Kan vi lita på säkerheten i Android och MeeGo?

Ben Smeets

Lunds Universitet

SUSEC Årsmöte 13 Oktober 2010

Översikt

- Android och MeeGo: vad är det?
- Säkerheten i Android
  - Grundidéen för säkerheten
  - Interaktion och access-kontrol i Android
  - Permissions
- MeeGo: skillnaden mot Android (säkerhetsmässigt)
- Androids platformssäkerhet
- Något om attacker på telefoner
- Några slutsatser - Kan vi lita på säkerheten?

Android och MeeGo: vad är det?

- En platform för mobiler med öppen källkod
  - Källkod: http://source.android.com
- Utvecklingen under kontroll av ~~Open Handset Alliance~~ Google
- Vem som helst kan installera appar
- Utvecklare kan använda sig av ett öppet SDK
  - SDK : http://developer.android.com

Android och MeeGo: vad är det?

- En platform för mobiler med öppen källkod
  - Källkod: http://meego.gitorious.org

THE LINUX FOUNDATION

- Utvecklingen under kontroll av MeeGo Technical Steering Group (TSG). Kommer från Intels Moblin och Nokias Maemo
- Vem som helst kan installera (öppen/stängd)
- Utvecklare kan använda sig av ett öppet SDK
  - SDK:http://wiki.meego.com/Getting_started_with_the_MeeGo_SDK_for_Linux

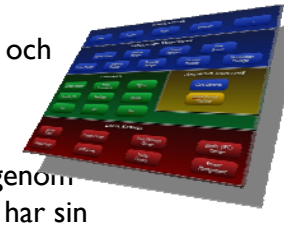
Android: vad är det?

- Modifierad Linux kärna (baserad på 2.6.32, för Android 2.2)
- Använder mer än 90st öppna källkodsbibliotek
- Integrerat WebKit baserad browser
- SQLite för strukturerad datalagring
- OpenSSL
- BouncyCastle
- libc baserad på OpenBSD
- Apache Harmony och Apache HttpClient
- Stöd för många vanliga ljud-, video- och bildcodecs
- APIstöd för kunna implementera I/O till en mobilenhet
- Datakommunikation: Bluetooth, EDGE, 3G, WIFI
- Periferistöd: kamera, video, GPS, kompass, accelerometer, ljud, vibrator

"The basics" i Android

Android säkerhet – grundidéen

- Applikationer utvecklade på andra platformer kan ej köras.
- Applikationer skrivs i ett Javaliknande språk och körs i en egen Dalvik VM- instans.



- Interaktion mellan applikationer begränsas genom en speciell API och genom att applikationer har sin egen (Linux) identitet.
- Genom permissionlabel-tilldelning styr man access till resurser och andra applikationer.

Fundament: Android Package (APK) fil struktur

/META-INF

- **MANIFEST.mf** - sha-1 digest av varje fil av applikationen
- **HelloWorld.sf** - sha-1 digest of each file (baserad på info från Manifest.mf)
- **HelloWorld.rsa** - PKCS#7 RSA signatur av HelloWorld.sf

/res

- **Applikationens resursfiler**

/

- **AndroidManifest.xml** – app information till Android systemet
- **classes.dex** - kompilerad Android Dalvik app:en
- **resources.arsc** – tabel om alla resurser in applikation

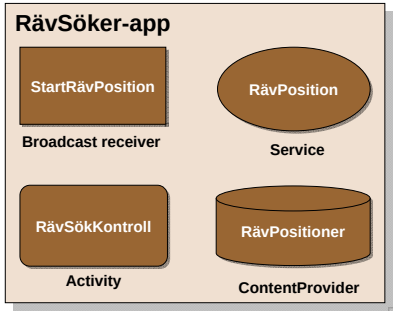
Fundament: Processer och Trådar

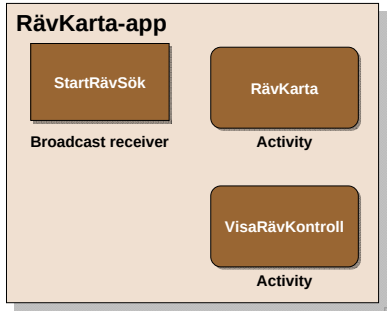
- Varje applikation får sin egen Linux process
- Per default körs varje komponent i processens huvudtråd
- Man kan skapa trådar för längre processer
- Trådar kan skapas i appens kod med standard Java Thread objects

Fundament: Android applikationer byggs enligt en gemensam struktur

- **Activities** står för presentationslagret; en för varje skärm och Views ger UI för en aktivitet.
- **Intents** specifierar vad som ska utföras
- **Services** är bakgrundprocesser utan UI: updaterar data och utlöser händelser
- **Broadcast receivers** tillhandahåller brevlådor för meddelanden från andra applikationer och kan utlösa intents som startar en applikation
- **Content providers** tillhandahåller data/resurser

Android Applikationer - Exempel



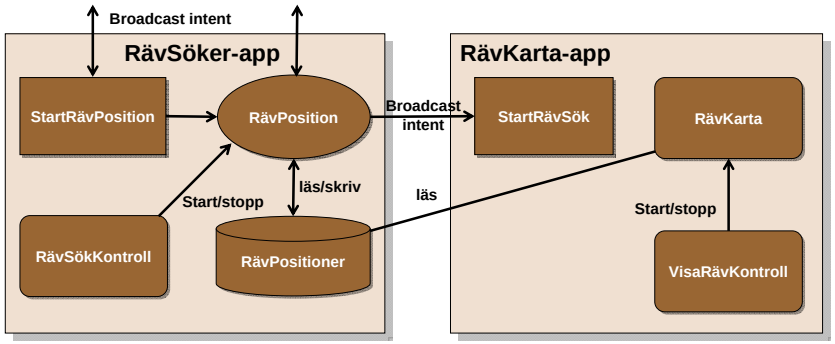


RävSökKontroll : UI för starta/stoppa sökfunktion
RävPosition: Kontaktar extern lokaliserings server
RävPositioner: Spara senasta positioner

Programinteraktion i Android

- De flesta interaktioner i Android är inter-komponent interaktioner (ICC):
  - Intents: meddelande object med måladress och data
  - Actions: själva processen för inter-komponent interaktion
- Androids filosofi är att applikationer kan göra (del av) sina APIer tillgängliga för andra applikationer

Exempel - Interaktion



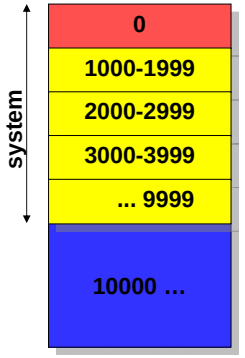
De flesta interaktioner i Android är inter-komponent interaktioner

Applikationssäkerhet - accesskontroll

Accesskontroll i Android

- Android skyddar applikationer på
  - (Linux) system nivå och
  - inter-komponent kommunikations nivå: Rätigheter (Permissions)
- Varje applikation kör som en unik Linux användare under en egen identitet (user ID) vilket begränsar skadeverkan av ett programmerings fel i en applikation:
  - Linux access kontroll (som vanligt)
  - User ID tilldelas (lokalt) vid installation av applikationen

User ID Tabell



Root: Ett fåtal root processer, t ex init, runtime, Zygote,

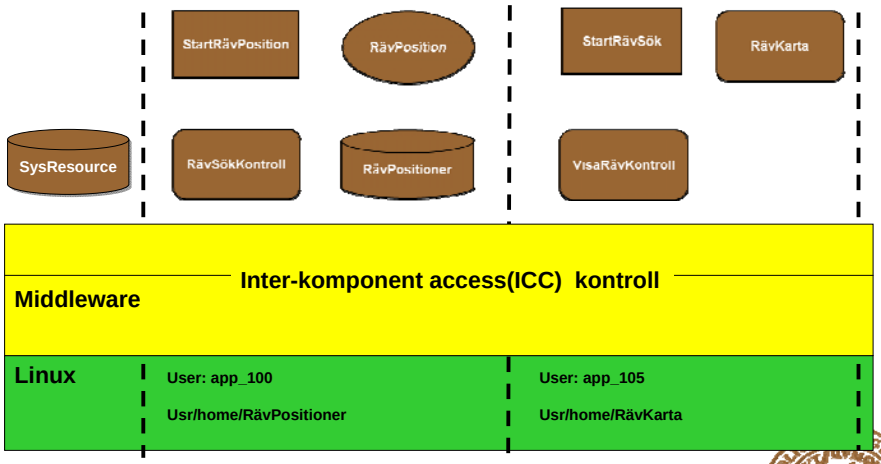
Systemprocesser

ADB och andra debug processer

Android processer med kopplingar till kärnan

Appar får en user ID ≥ 10000

Accesskontroll

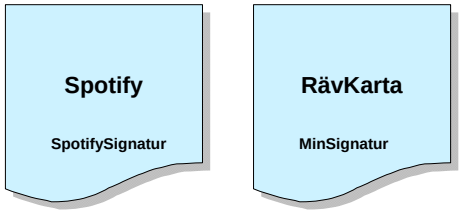


Signerade Applikationer

- För att kunna bestämma applikationens ursprung måste den signeras mot ett själv-signerad certifikat.
- Androids Appstore tar inte emot appar signerad med defaultnyckeln från SDK
  - Dvs det är en policyfråga om SDKs kända nyckel accepteras för en app.
- Två appar kan få samma user ID om de är signerade med samma nyckel/certificat (bör undvikas!)

PKI strukturen och accesskontroll

utan signaturrättigheter



- I standard Android där signaturerrättigheter inte används har signaturer inga (inte mycket i alla fall) säkerhetsbetydelse.
- Signaturen används för att urskillja (identifiera ?) applikations ursprung.

PKI strukturen och accesskontroll

med signaturrättigheter

Android skiljer på fyra kärn platformskomponent typer som har sin egen nyckel

Platform: en nyckel för packet som tillhör "core" platformen

Shared: a nyckel for komponenter vilka är delade i home /contact processen.

Media: a nyckel för packaget som tillhör media/download systemet.

Testkey/releasekey (Application) : default signeringsnyckel

Permissions (1/2)

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.app.myapp" >

    <uses-permission android:name="android.permission.RECEIVE_SMS"
      android:permissionGroup="android.permission-group.COST_MONEY"
      android:protectionLevel="dangerous" />

</manifest>
```

- Rättigheter definieras av en textsträng i en manifestfil.
  flexibelt men kommer användaren alltid begrippa denna sträng; språk, semantiken, etc. Finns fördefinierade tex Manifest.permission_group
- En applikation som vill använda en API måste få tillstånd att använda denna. Detta sker vid installation av applikationen.
- Beroende på API, dess rättigheter och vem som har signerad måste användaren godkänna access till APIn.

Permissions (2/2)

- ICC Accesskontroll sätts vid installation och kan sedan inte ändras: jmfr. regelstyrd accesskontroll (MAC).
- Vid uppdatering av en applikation ska samma nyckel och certifikat användas. Rättigheter ändras inte vid en uppdatering.
- Accessfel kan ge ett felmeddelande till applikationen (men inte alltid). Accessfel registreras felen normalt i systemloggen.

Rättighetsmodell: skydssnivåer

- **normal:**
  ger applikationer automatisk access
- **dangerous:**
  användning av APIn innebär risk och kräver användarens godkännande
- **signature:**
  access enbart om applikationen är signerad mot samma certifikat som applikationen som deklarerad rättigheten
- **signatureOrSystem:**
  som ovan men kan användas fritt av packet i system image (ska användas med försiktighet)

User ID och Fil access

- Filer tillhör normalt en user ID.
- Max två olika Android packet kan har samma user ID om de är signerad mot samma certificat och i manifestet finns ”sharedUserId” attributet.
- Man kan sätta MODE_WORLD_READABLE och/eller MODE_WORLD_WRITEABLE flagor så att alla packet får läsa/skriva från/till filen

MeeGo: skillnaden med Android *(säkerhetsmässigt)*

- MeeGo finns i öppen och stängd (closed) mode, i öppen mode kan utvecklare påverkar säkerhets policyn.
- MeeGo konceptet täcker flera typer av devices.

- MeeGo's säkerhet bygger mer på grupp accesskontroll i Linux, jmfr med rollbaserad accesskontroll.

MeeGo: Aegis (1/2)

- Mer säkerhetsstöd än Android i HW och skydd för nycklar samt användardata
  - ARM TrustZone, Intel Trusted Execution Technology
  - Intel/Nokia bestämmer säkerhetspolicyn
- Integritetsskydd av exefiler (binärer, bibliotek, scripts)
- Säker IPC (integritet + kryptering på D-bus)
- Säkerhetspolicyn kan uppdateras genom speciella uppdateringar

MeeGo: Aegis (2/2)

- Skyddade resurser är identifierade genom resurstokens som är kopplade till extra (eng: supplementary) grupper.
- Om en applikation (process) har rättigheten att göra access till en skyddad resurs så läggs tillhörande grupp id till de extra grupper för applikations process.

  jmfr. rollbaserad accesskontroll
- Process credentials assigner (MeeGo extension i kärnan) ger applikation rättigheter baserad på en credential lista. Som uppdateras vid installation av applikationer

IMPLEMENTATION

Androids platformssäkerhet

- De flesta tillverkare gör viss anpassning vid integrationen av Android på hårdvaran
  - Säkert bootprocess: vilka komponenter som verifieras är olika för olika tillverkare
  - Tillverkare försöker begränsa access till root. Dock har en del tillverkare mycket att lära sig på den punkten
  - Om modemet är ett eget hw subsystem så förblir modemet ganska isolerad från applikations systemet där Android kör imed att RIL gränssnittet begränsar möjligheter till attack

och resultatet blir ….

Något om Attacker på telefoner (1/2)

- Varför hackar man en telefon?
  - Ta bort märkesbegränsningar
  - Avlägsna SIM-lås
  - Ändra IMEI
  - Det är kul (att vara root)

Något om Attacker på telefoner (2/2)

- Inkörsportar för attacker
  - Kvalitetsbrister i mjukvara
  - Ej komplett trustkjedja vid boot
  - Systemdebug-möjligheter kvar
  - Produktions/Service programvara
  - Via periferigränssnitt (inkl GSM som är öppen för falsk basstationsattack)

Kan vi lita på en Androidtelefon?

- Om frågan gäller en produkt så beror svaret på många faktorer (se nästa bild)

- Om frågan gäller om det är möjligt så finns det förutsättningar att besvara frågan positivt.

  Men det krävs
  - Produktgranskning (vad och hur är det implementerad ?)
  - Försiktighet vid användning (val av appar såklart)

Kan vi lita på en Androidtelefon?

- Inga fel i Androids säkerhetsdesign har hittats.
- Systempåverkan av fel i en app eller virus är oftast begränsad genom separationen i Android.
- Vilka appar är installerade ?
- Begriper användaren vad han godkänner vid installation av en app ?
- En rootad Android är i corporate-sammanhang en riskfaktor.
- Oerfarna tillverkare har mer problem med att lyckas med en säker systemimplementering som motstår rootning (men även de stora mislyckas då och då).
- Implementationsfel i Android kan i nödfall rättas av tillverkare i fall Google reagerar långsamt (öppen källkod).
- Telefonens appmiljö byggs på känd grundmjukvara

och MeeGo då?

- Förtidigt att säga hur det blir i praktik.
- Slutna MeeGo produkter har bättre säkerhets förutsättningar än Android produkter

  Men också här krävs
  - Produktgranskning (vad och hur är det implementerad ?)
  - Försiktighet vid användning (val av appar såklart)

Frågor

Ben.Smeets@eit.lth.se

Referenser

- Android developer site pages: (too many to list here)
- Understanding Android Security, William Enck, et. al, Security & Privacy January/February 2009
- Developing Secure Mobile Applications For Android, iSEC Partners 2008
- Maemo 6 Platform Security, Elena Reshetova, Nokia, maemo.org
- git://gitorious.org/meego-platform-security/aegis-examples.git
- git://gitorious.org/meego-platform-security/librestok.git
- http://meego.gitorious.org/meego-platform-security/pages/Functionality