Android Security

Softwareprojekt Mobilkommunikation

Institut für Informatik
FU Berlin

Inhalt

- Zusammenfassung eines Vortrages von Christian Küster
  - Arbeitet bei IT-Unternehmen tarent
  - Vortrag gehalten am 25.06.2009 auf dem LinuxTag

- Berechtigungssystem
- Zugriffsschutz
- Erweiterungen

Einleitung

- Es gibt nur ein Entwicklermodell (G1)
- Retail-Geräte sind sehr geschlossen

- Sicherheit der Android-Plattform war Entwurfsziel
  - Im Sinne eines Sicherheitsmodells
  - Nicht im Sinne von „Code-Review“

- Motivation
  - Smartphones enthalten viele sensible Daten
  - Kontakte, Browser-History, SMS, …
  - Quelloffenes System weckt Neugier

Berechtigungssystem

Freie Universität Berlin

- Android verwendet modifizierten Linux-Kernel
  - System-V-IPC entfernt
  - NetBSD-Standard-C-Bibliothek „Bionic“
- Binäre Einschlüsse vorhanden
  - WLAN-Kernel-Modul
  - GPS-Treiber
  - Audio-Treiber
  - Kamera-Treiber
- Alle Applikationspakete sind vom Entwickler signiert
  - Applikationsdeployer weißt automatisch neue UID zu
  - UID-Sharing nur bei gleicher Entwicklersignatur
- Kein /etc/passwd und /etc/shadow

Zugriffsschutz

Freie Universität Berlin

- Anwendungen benötigen für bestimmte Aktionen Berechtigungen
  - Deny-All-Ansatz
  - Berechtigungen werden statisch vergeben
  - AndroidManifest.xml
  - Benutzer entscheidet bei Installation
- Kernel wurde erweitert
  - Beispiel: INTERNET_ACCESS
  - Kernel überprüft Gruppenzugehörigkeit „inet“
- Anwendungen legen Dateien unter ihrer UID ab
  - /data/data/<paketname>/
    - Nur /data ist schreibbar
    - Alle Systemprogramme und Bibliotheken in /system
    - /system und Root-Verzeichnis sind nur lesbar
  - SD-Karten werden mit „noexec“ eingehangen
  - Updates werden durch ein Overlay ermöglicht

Erweiterungen

Freie Universität Berlin

- Sicherheitsfunktionen erweitern
  - Erfordert Flash-Möglichkeit und Änderungen am Kernel
  - VPN
    - Kernel-Config-Änderungen notwendig
    - Fehlende Funktionen in Bionic
  - Dateiverschlüsselung
    - Außer DM-Crypt keine Lösung im Kernel
    - EcryptFS benötigt Shared-Memory
    - Sinnlos, da Geräte meist angeschaltet
  - SmartCards
    - Fehlender Leser für SmartCards
    - Mögliche Lösung: USB On-the-go
    - PKCS11-API wie in Sun JVM fehlt
  - GnuPG benötigt Random Number Generator
    - Vorhandener RNG bekommt zu wenig Entropie
    - Programm bleibt deshalb stehen