Smartphone Security and Privacy

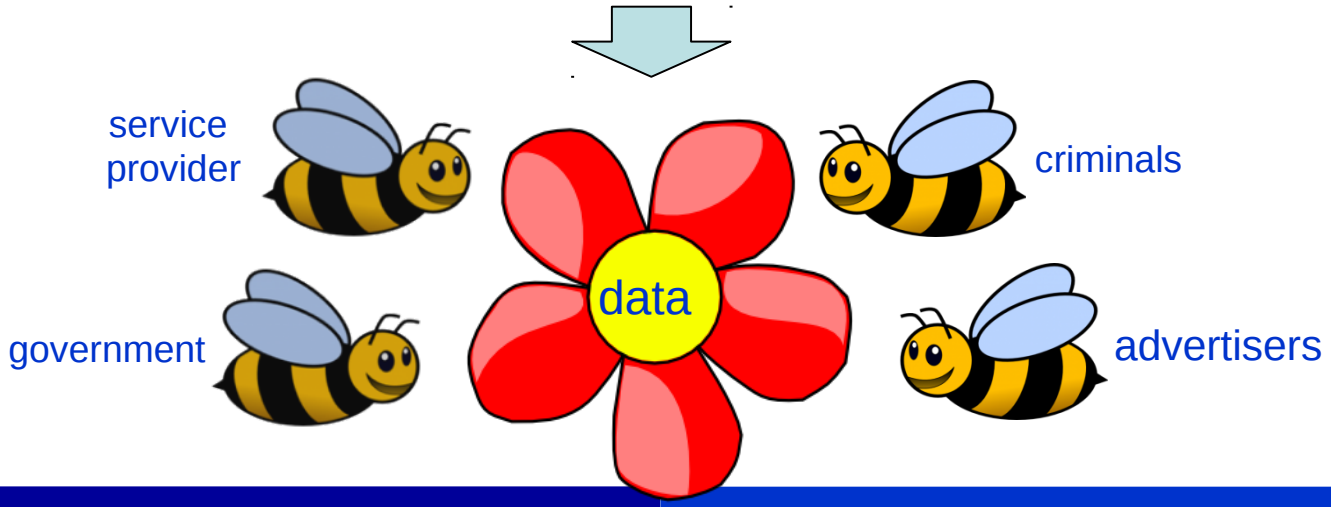
Android security model;

Privilege delegation attack and solution;

The issue

“Security and Privacy in the smartphone age”

- **Is important because smartphones are**
  - undoubtedly becoming ubiquitous
    - 4 time faster than mobile phone market (IDC report)
  - more than just a phone or a desktop computer
  - increasingly with new functionalities
    - i.e., NFC-enabled smartphone as payment tokens (Google Wallet)
  - …



How NFC phones can steal your credit card info

- http://www.youtube.com/watch?v=EKks3vfiy6Q

The issue

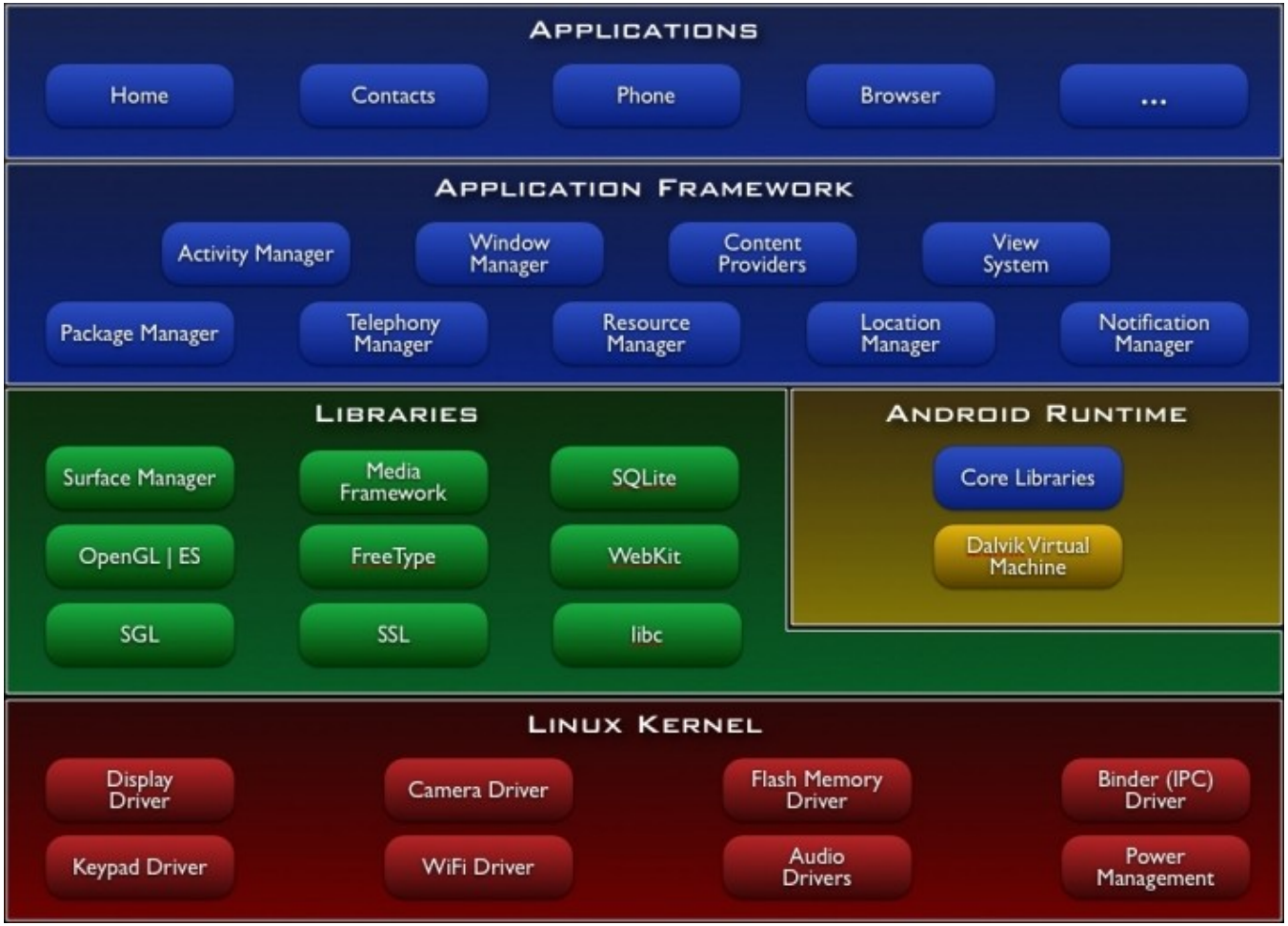
"Security and Privacy in the smartphone age"

- **Will become even more important**
  - When we shift to a mobile, cloud-based computing world
    - Increased risk of private data falling prey to snooping by
      - the government, private hackers, or the cloud service provider itself
    - Still cloudy on whether server-side data is protected by law, e.g., the Fourth Amendment
  - When users are continuously supplied with unlimited amounts of free apps
    - Apps gather sensitive phone/user information
    - Apps may contain malware
      - Reputable apps can be repackaged and injected with malicious links

Targets to Attackers

- Various forms of security threats exist on smartphone platforms
  - Apple's iOS
    - Apple utilizes a vetting process (application review) which is secret
    - Less malware found so far, but a lot of permission misuses and privacy invasion
  - Android
    - A much complete permission system
    - An opener platform, thus much harder to control

Android Architecture



Android Applications

- Usually, Android applications consist of separated modules, or components. Components communicate through the mechanism of Inter Component Communication (ICC).

- Android applications may utilize libraries written in native code via JNI
  - May bypass the security provided by the Java programming language

Android Security Mechanisms

- Discretionary Access Control
  - Inherited from Linux
  - Each file is assigned access rules for three sets of subjects
    - User, group, or everyone
  - Each subject set may have permissions to read, write, and execute a file

Android Security Mechanisms (cont.)

- Sandboxing
  - Sandboxing isolates applications form each other and from system resources.
  - In Android’s sandboxing, each application is assigned a unique ID.
  - An application can only have access to files owned by itself, or files of others which are explicitly announced to be public for others

Android Security Mechanisms (cont.)

- Permission mechanism
  - Security sensitive interfaces are protected by standard Android permissions. (e.g. PHONE_CALLS, INTERNET, SEND_SMS)
  - Required permissions of an app are written in a *Manifest* file. The permissions should be confirmed by user upon installation.
  - At runtime, when an ICC call is requested by a component, a reference monitor checks whether the application has proper permissions
    - Application developers may also add reference monitors into their own applications to verify permissions granted to the ICC call initiator

Android Security Mechanisms (cont.)

- Accessibility of components
  - Components can be public or private
  - Private components
    - Accessible only by components within the same applications
  - Public components
    - Reachable by other applications
      - Full access can be limited by requiring calling applications to have specified permissions

Android Security Mechanisms (cont.)

- Application signing
  - Use cryptographic signature to verify the origin of applications
    - Developers have to sign their apps
  - This enables signature-based permissions
    - Applications from the same origin (i.e., signed by the same developer) share the same UserID
  - A certificate of the signing key can be self-signed and does not need to be issued by a CA

Issues with the current practice

- Major manufactures employ **application permissions** to prevent sensitive data from unauthorized access
  - Sensitive: GPS, camera, microphone, SMS, ….

- However,
  - It relies upon user diligence and awareness
  - Permissions are granted **all-at-once and only at installation time**
    - Subsequent permission check is transparent to users
  - Permission check can be circumvented through **permission attacks**

Permission Escalation Attack and One Solution

- IPC Inspector
- https://plus.google.com/photos/110581955720098741626/albums/5638277509860549393/5643028883450066674

Other solutions to permission delegation attacks

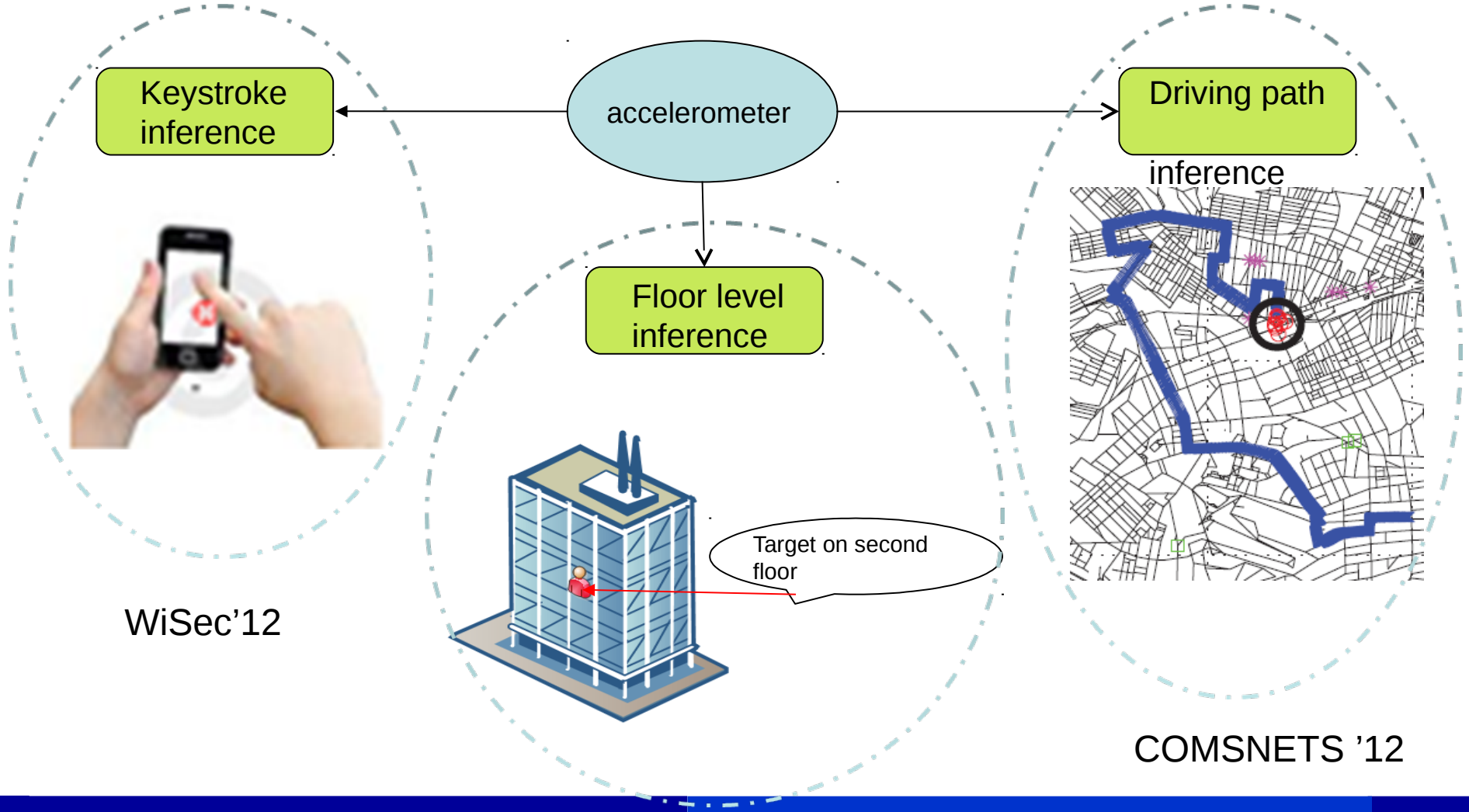
- DroidChecker (WiSec'12)
  - Parse the AndroidManifest.xml to find out if
    - The application uses at least one permission, AND,
    - There exists an activity or service component that does not require any permission and is publicly visible
  - Components satisfying both conditions have the potential of capability leak (as being victim of permission delegation attack)
- TaintDroid (OSDI'10)
  - Dynamically taint-tracking of the flow of privacy sensitive data
  - Monitors in real-time how applications access and manipulate users' personal data

Issues with the current practice

- Even sensitive data can be protected, is it enough? how about non-sensitive data?
  - Non-sensitive: accelerometer, proximity sensor, light sensor, ...

New privacy attacks

non-sensitive data can reveal **sensitive** information!!!



The challenges

- **Understand the implications of various data and their fusion on privacy**
  - Non-sensitive data can reveal sensitive information
  - Non-sensitive data, collected over **a sufficiently long time**, can reveal sensitive information
  - Multiple non-sensitive data can reveal sensitive information

- **Communicate the result to users in a comprehensible way**
  - To assist them to have **controlled release** of personal information
    - Privacy is culture-dependent, individual-dependent, time-dependent, situation-dependent …

- **Develop automatic and adaptive defenses**
  - to satisfy the requirement for controlled release of personal information

Improve Users’ Comprehension of Android Permissions

- **Presentation**
  - Enhancing Users' Comprehension of Android Permissions
  - By Padmini

Other ideas (please comment!)

- Use a numerical value to represent the risk level associated with an app
  - Called as risk index
  - Index can be at different granularity level
    - Summary index, medium index, detail index
  - Where to collect risk information?
    - User review, static analysis, dynamic analysis
  - How to model risk index?

- Use nutrition label to represent risks associated with each resource