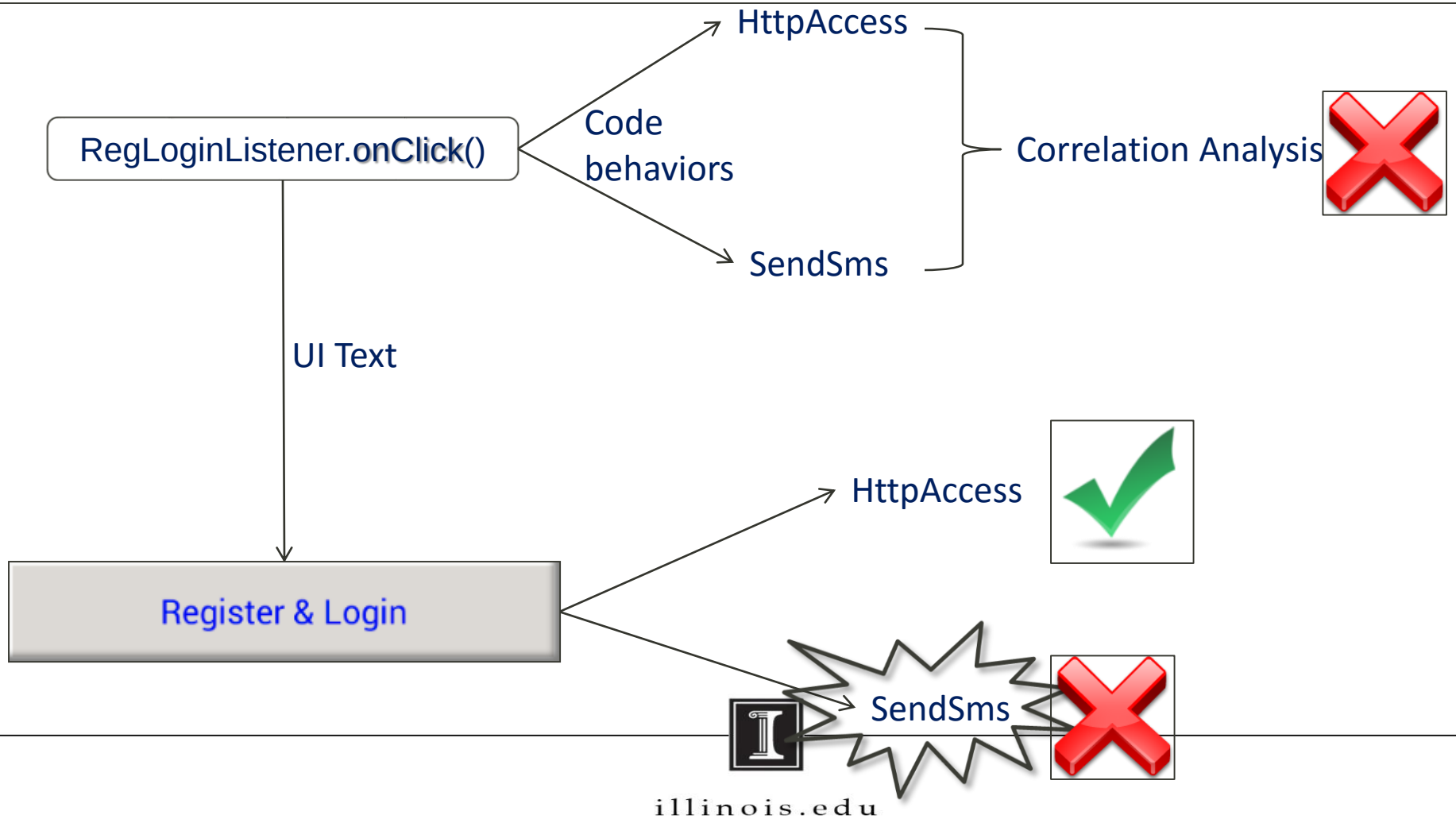
Motivating Example

```
public class RegLoginListener implements OnClickListener {
  public void onClick(View view) {
    String uid = ...;
    String pass = ...;
    if (pref. getBoolean("registered", false)){
      LoginTask.doLogin(uid, pass);
    } else {
      sendRegisterSms(getPhoneNumber());
      doRegister(uid, pass);
      ...
    }
  }
  private void sendRegisterSms(String phoneNum) {
    String msg = String.format("Register Phone: %s",
      phoneNum);
    SmsManager sm = SmsManager.getDefault();
    sm.sendTextMessage("106053", null, msg, null, null);
  }
}
public class LoginTask extends AsyncTask {
  protected String doInBackground(String... params) {
    http.execute(get); // http & get are fields
  }
  public static void doLogin(String uid, String pass) {
    LoginTask login = new LoginTask();
    String[] params = new String[] { uid, pass };
    login.execute(params);
  }
}
```



Acknowledgment: slide adapted from AsDroid authors'

AsDroid Approach



Acknowledgment: slide adapted from AsDroid authors'

Our Own Insight

Different goals of benign apps vs. malware.

- Benign apps
  - Meet requirements from users (as delivering utility)
- Malware
  - Trigger malicious behaviors frequently (as maximizing profits)
  - Evade detection (as prolonging lifetime)

Differentiating characteristics

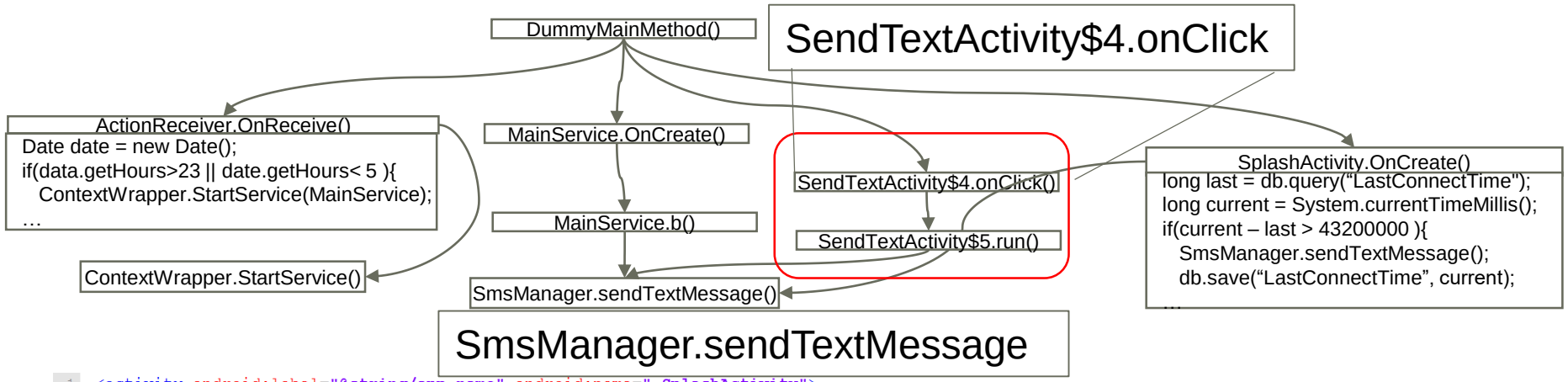
Mobile malware (vs. benign apps)

- **Frequently enough** to meet the need: **frequent** occurrences of **imperceptible** system events;
  - E.g., many malware families trigger malicious behaviors via background events.

Balance!!!

- **Not too frequently** for users to notice anomaly: **indicative** states of external environments
  - E.g., Send premium SMS every 12 hours

Example of malicious app

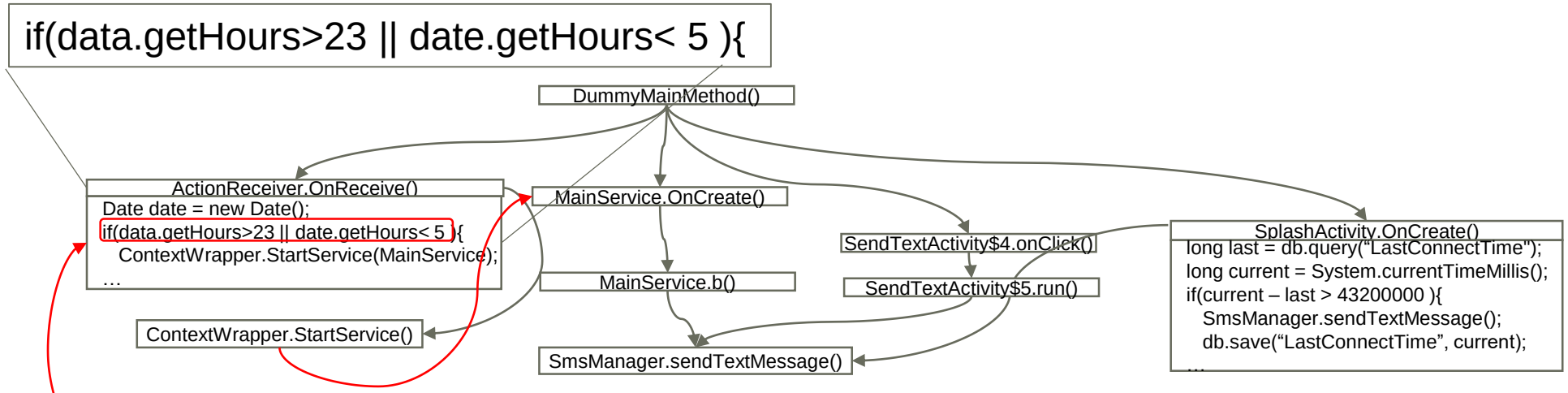



```
<activity android:label="@string/app_name" android:name=".SplashActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<service android:name="com.android.main.MainService" android:process=":main" />
<receiver android:name="com.android.main.ActionReceiver">
    <intent-filter>
        <action android:name="android.intent.action.SIG_STR" />
    </intent-filter>
</receiver>
```

The app will send an SMS when

- user clicks a button in the app

Example of malicious app



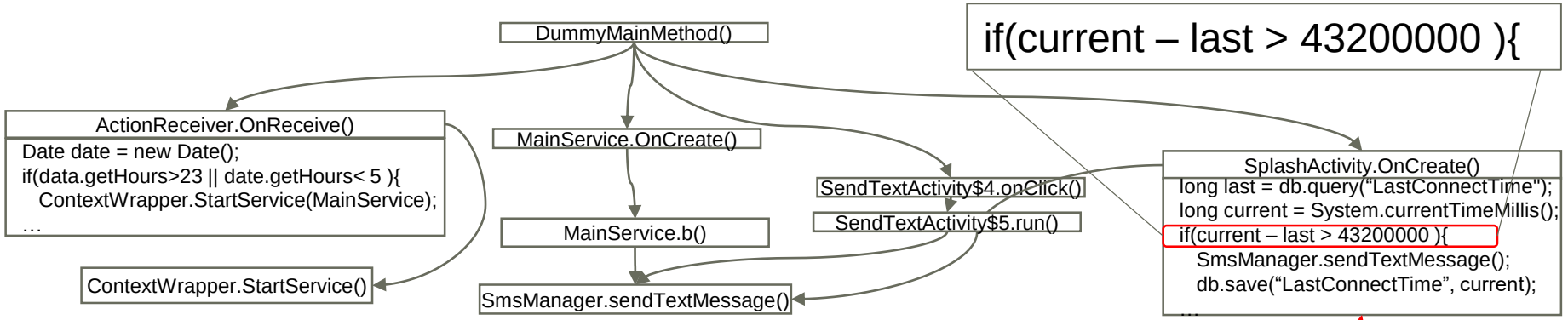

```
1  <activity android:label="@string/app_name" android:name=".SplashActivity">
2      <intent-filter>
3          <action android:name="android.intent.action.MAIN" />
4          <category android:name="android.intent.category.LAUNCHER" />
5      </intent-filter>
6  </activity>
7  <service android:name="com.android.main.MainService" android:process=":main" />
8  <receiver android:name="com.android.main.ActionReceiver">
9      <intent-filter>
10         <action android:name="android.intent.action.SIG_STR" />
11     </intent-filter>
12 </receiver>
```

Android.intent.action.SIG_STR

The app will send an SMS when

- phone signal strength changes (frequent)
- current time is within 11PM-5 AM (not too frequent, User not around)

Example




```
<activity android:label="@string/app_name" android:name=".SplashActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<service android:name="com.android.main.MainService" android:process=":main" />
<receiver android:name="com.android.main.ActionReceiver">
    <intent-filter>
        <action android:name="android.intent.action.SIG_STR" />
    </intent-filter>
</receiver>
```

The app will send an SMS when

- user enters the app (frequent)
- (current time – time when last msg sent) >12 hours (not too frequent)

AppContext

- Capture differentiating characteristics with contexts of security-sensitive behavior.

- Leverage contexts in machine learning (classification) to differentiate malware and benign apps.

Yang et al. AppContext: Differentiating Malicious and Benign Mobile App Behavior Under Contexts. ICSE 2015.
http://taoxie.cs.illinois.edu/publications/icse15-appcontext.pdf

Different Insight by Other Researchers

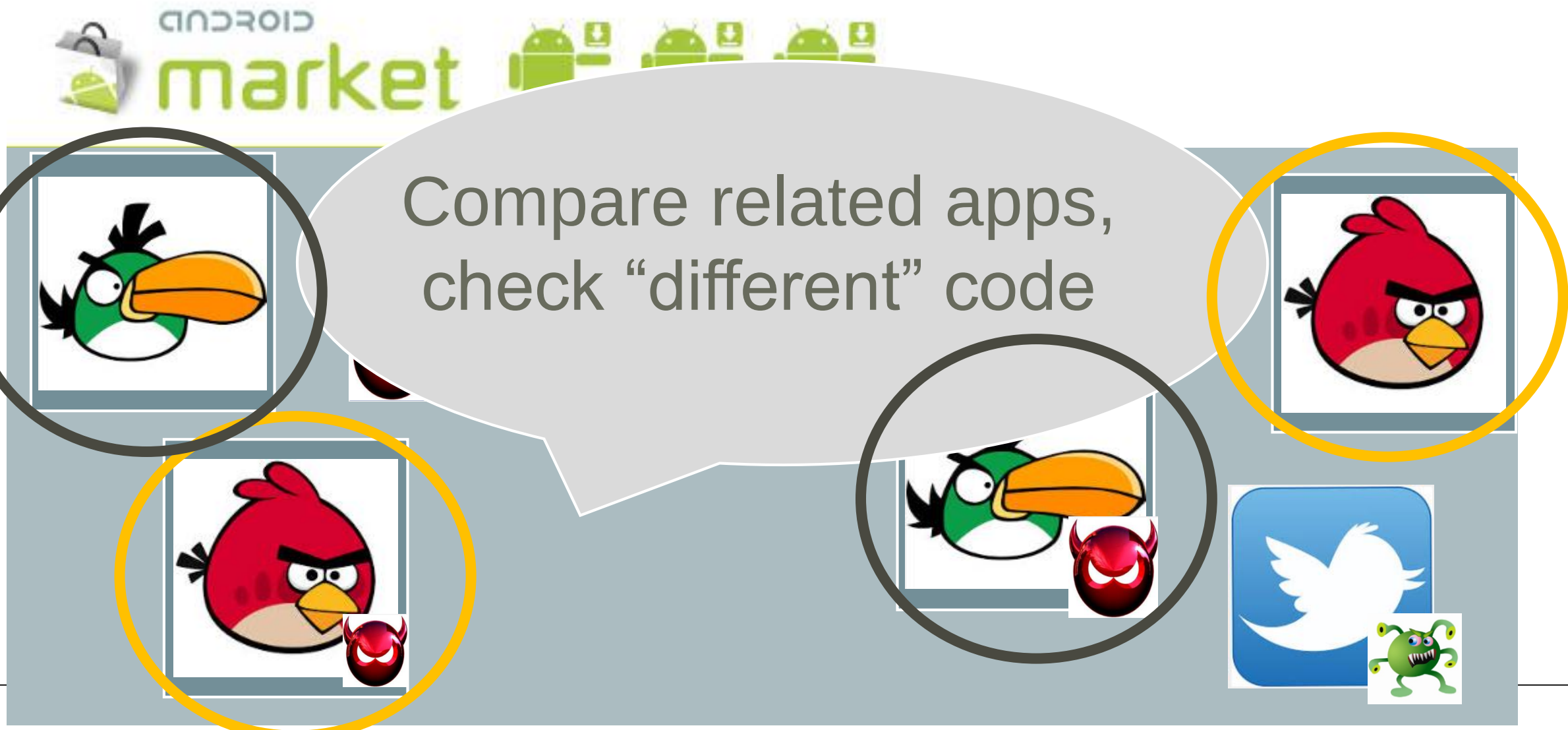
Attackers like to **piggyback** the **same** attack payload to different legitimate apps.

http://www.appomicsec.com

Chen et al. Finding Unknown Malice in 10 Seconds: Mass Vetting for New Threats at the Google-Play Scale. USENIX Security 2015. https://www.usenix.org/node/190925

Acknowledgment: slide adapted from Kai Chen's

Results of Repackaging

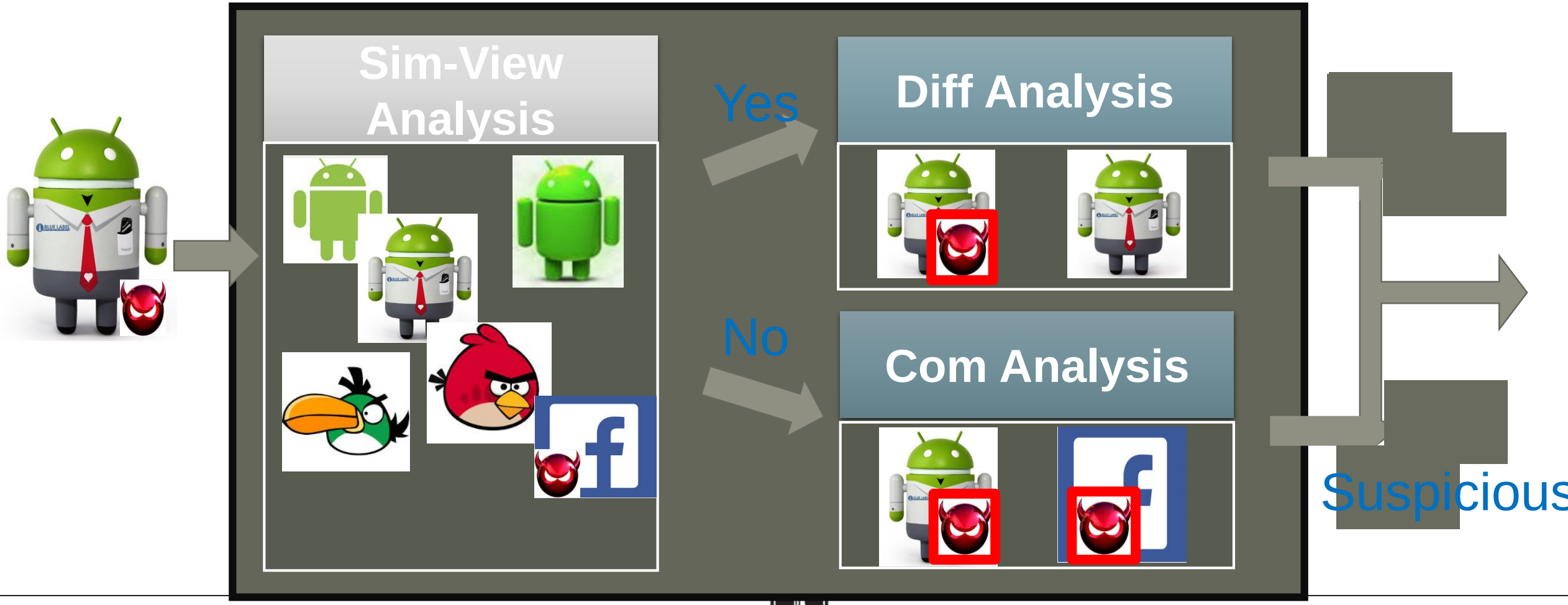


Acknowledgment: slide adapted from Kai Chen’s

Results of Re

Acknowledgment: slide adapted from Kai Chen's

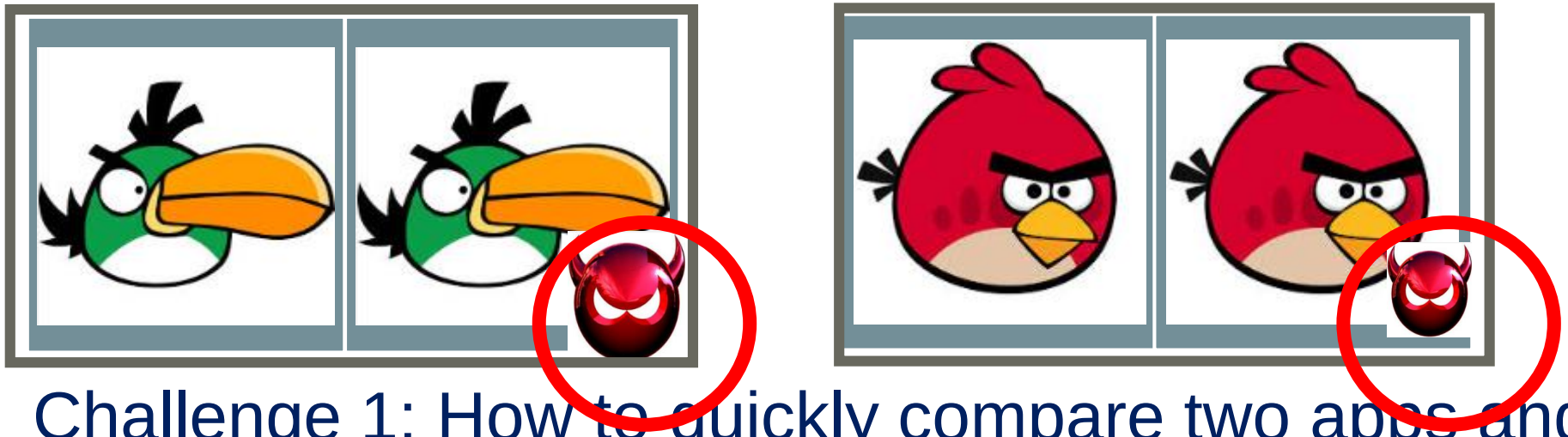
MassVet approach: DiffCom Analysis



Acknowledgment: slide adapted from Kai Chen's

MassVet: Diff Analysis

- For apps having **the same view** and **different signatures**, the different methods between the two apps may be malicious



- Challenge 1: How to quickly compare two apps and find the different methods?
- Challenge 2: Are the different methods malicious?

Chen et al. Finding Unknown Malice in 10 Seconds: Mass Vetting for New Threats at the Google-Play Scale. USENIX Security 2015. https://www.usenix.org/node/190925

Acknowledgment: slide adapted from Kai Chen's

MassVet: Com Analysis

- For the apps with different views, find the common code

- Challenge 1: Are the two apps really unrelated?
- Challenge 2: Is the common code really malicious?

Chen et al. Finding Unknown Malice in 10 Seconds: Mass Vetting for New Threats at the Google-Play Scale. USENIX Security 2015. https://www.usenix.org/node/190925

Acknowledgment: slide adapted from Kai Chen's

Putting Pieces Together

informal: **app description**, etc.

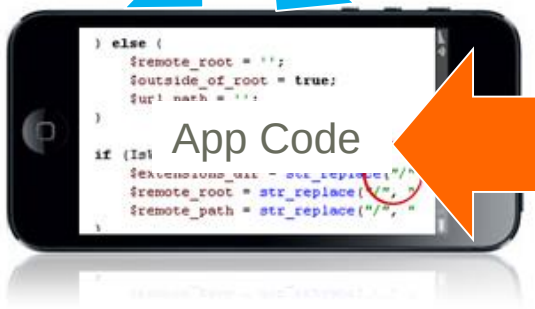
permission list, etc.

APP DEVELOPERS

WHYPER

App Functional Requirements

App **Security** Requirements



User Functional Requirements

User **Security** Requirements

APP USERS

AsDroid

AppContext

MassVet

Pre-Installed Apps/Malware

Chinese Android smartphones now shipping with pre-installed malware

Share this article:

Rather than wait for the user to do it himself, middle men in the Chinese mobile phone industry are pre-stalling malware according to G Data.

According to a newly published report on mobile malware from researchers at G Data, "well over 26" smartphones have been discovered shipping complete with pre-installed malware in the device firmware.

Earlier this year the same company revealed the presence of adware on Android devices, along with 'potentially unwanted programs' or PUPs. Now it says that monitoring applications – aka spyware – to collect data without the smartphone owner realising, along with other malware, is also becoming a problem on certain Chinese handsets.

Some mobile phones are coming pre-installed with toxic software

http://www.scmagazineuk.com/chinese-android-smartphones-now-shipping-with-pre-installed-malware/article/436631/

Spyware Capabilities

The spyware is capable of doing the following actions:

- Listening in to telephone conversations
- Accessing the Internet
- Viewing and copy contacts
- Installing unwanted apps
- Asking for location data
- Taking and copying images
- Recording conversations using the microphone
- Sending and reading SMS/MMS
- Disabling Anti-Virus software
- Listening in to chats via messaging services (Skype, Google+)
- Reading the browser history

http://thehackernews.com/2015/09/android-smartphone-malware.html

Pre-Installed Apps/Malware: Middlemen

- “According to the G Data researchers, there is unlikely to have been anything accidental about the malware it discovered pre-installed on at least **26 different smartphones** from manufacturers including Huawei, Lenovo and Xiaomi.”
- “Which isn't to say the security firm thinks that the manufacturers are the perpetrators here, far from it. In fact, G Data reckons it is down to **'middlemen**' in the distribution chain who are looking to add to their revenue by making "additional financial gains from stolen user data and enforced advertising".”

http://www.scmagazineuk.com/chinese-android-smartphones-now-shipping-with-pre-installed-malware/article/436631/

Pre-Installed Apps/Malware: Removal

Removal of Spyware Not Possible

The pre-installed spyware, disguised in popular Android apps such as ***Facebook*** and ***Google Drive***, can not be removed without unlocking the phone since it resides inside the phone's firmware.

> *"Over the past year, we have seen a significant [growth] in devices that are equipped with firmware-level [malware and spyware] out of the box which can take a wide range of unknown and unwanted actions,"* Product Manager **Christian Geschkat** from G Data said in a statement.

http://thehackernews.com/2015/09/android-smartphone-malware.html

http://www.gsmarena.com/samsung_lets_users_delete_preinstalled_apps_in_china_in_light_of_lawsuit-blog-13348.php

Internet of Things Security: Mobile or Not

“Internet of Things” security is hilariously broken and getting worse

Shodan search engine is only the latest reminder of why we need to fix IoT security.

by J.M. Porup (UK) - Jan 23, 2016 9:30am CST

Shodan, a search engine for the Internet of Things (IoT), recently launched a new section that lets users easily browse vulnerable webcams.

The feed includes images of marijuana plantations, back rooms of banks, children, kitchens, living rooms, garages, front gardens, back gardens, ski slopes, swimming pools, colleges and schools, laboratories, and cash register cameras in retail stores, according to Dan Tentler, a security researcher who has spent several years investigating webcam security.

"It's all over the place," he told Ars Technica UK. "Practically everything you can think of."

We did a quick search and turned up some alarming results:

http://arstechnica.com/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/

Internet of Things Security: Mobile or Not

- “The cameras are vulnerable because they use the Real Time Streaming Protocol (RTSP, port 554) to share video but have no password authentication in place. The image feed is available to paid Shodan members at images.shodan.io. Free Shodan accounts can also search using the filter port:554 has_screenshot:true.”
- “Shodan crawls the Internet at random looking for IP addresses with open ports. If an open port lacks authentication and streams a video feed, the new script takes a snap and moves on.”

http://arstechnica.com/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/

Internet of Things Security:
The curse of the minimum viable product

- “Tentler told Ars that webcam manufacturers are in a race to bottom. Consumers do not perceive value in security and privacy. As a rule, many have not shown a willingness to pay for such things. As a result, webcam manufacturers slash costs to maximize their profit, often on narrow margins. Many webcams now sell for as little as £15 or $20.”
- “"The consumers are saying 'we're not supposed to know anything about this stuff [cybersecurity]," he said. "The vendors don't want to lift a finger to help users because it costs them money."”

http://arstechnica.com/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/

(Mobile) Privacy vs. Utility: A Balancing Act

- A likely scenario for a professor
  - **Student A**: “May I record our 1-on-1 meeting so that I don’t miss anything?”
  - **Professor**: “Hmmhh… OK… but please don’t post it on public domain or redistribute it…”
  - Hopefully….
- Mobile utility apps: app store management, Input method, IME (input method editor)
  - even non-mobile ones: medical devices, search engines, ….
- Assurance case for privacy policy compliance by app or service providers

Sen et al. Bootstrapping Privacy Compliance in Big Data Systems, Oakland 2013.
http://research.microsoft.com/apps/pubs/default.aspx?id=208626

User Expectations in Mobile App Security

informal: **app description**, etc.

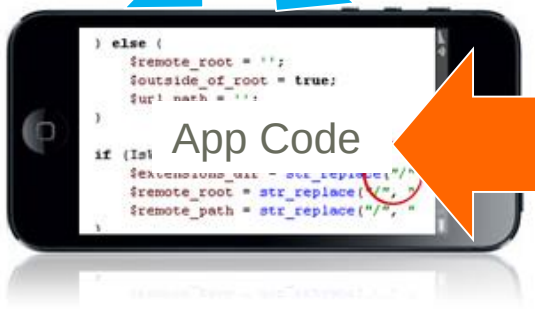
permission list, etc.

APP DEVELOPERS

WHYPER

App Functional Requirements

App **Security** Requirements



AsDroid

AppContext

MassVet

User Functional Requirements

User **Security** Requirements

APP USERS

User Expectations in Mobile App Security

informal: **app description**, etc.

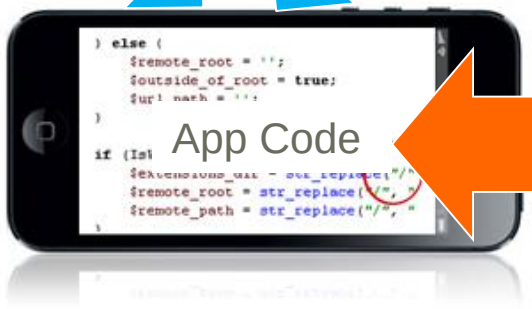
permission list, etc.

APP DEVELOPERS

WHYPER

App Functional Requirements

App **Security** Requirements



User Functional Requirements

User **Security** Requirements

APP USERS

AsDroid

AppContext

MassVet

NSA SoS Lablet, NSF Medium CNS-1513939,
Google Faculty Research Award

illinois.edu

taoxie@illinois.edu