Gegenüberstellung der technischen iPhone und Android Sicherheit

Welchen Sicherheitsgewinn bringt Mobile Device Management?

54. DFN-Betriebstagung – AK Mobile IT
15.3.2011 – Berlin

Fraunhofer SIT

Überblick

- iOS Schutzfunktionen
  - Sandbox Grundlagen
  - Data Protection
- Vergleich Android
- Häufige Fragen zur Sicherheit
  - Jailbreaking
  - Folgen von Lücken
- Device Management
  - Grundlagen
  - iOS / Android Sicherheitsgewinn
- Fazit

iOS Application Sandboxing

- Jede App kann nur auf eigene Dateien und Einstellungen zugreifen
  - Aber auf Filesystem-Ebene haben alle Apps Zugriffsrechte auf die Dateien
  - Alle Apps laufen unter der selben User ID
  - Bei einem Fehler der Sandbox kann dies ausgenutzt werden
- Apps können nur auf APIs zugreifen
  - Kein direkter Zugriff auf OS Ressourcen
  - Beispiel: App kann auf CFNetwork API zugreifen, um mit der Außenwelt zu kommunizieren, aber kein direkter Zugriff auf Network Interfaces möglich

iOS Application Sandboxing Privacy-Lücke

http://bit.ly/6tdgP3

- Lesender Zugriff von einer App auf Daten außerhalb der App-Verzeichnisse
  - Nicht erlaubt im Store, aber im Gerät technisch nicht unterbunden
  - Einschränkung: In manchen Fällen müssen Dateinamen bekannt sein
  - Demo-App „SpyPhone“ von Nicolas Seriot zeigt mögliche Folgen für Standard-Apps
  - SpyPhone nicht im App Store, aber im Quellcode frei verfügbar für Entwickler
  - Funktionen für Zugriff auf Dateisystem im Binärcode zwar auffindbar, aber Zieldatei und Zweck lassen sich beliebig verschleiern
    - Kaum Schutz gegen versierte Angreifer möglich

iOS Prozess Sicherheit
Sandbox Regeln

- Sandbox wird mit Regeln für jede App erzeugt; sog. Sandboxing Rules
  - Existenz des Mechanismus bedingt nicht die Ausführung der Regeln
  - Regeln werden nur angewendet wenn ein Programm / Prozess mit diesen Regeln eingesperrt wird
  - Regeln werden erst dann auf Kernel-Level ausgeführt
  - Programme können sich selbst einsperren mittels *sandbox_init*
  - Programme können von außen eingesperrt werden mittels *sandbox-exec*
- Einschränkungen unabhängig von POSIX-Rechten und ACLs
  - Sandbox kann nichts zusätzlich ermöglichen, nur weiter einschränken

Android Sandbox
Vergleich iOS

- Berechtigungen im Kernel verankert
  - Default: Deny All ⇨ kein Zugriff auf App-fremde Daten und kein Zugriff auf Gerätefunktionen
  - Erteilen weiterer Rechte über statische *Permissions*
  - Data-Sharing über Provider-Konzept
  - Java-, Native- und Hybrid-Apps gleich behandelt
- Benutzerkonzept
  - Jede App wird mit eigener User ID gestartet
  - Schutzebene: Filesystem-Rechte
  - Prozessisolation durch Kernel

⇨ Stärkere Isolierung der Apps gegenüber iOS

- ⇨ Kann aber auch keine Trojaner verhindern
- ⇨ jedoch max. Potential von Apps definiert
- ⇨ Root-Exploits können dennoch Konzept umgehen

iOS Jailbreaking
Auswirkungen auf die Sicherheit

Fraunhofer SIT

- Aufheben der Sandbox
  - Anwendungen können als *root* außerhalb der Sandbox gestartet werden und unterliegen dann keinen Beschränkungen mehr!
  - Schreib- und Ausführungsrechte auf gesamtem Dateisystem
  - Ausführung von selbst signiertem Code
- Vertrauenswürdigkeit der Tools umstritten
  - Manipulation an Firmware könnte auch Hintertüren einbauen
  - Handhabung nicht völlig gefahrlos: z.B. Würmer durch Standardpasswörter

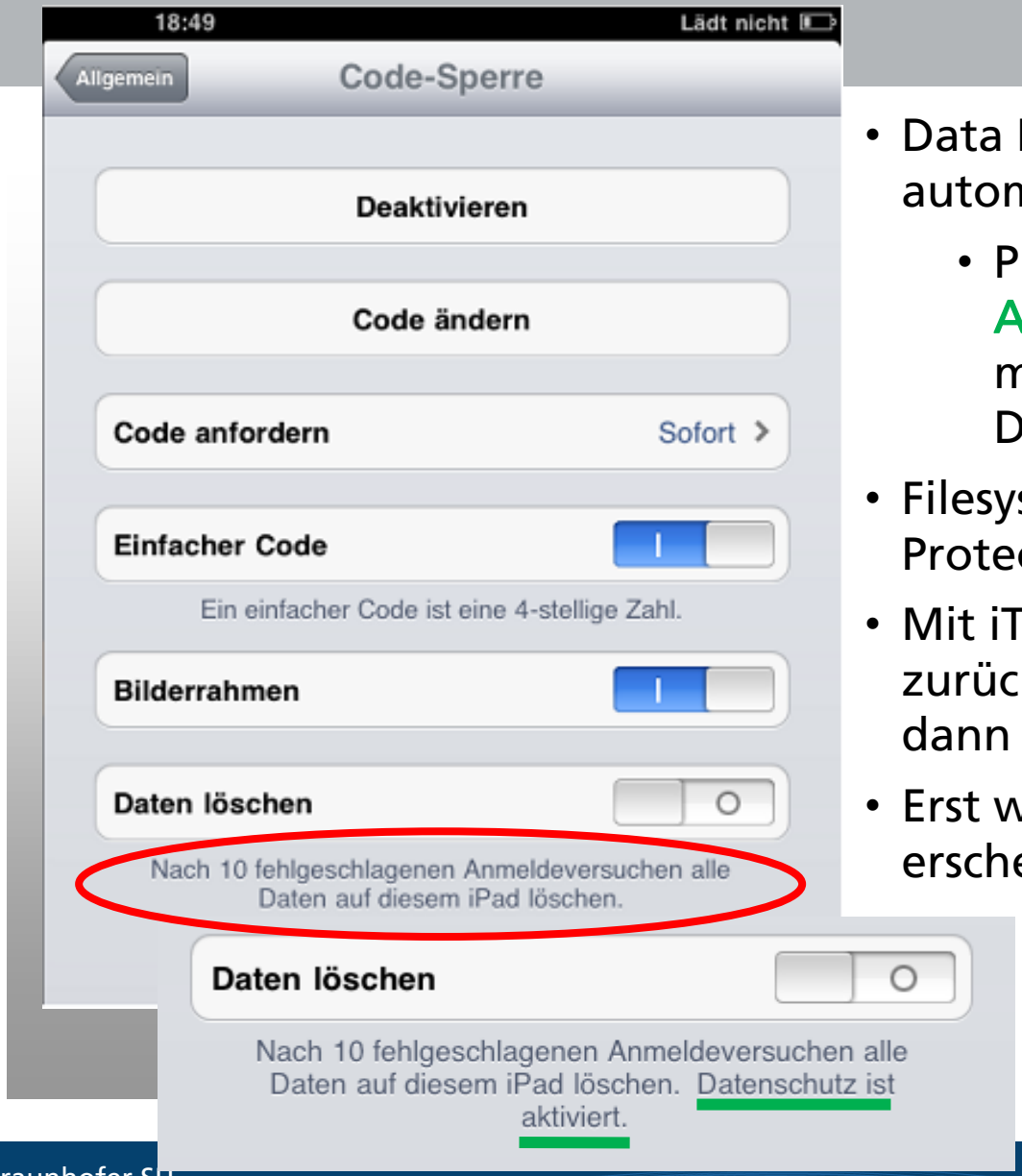
iOS Jailbreaking
Auswirkungen auf die Sicherheit (2)

- Preisgabe von sicherheitsrelevanten Informationen
  - SHSH-Blobs via Cydia (Saurik Server) nicht geschützt
  - Auch „Finder“ eines Gerätes können die SHSH-Blobs dort abrufen
  - Gefährdung durch Jailbreak besteht somit auch für Geräte mit Versionen die gegenwärtig nicht gejailbreakt werden können, deren SHSH-Blogs aber früher vom Nutzer gespeichert wurde
  - Erlaubt Downgrade auf frühere iOS Version, die häufig leichter angreifbar ist
- Verfügbarkeit bei Updates
  - Updates werden häufig bewusst nicht mehr eingespielt, um Jailbreak nicht zu verlieren
  - Geräte bleiben daher oft gegen Bedrohungen ungeschützt bis neuer Jailbreak verfügbar

iOS Data Protection Keybags

- Dateien sind nur bei expliziert Anforderung des Schutzes verschlüsselt
  - Auf Anwendungsebene sind zunächst alle Dateien unverschlüsselt
  - Auf Anforderung einer App wird eine verschlüsselte Datei mit einem individuellen Schlüssel angelegt
- Unterschiedlicher Schutz der Schlüssel
  - Immer verfügbar (default)
  - Verfügbar wenn entsperrt
  - Verfügbar nach dem ersten Entsperren

iOS Data Protection



- Data Protection ist nach Update nicht automatisch aktiv!
  - Prüfen in: **Einstellungen** / **Allgemein** / **Code-Sperre** / (rot markierter Text erwähnt Datenschutz nicht -> **nicht aktiv**)
- Filesystem muss nach Update für Data Protection angepasst werden
- Mit iTunes Auf Fabrikeinstellungen zurücksetzen (**wiederherstellen**), dann Backup zurückspielen
- Erst wenn grün markierter Hinweis erscheint ist Data Protection **aktiv**!

iOS Data Protection
Angriffe auf Keybags

- Backup Keybag
  - Auf Computern gespeichert mit denen Gerät synchronisiert wurde; geschützt mit Backup Passwort
- System Keybag
  - Bruteforce Angriff **nur auf Gerät** ausführbar, wegen Schutz des Device keys
  - Fehlversuchsschutz nur GUI;
    Jailbreak ermöglicht direkten Angriff
  - Bewusste Verlangsamung der Schlüsselgenerierung: ~20 Versuche pro Sekunde auf aktuellem iPhone 4
    - ⇨ 4 stelliger Zahlencode: ca. 8 Minuten
    - ⇨ 6 stelliger Zahlencode: ca. 14 Stunden
    - ⇨ 6 stelliger alphanum. Code ca. 3,5 Jahre;
      aber Wörterbuchangriff beachten!

iOS Physischer Zugriff

- Jailbreaking: *Welche Privilegien hat ein Angreifer nach Anwendung eines Standard-Tools?*
  - Kann das verschlüsselte Dateisystem mounten und hat damit Lese- / Schreib-Zugriff auf alle Dateien
  - Keychain, Keybags und Dateien mit Data Protection sind selbst nochmals verschlüsselt
  - Keychain jedoch teilweise nur verschlüsselt mit Device Key; kann daher auf dem Gerät entschlüsselt werden, dies benötigt aber gegenwärtig noch zusätzliches Knowhow
    - (siehe http://bit.ly/fKi2Nn)

iOS logischer Zugriff

- Sicherheitseskalation nach Ausnutzen einer iOS-Lücke: Zugriff auf Data Protection Files?
  - E-Mails ab iOS 4.2 geschützt mit Data Protection (wenn aktiviert)
  - Wenn Passcode gesetzt, mit diesem im Filesystem verschlüsselt;
  - Aber: auch diese Verschlüsselung ist transparent!
    - Root erlangt ⇨ Filezugriff möglich
    - E-Mail Passwort dann ohnehin auslesbar ⇨ Zugriff auch über E-Mail-Server

⇨ Somit ist der zusätzliche *Data Protection* Schutz eher gegen physischen Zugriff gerichtet.

Was fehlt Android gegenüber iOS für den Unternehmenseinsatz?

Fraunhofer SIT

- Android ≠ Android
  - Kein einheitlicher Funktionsumfang
  - Hersteller- / Versionsabhänige Unterschiede in der Verwaltung
- Dateisystem Verschlüsselung
  - keine direkte OS-Unterstützung (< 3.0 Honeycomb)
  - Sicherer Schlüsselspeicher
- Patch-Management
  - Durch Gerätevielfalt in Hand der Hersteller
  - Kaum Patches; lange Wartezeiten

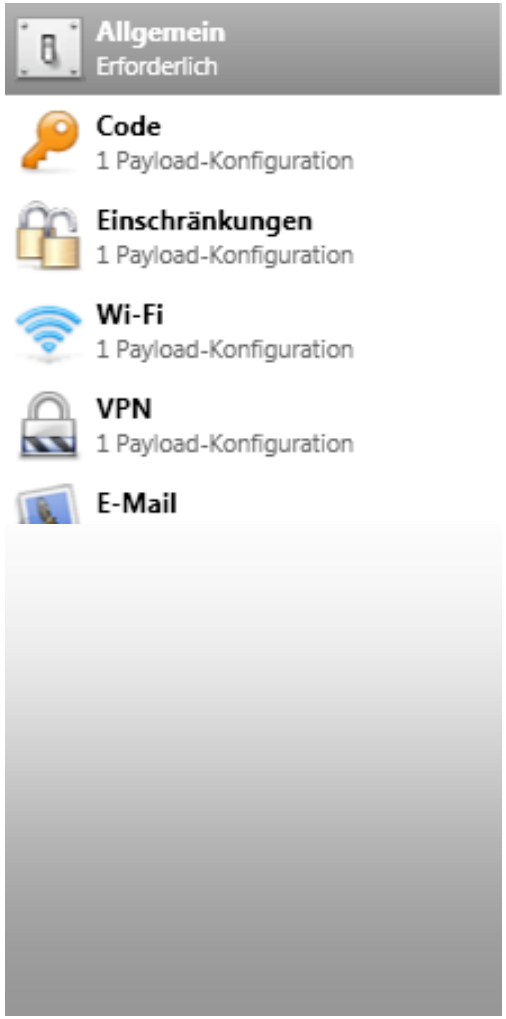
Was fehlt Android gegenüber iOS für den Unternehmenseinsatz? (2)

- Durchsetzbarkeit der Richtlinien
  - Durch Nutzung alternativer E-Mail-Clients bspw. ActiveSync Richtlinien umgehbar
  - Mit anderen Tools ebenso umgehbar, z.B. LockPicker, Extend Lock Time, …
- Fernwartung
  - Nur Basisfunktionen für Ferninstallation
  - Kein Remote Wipe (Herstellerabhängig)

⇨ Noch offen wann „Enterprise Initiativen“ hier Abhilfe schaffen

- [Samsung, Sybase]
- [HTC, Motorola, Pantech, Sharp, Sony Ericsson]

iOS Device Management Schnittstellen



- Logische iOS Verwaltungsschnittstellen
  - Authentication: *Registrierung Nutzer / Gerät*
  - Enrollment: *gemeinsames Geheimnis erzeugen*
  - Device Configuration: *Übermittlung Konfiguration*
- iPhone Configuration Utility
  - Zum Erstellen der Konfigurationen / Richtlinien
  - Zur Verteilung und Verwaltung jedoch zusätzliche Software nötig
  - Einfachster Fall: Webserver liefert manuell .mobileconfig Dateien an Endegräte aus
  - „Bastel-Lösungen“ zur Integration mit IIS und SCEP möglich
  - 3rd Party-Produkte nutzen die selben Schnittstellen, erzeugen aber deutlich mehr Funktionalität

Mobile Device Management
Grundfunktionen 3rd Party Lösungen

- Softwareverteilung
  - Zentrale Bereitstellung von Software
  - Zuordnung zu Endgeräten/-gruppen
  - Automatische Übertragung und Installation
  - Definierbare Übertragungszeitpunkte
  - Wahlmöglichkeit der Übertragung (GPRS,UMTS,WLAN)
- Remote Konfiguration
  - Vollständiger Remote-Zugriff
  - Auslesen von Konfigurationen
  - Konfiguration von Applikationen, Netzwerkinstallation
  - Möglichkeit zur Benachrichtigung

Mobile Device Management
Grundfunktionen 3rd Party Lösungen (2)

- Inventarisierung
  - Hardware
  - Software
  - Verwaltung von Software-Lizenzen
- Backup and Restore
  - Vollständiges Backup / definierbare Einzeldateien
  - Verschlüsselung
  - Intervallsteuerung
  - Wahlmöglichkeit der Übertragung
- Endgerätesicherheit
  - Konfiguration von Passwort/Sicherheitsrichtlinien
  - Fernsperrung/-löschung
  - Patch-Management

Mobile Device Management
Grundfunktionen 3rd Party Lösungen (3)

- Container-Funktionalität
  - Trennung privat / geschäftlich
- Infrastruktur-Kontrolle
  - Nur registrierte und kontrollierte Geräte erhalten Zugriff auf Unternehmensdienste
  - Erkennung von Geräten mit Jailbreak (umstritten)

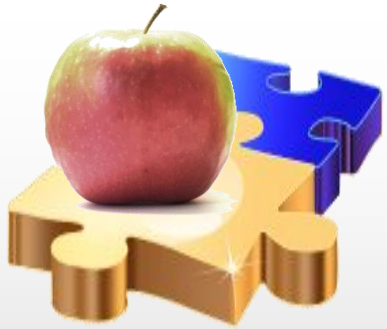
Mobile Device Management
Wirkungsweise 3rd Party Lösungen

Fraunhofer SIT

- Benötigt App auf Endgerät
  - Installation über App-Store / Android Market
  - Dient auch als Enforcement Agent
  - Dupliziert PIM-Anwendungen für Unternehmensumfeld
  - Regelt Zugang zu Unternehmensbereich
- Server-Gegenstelle
  - Fragt Device-Informationen für Registrierung ab
  - Sendet Konfiguration
  - Datenbank über Nutzer, Geräte, Einstellungen
  - Steuert Zugriff auf Unternehmensressourcen (blockiert private / nicht verwaltete Geräte)

Mobile Device Management
Sicherheitsgewinn für iOS

Fraunhofer SIT



- Organisatorisch
  - Vergleichbare Möglichkeiten wie bei Desktop-Systeme: verwalten, kontrollieren, beschränken
  - Sicherheitsgewinn durch effektive Umsetzung der Richtlinien
  - Startpunkt für sichere Prozesse rund um das Lifecycle Management von Endgeräten
- Funktional
  - Management App besitzen selbst nur App-Rechte; kann OS-Schwächen nur bedingt beheben
  - Container-Verschlüsselung und Trennung der Daten bringt Sicherheitsgewinn, ist aber nicht sicher gegen alle Angriffsvektoren
  - Gesamtsicherheit abhängig vom Einsatz und Konfiguration der Lösungen

Mobile Device Management
Sicherheitsgewinn für Android

- Organisatorisch
  - Vergleichbar empfehlenswert wie bei iOS
- Funktional
  - Größerer Gewinn im Vergleich zu iOS, da mehr Funktionen „nachgerüstet“ werden
    - z.B. Backup, Verschlüsselung, Restiktionen, …
  - Nutzer können Restriktionen durch offenere Architektur jedoch leichter umgehen
  - Container-Verschlüsselung: großes Plus gegenüber ungemanagten Geräten und potentiell sicherer als bei iOS durch stärkere Prozess-Isolation
  - Gesamtsicherheit auch hier abhängig vom Einsatz und Konfiguration der Lösungen

Fazit

- iOS und Android holen bei Sicherheitsfunktionen gegenüber BlackBerry weiter auf
  - Android mit sichererer Architektur, aber nutzt noch nicht volles Potential weil noch jünger
- Apple überlässt Unternehmensfunktionen Drittanbietern
- Mobile Device Management erhöht vor allem den Schutz im „normalen“ Betrieb
  - Kontrolle angebracht wie bei PC-Systemen
  - Gegen gezielte Angriffe fehlt Verankerung im OS
- Mobile Endgeräte bieten inzwischen mehr Schutz als viele Desktopsysteme
  - Bei physischem Zugriff dennoch häufig umgehbar

Kontakt

Jens Heider

Rheinstr. 75
D-64295 Darmstadt

E-Mail: jens.heider@sit.fraunhofer.de
Web: http://www.sit.fraunhofer.de
http://testlab.sit.fraunhofer.de