ANDROID™ OS Security

A brief introduction to the Android Operating System and its security

The ANDROID™ OS

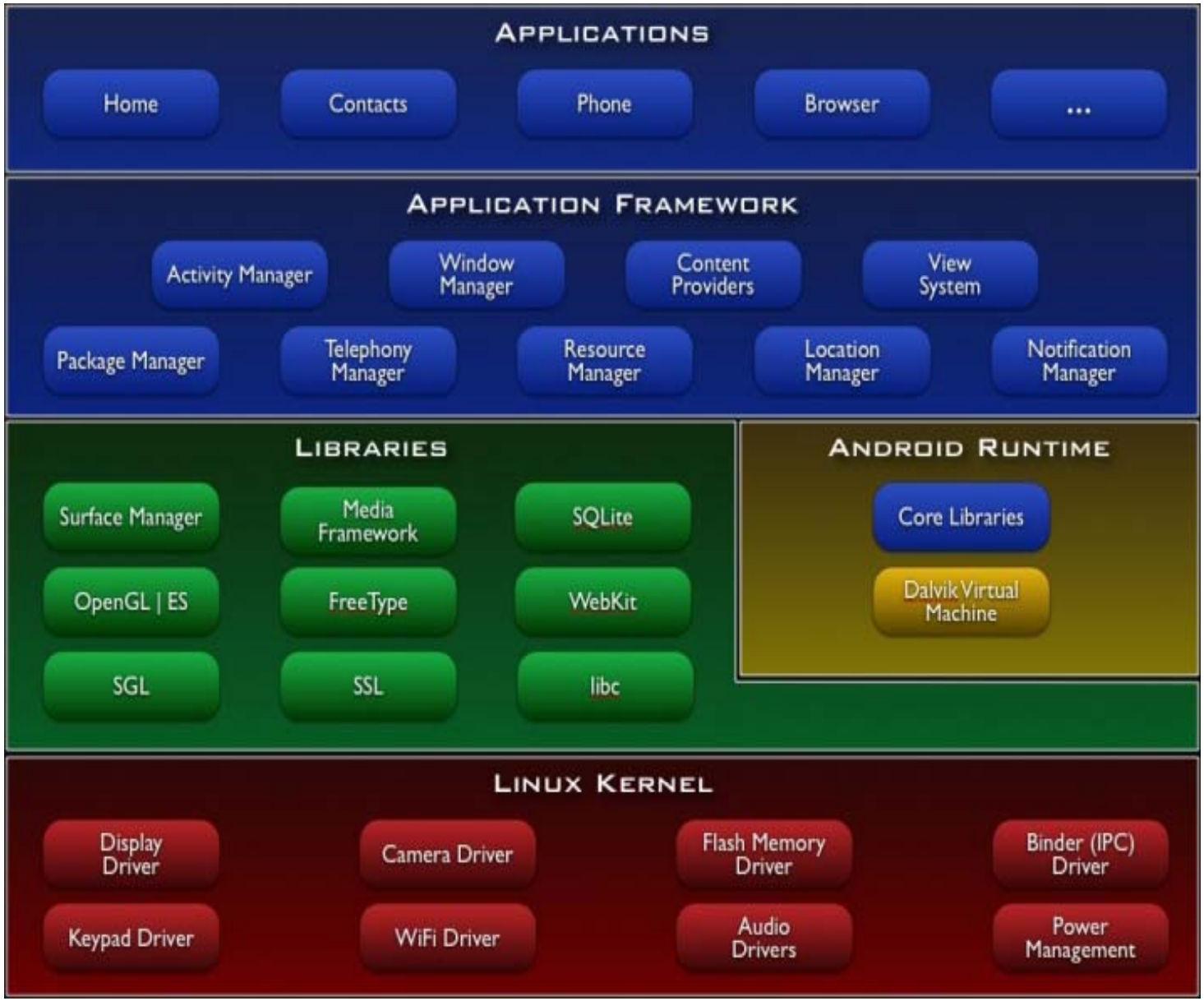
- History
  - Google acquires mobile software startup Android™ in 2005
  - Open Handset Alliance officially starts on November 5th, 2007
  - Android™ 1.0 source and SDK released in Fall 2008 (http://www.android.com/timeline.html)

The ANDROID™ OS

- Versions
  - 1.1 February 2009
  - 1.5 (Cupcake) April 2009
  - 1.6 (Donut) September 2009
  - 2.0/2.1 (Éclair) October 2009
  - 2.2 (Froyo) May 2010
  - 3.0 (Gingerbread) Not expected before Q4 2010

The ANDROID™ OS

- System Architecture
  - Linux Version 2.6
  - Davlik Virtual Machine (VM)
  - Application Framework



http://developer.android.com/images/system-architecture.jpg

The ANDROID™ OS

- Applications
  - Applications are Java
  - Applications are run on Davlik Virtual Machine
  - Development done by Android™ SDK
  - Development is open to all
  - User driven Android™ Market

ANDROID™ Security

- Security triad applicability
  - Confidentiality
  - Integrity
  - Availability

ANDROID™ Security

- Android Security
  - Relies on security of it's foundations; Linux, Davlik, and Java.
  - Security Goal: "A central design point of the Android security architecture is that no application, by default, has permission to perform any operations that would adversely impact other applications, the operating system, or the user."

ANDROID™ Security

- Enforcement strategy
  - Application signing and certification.
  - Linux user name base access restriction
  - Capability permissions

ANDROID™ Security

- Application Sandboxes
  - All Applications run as their own Linux user.
  - Several Inter-Process Communication methods:
    - Activities
    - Services
    - BroadcastReceiver
    - ContentProvider
    - Intent
  - Applications utilize a capability like model to protect the system and the user.

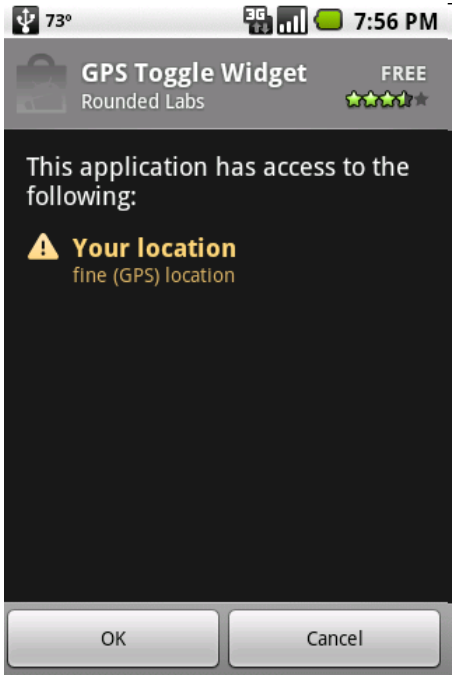
ANDROID™ Security

- Android™ Capabilities and Permissions
  - Capabilities default to safe state
  - Must be explicitly defined to enable capabilities
  - Permissions are static on install
  - Users have open view of permissions

ANDROID™ Security

String	ACCESS_COARSE_LOCATION	Allows an application to access coarse (e.g., Cell-ID, WiFi) location
String	ACCESS_FINE_LOCATION	Allows an application to access fine (e.g., GPS) location

http://developer.android.com/reference/android/Manifest.permission.html



http://www.simplehelp.net/images/quick_gps/img06.png

ANDROID™ Security

- Security Concerns for developers
  - Protect your application, use least privilege principle.
  - If you expose, mediate IPCs
  - Provide maximum availability
    - Minimize memory footprint
    - Minimize battery usage

ANDROID™ Security

- Security Concerns for users
  - Do your research
    - Read reviews.
    - Analyze capabilities/permissions before installing.
    - Use Common sense.
    - http://www.downloadsquad.com/2010/06/28/understanding-the-android-market-security-system/

ANDROID™ Security

- Security Analysis
  - Mediation
  - Verifiability
  - Integrity of TCB

ANDROID™ Security

- Principles of Secure Design
  - Least Privilege
  - Fail Safe Defaults
  - Economy of Mechanism
  - Complete Mediation
  - Defense in depth
  - Open Design
  - Separation of Privilege
  - Least Common Mechanism
  - Psychological Acceptability

Conclusion

- Secure architecture
- Reliance on trust
- As with all things, use your head.

References

Burns, Jesse. “Mobile Application Security on Android.”
blackhat.com. June 2009. Web. 27 July 2010.
<http://www.blackhat.com/presentations/bh-usa-09/BURNS/BHUSA09-Burns-AndroidSurgery-PAPER.pdf>

Android Developers, “Security and Permissions.”
developer.androide.com. 26 July 2010. Web. 27 July 2010
<http://developer.android.com/guide/topics/security/security.html>

Android (operating system) Wiki.
<http://en.wikipedia.org/wiki/Android_%28operating_system%29>

Elgin, Ben. “Google Buys Android for Its Mobile Arsenal”.
businessweek.com. 17 August 2005. Web. 27 July 2010.
<http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm>