Mobile Security

Mustakimur R. Khandaker

Mobile Threats and Attacks

Mobile devices make attractive targets:

- People store much personal info on them: email, calendars, contacts, pictures, etc.
- Sensitive organizational info too…
- Can fit in pockets, easily lost/stolen.
- Built-in billing system: SMS/MMS (mobile operator), in-app purchases (credit card), etc.
    - Many new devices have near field communications (NFC), used for contactless payments, etc.
    - Your device becomes your credit card.
- Location privacy issues.

NFC-based billing system vulnerabilities.

Android Platform Model

Architecture components:
- Operating system, runtime environment.
- Application sandbox.
- Exploit prevention.

Permission system:
- Granted at install time.
- Checked at runtime.

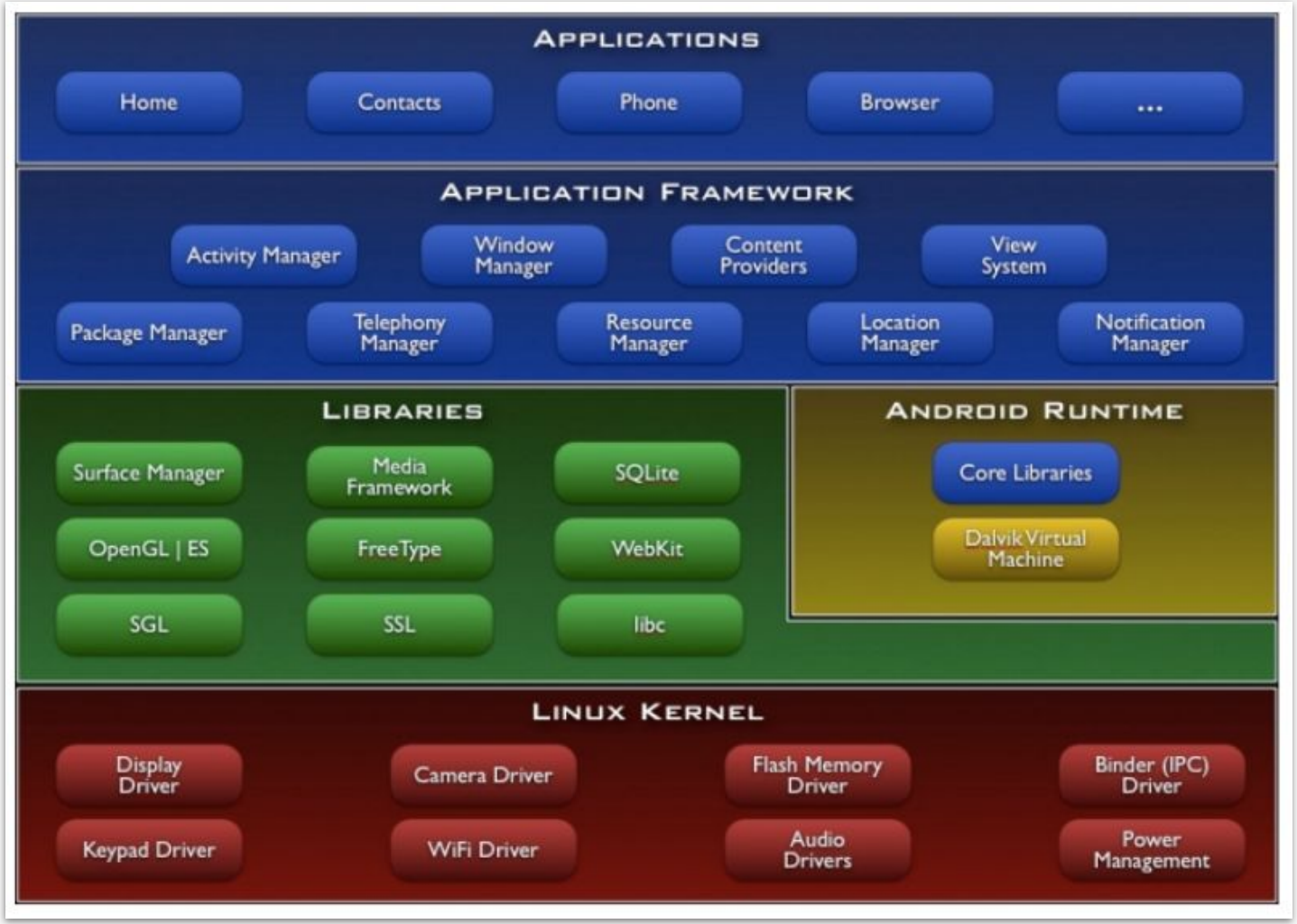
Inter-app communication:
- Intent system.
- Permission redelegation (intent input checking).

Linux kernel, browser, SQLite database.

Software for secure network communication.
- OpenSSL, Bouncy Castle crypto API and Java library.

C language infrastructure.

Java platform for running applications.
- **Dalvik bytecode**, virtual machine / Android runtime (ART).


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Exploit Prevention

Open source: public review, no obscurity.

Goals:

- Prevent remote attacks, privilege escalation.
- Secure drivers, media codecs, new and custom features.

Overflow prevention:

- ProPolice stack protection
  - First on the ARM architecture.
- Some heap overflow protections.
  - Chunk consolidation in DL malloc (from OpenBSD).

ASLR:

- Avoided in initial release.
  - Many pre-linked images for performance.
- Later developed and contributed by Bojinov, Boneh.

Application Sandbox

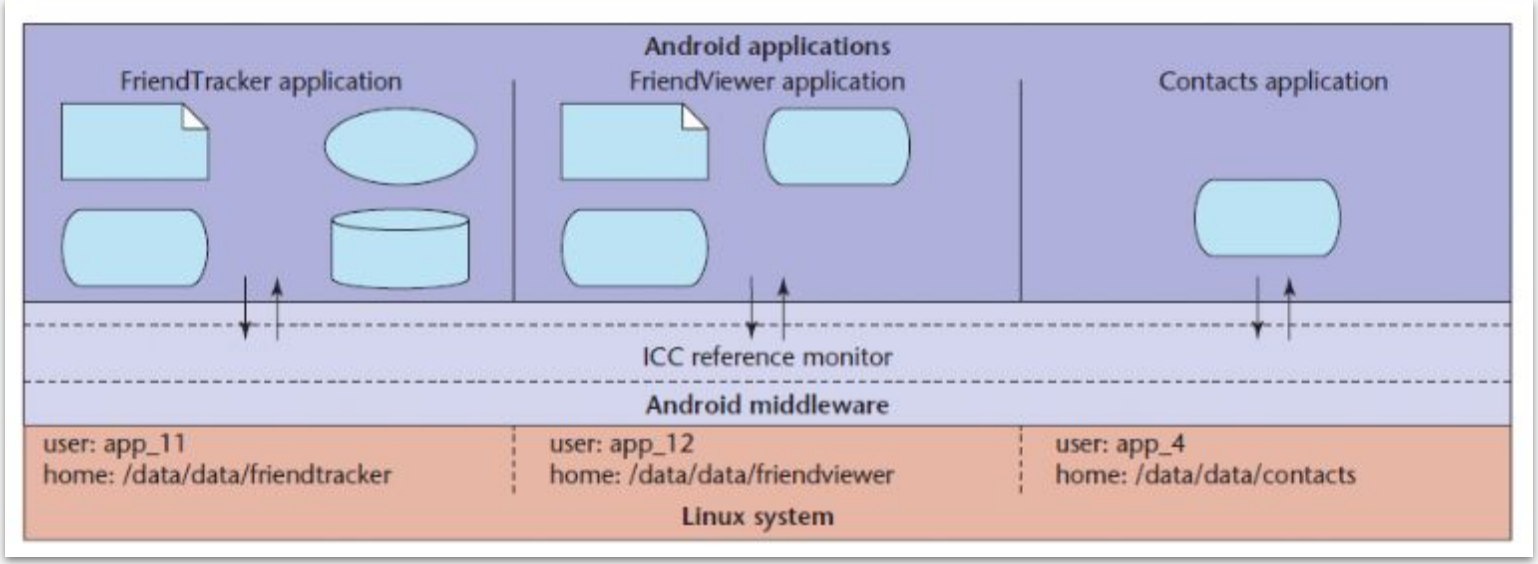
Application sandbox:
- Each application runs with its UID in its own runtime environment.
  - Provides CPU protection, memory protection.
  - Only ping, zygote (spawn another process) run as root.

Applications **announce permission requirement**:
- Create a whitelist model – user grants access at install time (recently changed).

Communication between applications:
- **May share same Linux user ID**.
  - Access files from each other.
  - May share same Linux process and runtime environment.

Or communicate through application framework:
- **"Intents,"** reference monitor checks permissions.

Android Intents



Layers of security:

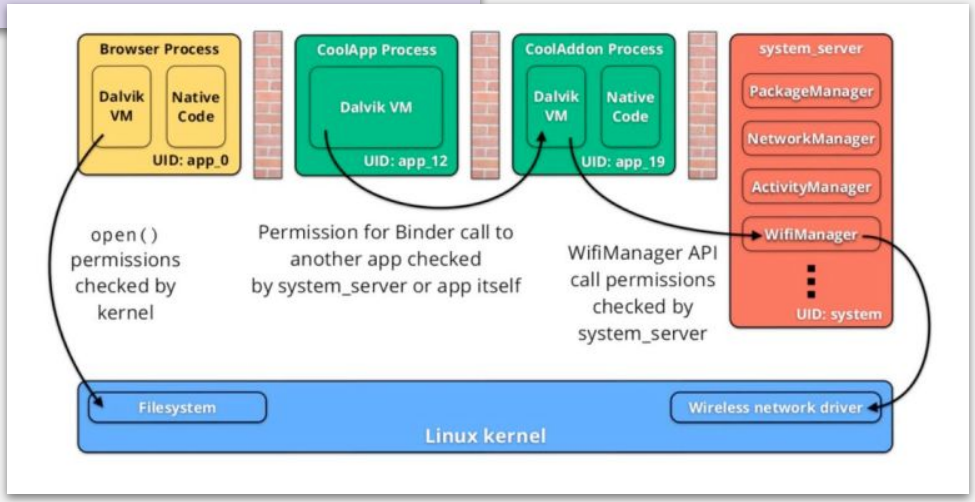
- Each application executes as its own user identity.
- Android middleware has reference monitor that mediates the establishment of inter-component communication (ICC).

Android Permissions

Example of permissions provided by Android:

- “android.permission.INTERNET”
- “android.permission.READ_EXTERNAL_STORAGE
- “android.permission.SEND_SMS”
- “android.permission.BLUETOOTH”

Also possible to define custom permissions.



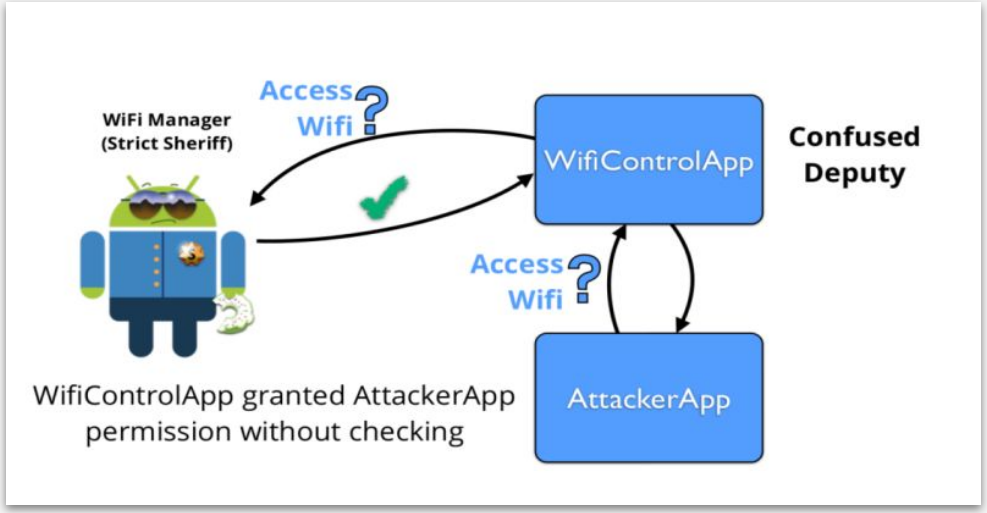
Attack: Permission Redelegation

An application without a permission gains additional privileges through another application.

- Example of the **“confused deputy”** problem.

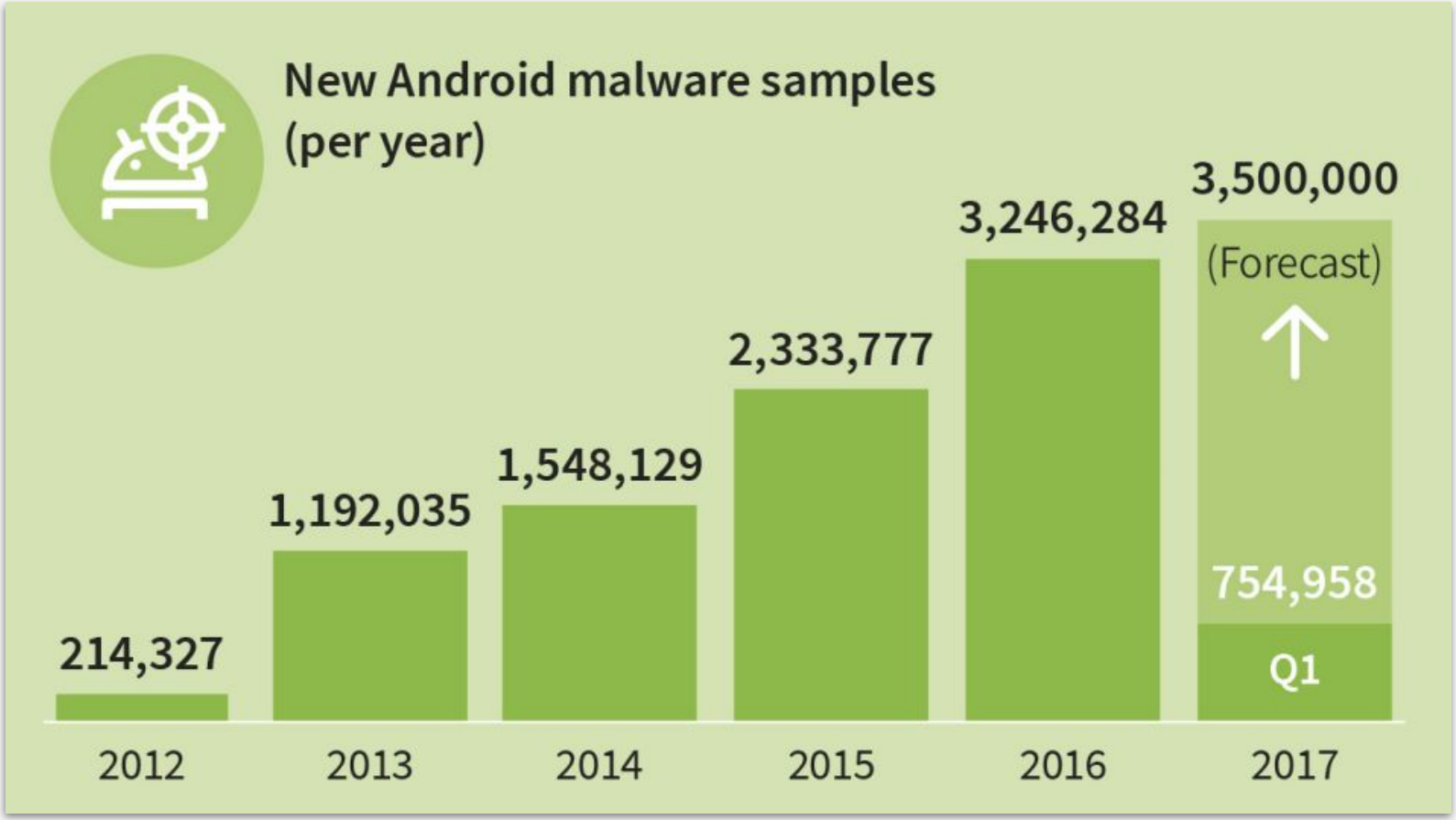
App w/ permissions exposes a public interface.

- Study in 2011, examine 872 apps.

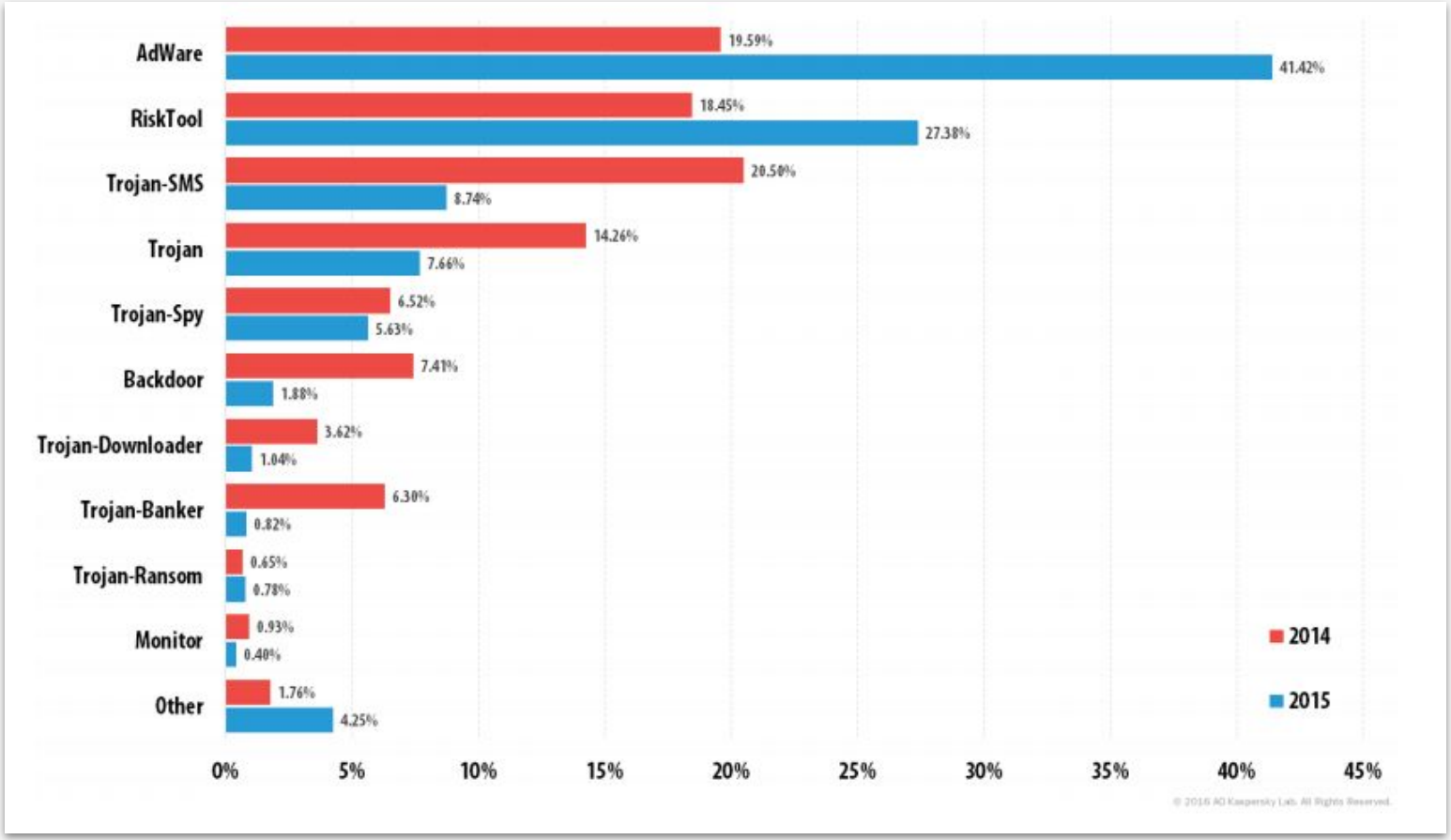


- **320 of these (37%)** have permissions and at least one type of public component.
- **Construct attacks using 15 vulnerabilities in 5 apps.**

Android Malware



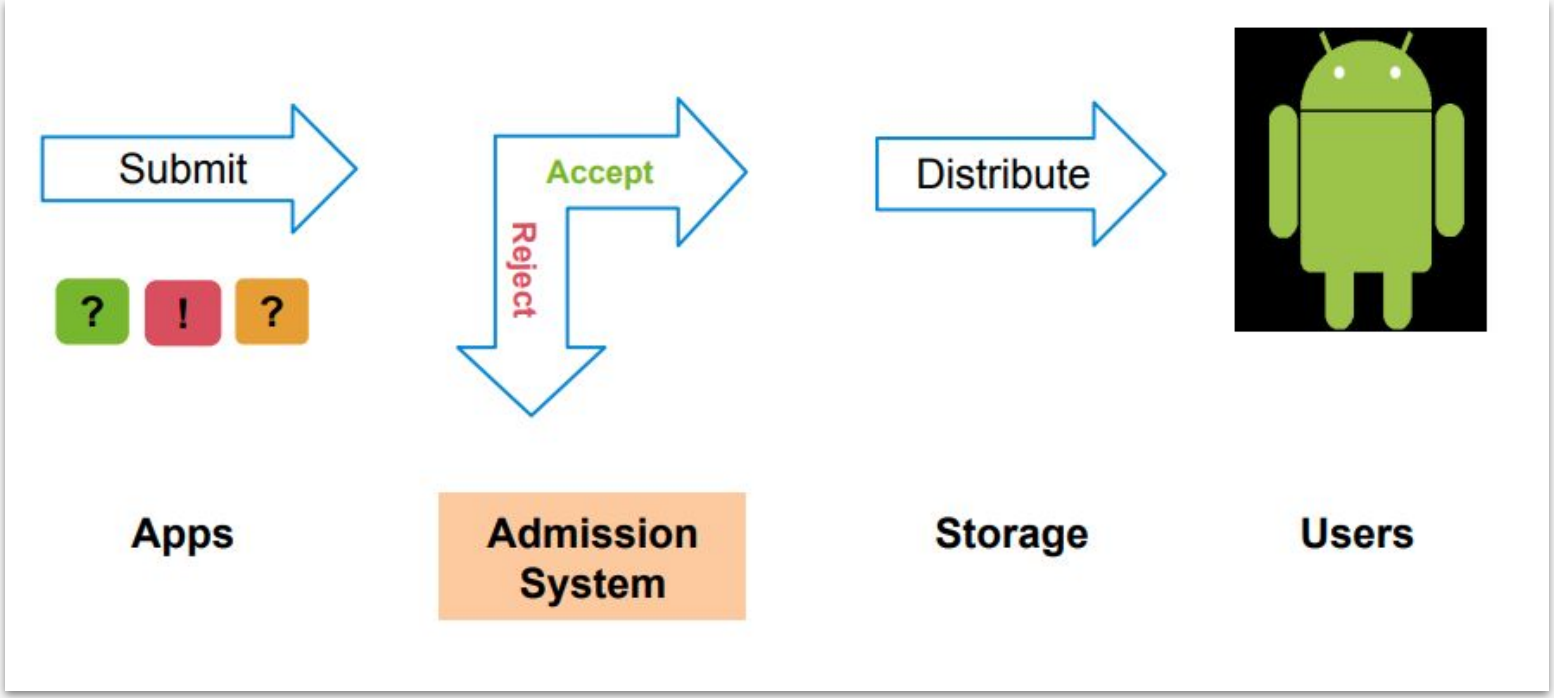
Trends by Categories



Malware Detection

Identifying malware:

- Detect at app store rather than on platform.



Policies

State acceptable/unacceptable behaviors:

- Data Theft: What personal data can leave device?
  - User impact: Data privacy (data-out)

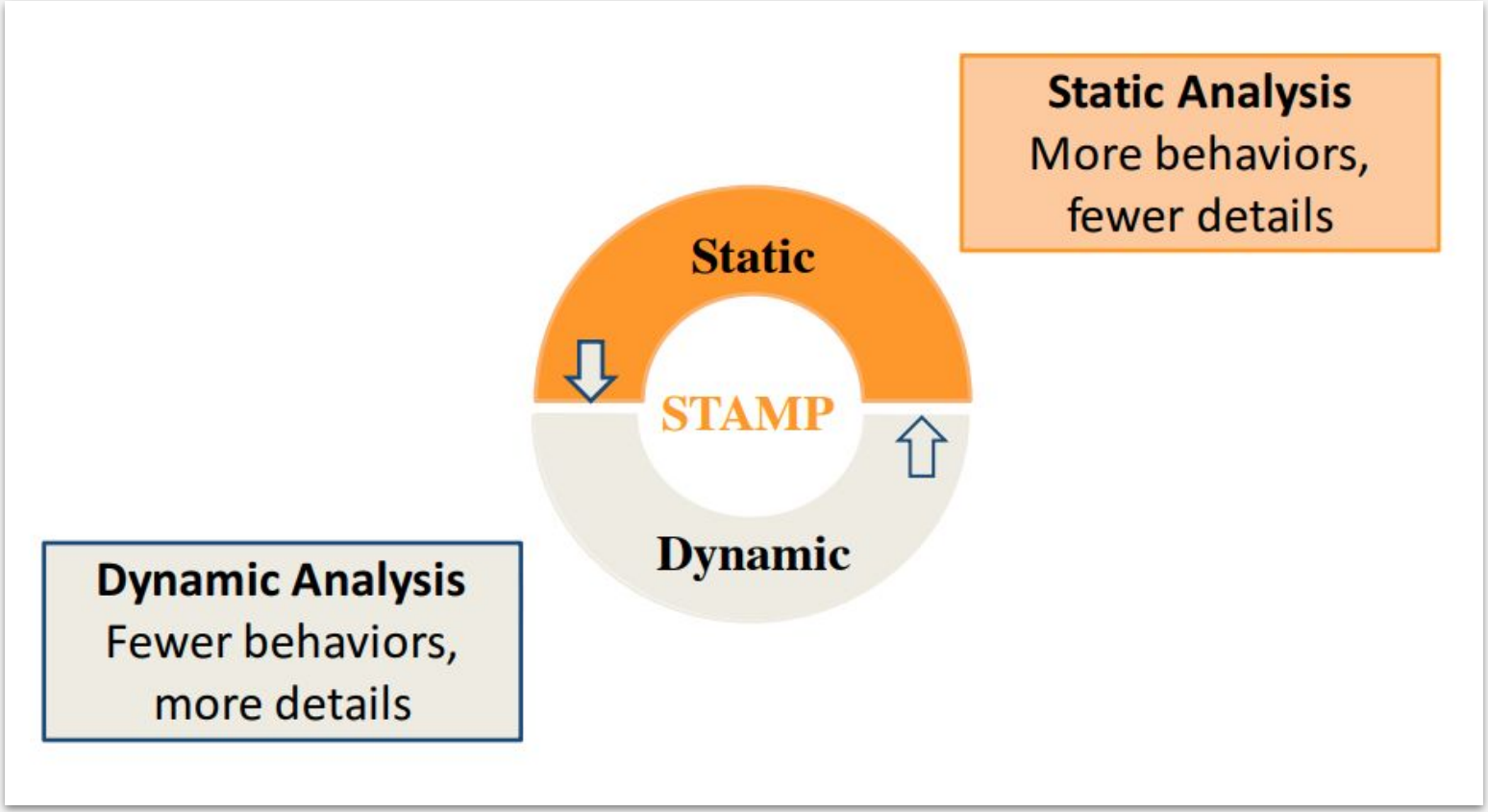
- Device Control: Exploit OS etc.
  - User impact: device integrity (data-in)

- Service Misuse: Premium SMS
  - User impact: $.

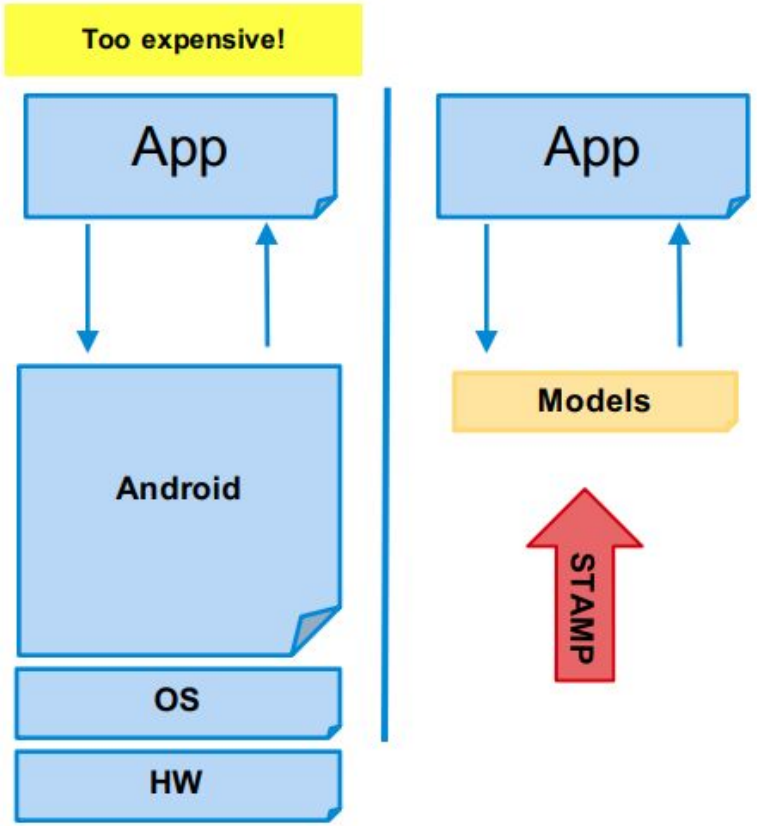
Spam: How many/which type of ads?
- User impact: time.

Others
- No comprehensive taxonomy.

Admission System



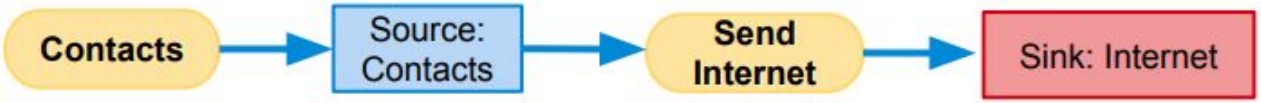
STAMP Approach



- Model Android/Java
  - Sources and sinks
  - Data structures
  - Callbacks
  - 500+ models
- Whole-program analysis
  - Context sensitive

Flow Policies

- Data theft



- Privacy policies
  - Avoid liability, protect consumer privacy

Privacy Policy
This app collects your:
Contacts
Phone Number
Address

- Injection vulnerabilities

< Mobile Security />