Mobile OS Security Architectures -

Rene Wiesel
SNET Projekt
Technische Universität Berlin
t0pd0g@gmx.de

Zusammenfassung—**Mobile Endgeräte erfreuen sich immer größerer Beliebtheit und werden immer leistungsfähiger. Sie dienen dem Verbraucher als eigene Informationszentrale und werden fast immer mitgeführt. Die mobilen Geräte sind so gut wie immer mit dem mobilen Datennetz des Betreibers oder fremden Netzen (offen WLAN-Netze, Hotspots, etc.) verbunden. Viele benutzen ihr Endgerät für wichtige Transaktionen (Mobile Banking mit Pin/Tan Verfahren) und besitzen darauf sensible Daten aus privatem oder beruflichem Umfeld. Die mobilen Geräte gehen schneller verloren oder werden gestohlen und besitzen standardmäßig keine Anti-Virus oder Firewall Software. Mobile Geräte sind einer größeren potentiellen Bedrohung durch Angriffen ausgesetzt als stationäre Plattformen, da sie durch die eben genannten Gegebenheiten ein attraktives Ziel für Spionage- und Netzwerkattacken darstellen.**
Das Sicherheitsbewusstsein vieler Nutzer ist noch immer sehr gering. Die Passwortabfragen oder Sicherheitswarnungen werden ignoriert oder als störend empfunden. Daher ist ein umfangreiches Sicherheitskonzept ein sehr wichtiger Bestandteil eines mobilen Systems und rückt immer mehr in den Vordergrund.
Der folgende Artikel soll einen Überblick verschaffen über die gängigsten Betriebssysteme und ihre momentanen Sicherheitsmechanismen, die Sicherheit der Daten und des Benutzers gewährleisten sollen.

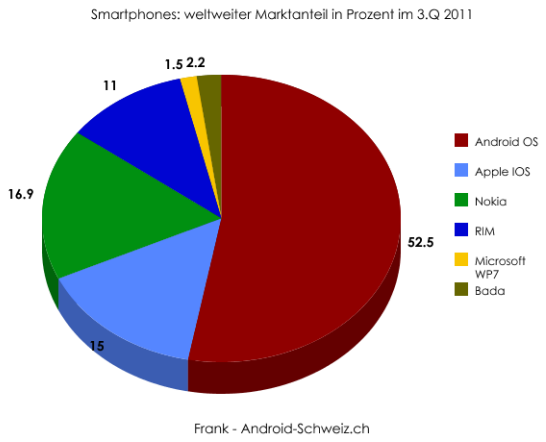
I. Einleitung

Der weltweite Verkauf von mobilen Endgeräten steigt seit den letzten Jahren immer weiter an. Das Marktforschungsinstitut Gartner stellte fest, dass der Verkauf von mobilen Endgeräten um 5.6% und bei Smartphones um 42% im 3. Quartal 2011 anstieg [1].
Mobile Endgeräte werden immer komplexer und ihre Einsatzmöglichkeiten immer vielfältiger. Der Nutzer hat heutzutage die Möglichkeit zeit- und ortsunabhängig auf sensible, personengebundene Daten zuzugreifen (z.B. Email-Konten, Exchange-Zugänge, VPN-Zugänge, etc.).
Deshalb muss bei dem Einsatz von mobilen Endgeräten deren Sicherheit gewährleistet werden. Laut Claudia Eckert [2] gibt es elementare Schutzziele, die jedes IT-System und somit auch jedes Mobile OS erfüllen sollte.

- **Datenintegrität** (engl. Integrity) - Veränderung der Daten nur nach erfolgter Autorisierung
- **Informationsvertraulichkeit** (engl. Confidentiality) - keine unautorisierte Informationsgewinnung möglich (Abhörsicherheit, Zugriffskontrollen, Anonymität, etc.)
- **Authentizität** (engl. Authentication) Benutzer oder Applikation besitzt eine eindeutige , nachprüfbare Identität (z.B. Passworteingabe des Benutzers oder bei einer Applikation eine digitale Signierung)
- **Verfügbarkeit** (engl. Availability) - Gesicherter Zugriff auf Informationen innerhalb einer festgelegten Zeit
- **Verbindlichkeit** (engl. Non-repudiation) - durchgeführte Aktion auf einem System kann einer Quelle zugeordnet werden

II. Mobile Betriebssysteme

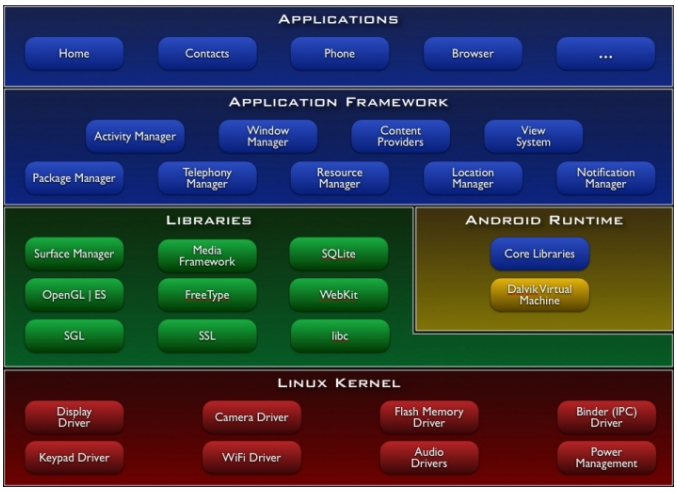
Es gibt verschiedene spezielle Betriebssysteme, die für mobile Endgeräte programmiert wurden. Wir beschränken uns in diesem Artikel auf die 4 größten mobilen Betriebssysteme, die vorrangig auf Smartphones zum Einsatz kommen, obwohl diese natürlich auch in modifizierter Form auf anderen mobilen Endgeräten (PDA, Tablets, etc.) eingesetzt werden können.
In Fig 1. wird deutlich, dass Googles Android mit 52.5% im 3. Quartal 2011 absoluter Marktführer im Bereich Smartphone Betriebssysteme ist. Nokias Symbian belegt mit großem Abstand den 2. Platz mit 16.9%, gefolgt von iOS von Apple (15%) und BlackBerry OS von RIM (11%). Statt das jetzt im Moment noch recht verbreitete Nokia Betriebssystem Symbian näher zu betrachten, widmen wir uns in diesem Artikel dem Windows Phone 7 von Microsoft, da alle neuen Nokia Smartphones mit Windows Phone 7 ausgeliefert werden sollen und es daher in naher Zukunft an Bedeutung gewinnen wird [3].



1. Fig Marktanteile [4]

A. Android

Laut Gartner läuft gut jedes zweite verkaufte Smartphone im 3. Quartal 2011 mit Google-Betriebssystem Android. Android steigerte damit seinen Anteil von 25.3% auf 52.5% im Gegensatz zum Vorjahr. [1]
Wie im Android-Buch von F. Baeumer [5] ausführlich beschrieben, basiert Android auf Linux und übernimmt somit dessen Kernsystem mit Sicherheits-, Netzwerk- und Speichermanagment, sowie die Gerätetreiber. Android ist eine freie Software und quelloffen. Auf diesem Grundsystem aufbauend beinhaltet Android verschiedene Bibliotheken, die das Verarbeiten der speziellen Applikationen und das Anpassen an die spezifischen mobilen Geräte ermöglicht. Die Anwendungen werden in Java geschrieben und werden dann auf einer virtuellen Maschine ausgeführt, die von Google entwickelte Dalvik VM.
Android ist ein sehr flexibles System, dass auf vielen verschiedenen mobilen Geräten einsetzbar und an dessen Hardwarespezifikationen anpassbar ist.
Die Software-Architektur von Android besteht im Wesentlichen aus 5 Komponenten [6]:



2. Fig Android Architektur

- **Applications** - Oberste Schicht beinhaltet neben eigenentwickelten Anwendung die Kernkomponenten (SMS, Kalender, Kontakte, Browser, etc.)

- **Applications Framework** - stellt API dar, die Zugriff auf Hardwarekomponenten aus Android-Anwendungen heraus erlaubt

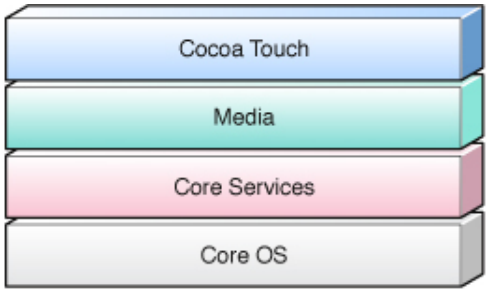
- **Libraries** - C/C# Bibliotheken die erforderliche Funktionalitäten bereit stellen (Datenbanken, Medienbibliotheken, etc)

- **Android Runtime** - beinhaltet wichtige Kernkomponenten, sowie Dalvik Virtual Machine
- **Linux Kernel** - enthält erforderliche Gerätetreiber, sowie Speichermanagement, Sicherheit und Prozessmanagement

Der Applikation Code läuft in einem abgetrennten Bereich, der Sandbox genannt wird. Android, iOS und Windows Phone benutzen das gleiche Prinzip des *application sandboxing*. Die Applicationen haben ihre eigenen, gekapselten Speicherbereich und dürfen nur über vorgegebene Schnittstellen agieren.

B. Appel iOS

Das mobile Apple Betriebssystem findet man nur auf den hauseigenen Apple Geräte, wie iPhone, iPod touch oder iPad. Es ist eine Weiterentwicklung des Mac OS X und die Systemarchitektur besteht aus folgenden Komponenten [7]:



3. Fig iOS Technologie Schichten [7]

- **Core OS:** Der Kernel des Systems

- **Core Services:** Grundlegende Systemservices

- **Media:** High-Level Frameworks zur Darstellung und Wiedergabe von Grafik, Sound und Video

- **Cocoa Touch:** Beinhaltet UIKIT Framework, welches wichtige Funktionen für die Entwicklung von iOS Applikationen beinhaltet

Apples iOS benutzt zwar das gleiche Sandboxing Model wie Android [8], aber bei iOS müssen noch alle Applikationen

und Drittanwendungen von einem offiziellen Apple Zertifikat signiert sein. Es stellt sicher, dass die Applikation nicht manipuliert wurden und es wird bei jeder Ausführung geprüft, ob die Anwendung nicht *untrusted* wurden seit dem letzten Aufruf [9].

C. Windows Phone 7

Gartner sagt voraus, dass durch den Deal mit Nokia Windows Phone bald den 3. Platz im weltweiten Ranking der Smartphone Betriebssysteme einnehmen wird [10]. Windows Phone benutzt das selbe Sandboxing Model wie iOS und Android [10]. Das XNA und Silverlight Framework bilden zusammen mit den Windows Phone spezifischen Komponenten und der Common Base Class Libary das Grundgerüst um eine Anwendung für Windows Phone zu entwickeln [11]:

- **Silverlight Framework** - dient zur Erstellung der Anwendungsinterfaces (Einbettung von Videos, Nutzung von Windows Phone Bedienelementen)
- **XNA Framework** - Software, Dienste und Ressourcen für die Spieleentwicklung
- **Sensors** - geben Daten zurück, die vom Entwickler verwendet werden können (z.B. Kompass, Multi-Touch Eingaben, etc.)
- **Media** - stellt API zur Verfügung, die für die Medienwiedergabe, Grafik und Animationen zuständig ist
- **Data** - isolierter Speicher nach dem Sandboxing Prinzip (ein virtueller Ordner für jede Anwendung)
- **Location** - erlaubt Entwickler den Zugriff auf bestimmte physische Daten des Benutzers

D. BlackBerry OS

BlackBerry OS ist das für Entwickler kostenlos nutzbare Mobile Betriebssystem der Kanadischen Firma Research In Motion (RIM). Mit 17.5 Prozent war BlackBerry OS im Jahre 2010 noch einer der wichtigsten Mobilen Betriebssystem, deren Kundenstamm hautpsächlich im Businessbereich zu finden ist. Gartner sagt aber eine absteigende Tendenz in den nächsten Jahren voraus [10]. Das Betriebssystem benutzt eine etwas andere Form des Sandboxing Models [12], sowie ein Signaturprinzip für die Anwendungen, welches auch die Zugriffsrechte beinhaltet. Dies ist notwendig da die Anwendungen, nicht wie bei Android oder iOS nur Zugriff auf einen bestimmten gekapselten Speicherbereich haben, sondern vollen Zugriff auf das Gerät und deren Speicherbereich.

III. Sicherheitsrichtlinien mobiler Betriebssysteme

A. Integrierte Sicherheitsmechanismen

Die mobilen Betriebssysteme der heutigen Zeit implementieren alle meistens die grundlegenden Sicherheitsmechanismen, welche die oben genannten elementaren Schutzziele gewährleisten sollen [13].
Die in allen Endgeräten verbauten SIM-Karten, die einen eingeschränkten Zugriff und geschütze Bauart aufweist, bieten außerdem eine sichere Möglichkeit, sensible Daten wie Kontaktdaten oder Netz-Zugangsdaten zu speichern. Des weiteren wird versucht die Informationsvertraulichkeit zu gewährleistet, indem man die lokalen Daten oder die Kommunikation verschlüsselt. Das dies aber nicht bei allen mobilen Betriebssystemen der Fall ist wird im folgenden Kapitel deutlich und genauer erläutert.

B. Android

1) Sicherheitsrichtlinien: Android beschreibt in seinen Android Developer Guides [14], welche grundlegenden Sicherheitsstrategien Android verwendet. Die Applikationen laufen isoliert voneinander ab und haben ohne Zustimmung des Benutzers erstmal keine Erlaubnis auf andere Applikationen, System oder Benutzerdaten zuzugreifen oder diese zu ändern. Jede Applikation läuft in einem eigenen Linux Prozess, der jeweils eine eigene Instanz der DVM startet. Weiterhin bekommt jede Anwendung eine eigene Unix User ID (UID), wodurch immer eine genaue Zuordnung im Laufe der Ausführung gegeben ist. Um aber eine Interprozesskommunikation zu ermöglichen, benutzt Android die sogenannten explizit deklarierten Richtlinien. Diese Rechte werden durch den Benutzter zum Installationszeitpunkt festgelegt und bleiben dann bestehen. Android verfolgt dabei ein Konzept aus einer Mischung des Linux-Rechtesystems und eines darauf aufsetzenden eigenen Verfahrens. Dabei sind prinzipiell Zugriffe auf Hardware und auf Inhalte zu unterscheiden. Zum einen werden I/O- und Hardware-Zugriffe aufgrund von Zugehörigkeit zu Benutzergruppen aufgelöst, wohingegen Zugriffe auf Kontakte oder SMS durch einen System-Service realisiert werden. Dieser Service überprüft dann, ob eine Applikation die angeforderten Rechte zugeteilt bekommt [15]. Die Methode der Zertifizierung der Anwendungen wird in Android zwar auch verwendet, aber die Signierung muss nicht von einer offiziellen Autorität durchgeführt werden, sondern liegt in der Hand des Entwicklers.
Zwei Anwendungen können zwar durch ihre unterschiedlichen UIDs nicht im selben Prozess gestartet werden, aber Android bietet noch die Shared User IDs, die es den Applikationen erlaubt, die selbe UID zu beantragen , wenn sie mit dem selben Schlüssel signiert sind [16]. Daraufhin gelten sie für das Betriebssystem als eine Applikation, was ihre Zugriffsrechte angeht. Das kann der Entwickler ausnutzen , um seinen Applikationen mehr Rechte zu verschaffen, als der Benutzer im eigentlich zugestanden hat.
Es wird deutlich, dass durch das Ausnutzen der unterschiedlichen Funktionen der Rechteüberprüfung und der nicht zustimmungspflichtigen Signierung der Applikationen es möglich ist, unberechtigt Informationen eines Geräts auszulesen oder diese an einen entfernten Server zu schicken.

Es können dadurch auch Daten auf dem Gerät ungeachtet vom Benutzer manipuliert und verändert werden.

2) Android Market : Der Android Market ist ein zentrales Softwarearchiv zur Installation von Applikationen. Die meisten mobilen Betriebssysteme benutzen solch ein Zentralarchiv, da es für den Benutzer bequemer ist. Außerdem dient es dem Hersteller zur Kontrolle und verschafft ihm einen Überblick über die Anzahl, Art und Nutzung der Applikationen. Um als Entwickler eine Anwendung über den Android Market zu veröffentlichen, benötigt dieser nur ein Google-Konto und muss via Kreditkarte eine Registrierungsgebühr von 25 Dollar entrichten [17]. Danach muss der Entwickler nur noch sein eigenes Zertifikat erstellen, um danach beliebige Anwendungen in den Android Market einstellen zu können. Eine weitere Überprüfung der Software oder ob der Kreditkarteninhaber wirklich der Entwickler ist, findet nicht statt. Außerdem ist es in Android möglich Anwendungen von alternativen Installationsquellen zu installieren, was weitere Risiken mit sich bringt. Diese Option muss aber erst vom Benutzer im Optionsmenü aktiviert werden.

C. iOS

1) Sicherheitsrichtlinien: Durch das Sandboxing Modell laufen wie bereits erwähnt die Anwendungen isoliert voneinander ab und sind vom System Kernel getrennt. Drittanwendungen haben keine Möglichkeit dies zu umgehen und besitzen alle die selben limitierten Rechte und sind dadurch komplett kontrollierbar durch das iOS und den Benutzer. Die Anwendungen haben keinen Einfluss darauf, welchen Systemstatus sie vom Betriebssystem erhalten und sind mit wenigen Schritten vom Benutzer selbst zu entfernen oder zu schließen [18].
Auf der anderen Seite erlaubt iOS jeder Anwendung auf folgende Ressourcen frei zu zugreifen:

- Adressbuch und Kalendereinträge
- ID des Gerätes
- Telefonnummern
- Musik, Fotos und Videos
- Browser Verlauf
- Kommunikation zu jedem Computer im Wireless Internet und Verbindungs Logs
- Mikrophone und Video Kamera

2) Apple App Store: Apple benutzt auch ein Zentralarchiv zur Verbreitung der Software auf ihren mobilen Endgeräten. Bevor die Entwickler Anwendungen im App Store veröffentlichen können und um die Entwickler Tools benutzen zu können, müssen sie einen Registrierungs Prozess durchlaufen und ein einjährige Lizengebühr von 90 Euro bezahlen.Die Entwickler müssen jede Anwendung mit einem Apple Zertifikat signieren.
Bei Apple wird jede Anwendung einer Prüfung unterzogen. Apple beschreibt in seinem *App Store Review Guideline* [19], welche Kriterien eine Applikation erfüllen sollte, damit sie durch diese Prüfung kommt. Zum Beispiel darf die Anwendung keine lokationsbezogenen Daten speichern und senden, keine eigene Werbung schalten (nur die iAds von Apple), keinen Schaden anrichten oder Pornografie beinhalten. Dadurch kann schon von vornherein potentiell schädliche Software so gut wie ausgeschlossen werden. Besteht die Anwendung die Prüfung und besitzt ein digitales Zertifikat, so kann es im App Store veröffentlicht werden.
Es ist außerdem nicht möglich wie bei Android von alternativen Quellen Software zu installieren. Das schränkt zwar die Freiheit des Benutzers ein, erhöht aber auch die Sicherheit des Systems.

D. BlackBerry OS

1) Sicherheitsrichtlinien: Die BES Richtlinien stellen das Kernkonzept der BlackBerry Sicherheitspolitik dar. BlackBerry besitzt über 400 BES Richtlinien, um die mobilen Endgeräte auch sicher im Businessbereich einsetzen zu können. Diese Richtlinien kann man in folgende Kategorien einteilen [20]:

- **Gruppen-IT-Richtlinien**, ermöglicht die Erstellung und Anpassung von Gruppenrichtlinien, um Datenzugriff und Datensicherheit zu gewährleisten.
- **Standard-IT-Richtlinien**, jedes Smartphone erhält bei seiner Aktivierung vom Administrator anpassbare Basis-IT-Richtlinien, um den Bedürfnissen des jeweiligen Unternehmens gerecht zu werden.
- **Mobile Erzwingung**, über die BlackBerry Enterprise Server werden die vom Administrator erstellten Richtlinien bei allen verknüpften BlackBerry Smartphones automatisch synchronisiert. Dies ist vom Benutzer nicht zu verhindern und stellt sicher, dass alle vorgegebenen Richtlinien des Unternehmens einhalten werden.
- **Malware-Kontrolle**, die BlackBerry Enterprise Solution Server beinhalten 19 Richtlinien zu Anwendungssteuerung, um Schaden durch bösartige Programme (Malware) zu verhindern. Der Administrator des Servers kann damit Anwendungen nur benötigte Ressourcen zugestehen und deren Zugriff auf beispielsweise Benutzerdaten, E-Mail oder Bluetooth verhindern.
- **Umfassende Kontrolle über die gesamte BlackBerry Enterprise Solution**, dadurch kann der Administrator spezielle Funktionen erzwingen, wie: Kennwortnutzung, Einstellung der Besitzerdaten, etc.

BlackBerry verfolgt einen sehr umfangreichen und sicheren Weg, die elementaren Schutzziele zu erreichen und sich speziell im Businessbereich weiter zu etablieren. Das bietet natürlich auch Nachteile:

- Unternehmen an BlackBerry Geräte gebunden
- für normalen Benutzer unbrauchbar

2) BlackBerry App World: Applikationen können unter anderem direkt über Internet URLs installiert werden. Außerdem hat BlackBerry ebenso wie Apple iOS und Google Android einen App Store - die BlackBerry App World. Um

eine Applikation in diesem App Store zu veröffentlichen , muss man sich einen Vendor Account zulegen und diesen mit einem PayPal Account verknüpfen. BlackBerry besitzt ebenso wie Apple iOS eine Validierung der Applikationen. Alle Applikationen müssen die BlackBerry App World Vendor Guidlines [21] erfüllen, um den Freigabeprozess zu bestehen.

E. Windows Phone

1) Sicherheitsrichtlinien: Das neue Windows Phone 7 besitzt weniger Management und Sicherheitsfähigkeiten als noch sein Vorgänger Windows Mobile oder die anderen großen Mobilen OS wie iOS, Android, etc. Windows Phone ist hauptsächlich für den normalen Benutzer konzipiert und unterstützt nur einige Basis Sicherheitsrichtlinien. Es werden aber einige Exchange ActiveSync (EAS) Richtlinien unterstütz, was weitere Sicherheit bietet kann [22]:

- **Password Required** - PIN-Code zum Entsperren und Bildschirmsperre aktiv
- **Minimum Password Length** - legt minimale Passwortlänge fest
- **Idle Timeout Frequency Value** - definiert Zeitspanne, bis Smartphone automatisch gesperrt wird
- **Device Wipe Threshold** - Anzahl erlaubter Fehlversuche bei PIN-Eingabe bis sämtliche Daten und Anwendungen auf dem Gerät gelöscht werden
- **Allow Simple Password** - erlauben oder untersagen von einfachen Passwörtern
- **Password Expiration** - Zeitspanne bis Passwort erneuert werden muss
- **Password History** - verhindert, dass Benutzer dasselbe Passwort setzt

Einige elementare Schutzmechanismen wie die Datenverschlüsselung auf dem Gerät und die Benötigung eines komplexen Passworts sind noch nicht vorhanden. Sie sollen aber laut Microsoft später hinzugefügt werden. Dadurch ist ein Windows Phone für die meisten Unternehmen im Moment unbrauchbar. Dadurch das Windows Phone keine herausnehmbaren Speicher unterstützt, spielt auch die Speicherkartenverschlüsselung in der Windows Phone Ausstattung keine Rolle.

2) Windows Phone Marketplace: Ähnlich wie bei Apple werden nur Applikationen auf dem Marketplace zugelassen und signiert, wenn diese den Zertifizierungsprozess bestehen [23].

F. Vergleich der Betriebssysteme

Um sich einen Überblick zu verschaffen, wird in Tabelle 1 verdeutlicht, welche allgemeinen Hürden ein Entwickler überwinden muss, um eine Anwendung für das jeweilige System zu veröffentlichen.

	iOS 4.x	Android 3.0	Windows Phone 7	RIM Black-Berry 6.x
Registrierung	Ja	Ja	Ja	Ja
Kosten	Ja (90 Euro/Jahr	25 Dollar	Nein	Nein
Entwicklungs-programme	haus-eigene Apple Tools	Java-Entwick-ler-umge-bung	Java-Entwick-lerum-gebung	Win-dows Phone Devel-oper Tools
System	Mac OS X	beliebig	beliebig	Win-dows
Prüfung der Anwendung	Ja	Nein	Ja	Ja
nutzbare End-geräte	nur iDevi-ces	alle Android-Handy	Black-Berry Geräte	alle Win-dows Pho-nes
Autoritäre Signierung	Ja	Nein	Ja	Ja

Tabelle 1. Entwicklerhürden

Bei dem Vergleich der Sicherheitsrichtlinien der einzelnen Systeme wird deutlich, dass BlackBerry und iOS ein deutlich umfangreichere Sicherheitspolitik als Android und Windows Phone verfolgen. Trotzdem sind bei Android und Windows Phone die wichtigen und grundlegenden Mechanismen vorhanden.
Das es bei iOS kaum schädliche Software gibt, folgt auch aus der Schwierigkeit der Hacker eine Anwendung auf iOS zu vertreiben. Durch den umfangreichen Registrierungsprozess mit Identitätsprüfung, der Bezahllizenz, Zertifizierung und Kontrolle der Anwendungen werden viele Hacker abgeschreckt. Versucht der Hacker deshalb geprüfte Anwendungen zu verändern, wird dadurch die digitale Signatur verändert, was diese Möglichkeit auch kompliziert macht.

Android hingegen ist offener gegenüber den Entwicklern und Benutzern. Google fordert Eigenverantwortung. Dadurch ist Android natürlich auf vielen Geräten zu finden, aber auch potenziell unsicherer. Die vielen verschiedene Versionen auf unterschiedlichen Geräten und die hohe Verbreitung verzögern sicherheitsrelevante Updates um einiges.
Einen kleinen Überblick welche Richtlinien von den verschieden mobilen Systemen unterstützt werden gibt Tabelle 2. [24]:

- **On-device encryption/over-the-air data encryption**: Verschlüsselung der Daten auf dem Gerät und bei der Übertragung

- **Complex passwords**: Verwendung eines komplexen Passworts
- **Enforce password policies**: durchsetzen bestimmte Passwort Richtlinien
- **Remote wipe**: Fernlöschung der Daten
- **Remote lockout**: Ausloggen und Passworteingabe erzwingen

Die Abbkürzungen EAS steht für Exchange ActiveSync und BES füer BlackBerry Enterprise Server. Sie stellen externe Software auf Servern da, die zum Nachrichtenaustausch, Verwaltung, Überwachung oder zur Datenverschlüsselung bei mehreren Endgeräten dienen können. Damit kann man detaillierte Sicherheitsrichtlinien für die Geräte vorgeben, die vom Benutzter nicht umgangen werden können.

Policies	iOS 4.x	Android 3.0	Windows Phone 7	RIM Black-Berry 6.x
On-device encryption	yes	yes	no	yes
Over-the-air data encryption	yes	yes	yes	yes
Complex passwords	yes	yes	no	yes
Enforce password policies	yes	yes	EAS	BES
Remote wipe	yes	yes	EAS	BES
Remote lockout	yes	yes	EAS	BES

Tabelle 2. Mobile Sicherheit und Management Möglichkeiten

G. Sicherheitslücken

Im folgenden Kapitel werden einzelne oben genannte Sicherheitsmechanismen genauer behandelt. Es werden ihre Lücken aufgezeigt und an Beispielen verdeutlicht, wie diese genutzt werden können um damit Schaden anzurichten.

1) Android Malware: In den vorigen Kapiteln wurde schon mehrmals erwähnt, dass Google sehr geringe Sicherheitsrichtlinien hinsichtlich der Veröffentlichung von Anwendungen für Android vorgibt. Welche Auswirkungen das haben kann, wurde zwischen 2010 und 2011 deutlich. Es erschien Malware wie Android.Rootcage, Android.Pjapps und Android.Bgserv. Die genannten Anwendungen nutzen alle die gleichen Schwachstellen in Googles Sicherheitspolitik. Sie benutzen ein schon veröffentliche Anwendungen , veränderten dessen Code und erstellen eine neue unzertifizierte Digitale Signierung [18]. Danach veröffentlichten sie diese Anwendung im Android Market oder im Internet.
Im März 2011 tauchte erstmals der Android.Bgserv auf. Der Android.Bgserv ist die veränderte Version das eigens von Google entwickelt und veröffentlichten Security Tool für die Entfernung des Android.Rootcager, der 58 Anwendungen im Android Market infizierte. Der Rootcager extrahierte verschiedene Geräteinformationen und konnte gefährliche Pakete aus dem Internet runter laden und diese ohne das Wissen des Nutzers installieren. Die trojanische Fake Version des Security Tools von Google hingegen schickte Benutzerdaten und die IMEI Nummer des Gerätes zu einem Server in China. Der geringe aber gefährliche Unterschied zum Original wird in folgender Grafik deutlich [25].

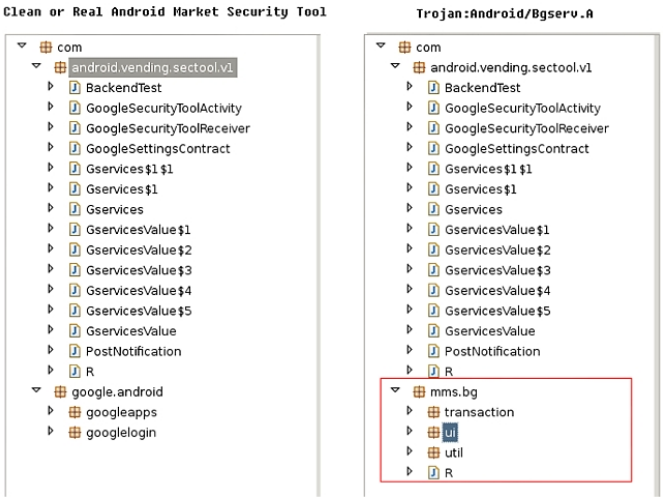


Fig 5. Vergleich Security Tool und Trojaner

2) Apple iOS Jailbreak: Ein Jailbreak auf einem iOS Gerät schaltet die Signierung der Anwendungen aus und ermöglicht auch Anwendungen zu installieren, die nicht von Apple signiert und daher auch nicht im offziellen App Store veröffentlich wurden. Eine Möglichkeit bietet die Anwendung für iOS **Cydia**, welche meist beim Jailbreaken mit installiert wird. Damit kann der Nutzer auch Software von anderen Quellen außer dem App Store auf sein Gerät installieren. Es sollte erwähnt werden, dass ein Jailbreak an sich nicht illegal ist. Der Benutzer ist zwar jetzt in der Lage, dass Endgerät an seine Wünsche anzupassen, umgeht aber dadurch auch das grundlegende Sicherheitsprinzip, auf das iOS aufgebaut ist.
Ein Jailbreak ist aber nur durch eine Sicherheitslücke in der Hardware oder Software möglich. Dadurch wird deutlich, dass auch wenn iOS als sehr sicheres System gilt, es immer Exploits gibt, die eine Gefahr darstellen können. Die Hardware Exploits stellen aber nur ein Risiko dar, wenn der Angreifer im Besitzt des Gerätes ist. Bei einer Softwarelücke kann der Hersteller diese durch einen Patch wieder schließen. Ein gutes Beispiel für Software Exploits is die 3. Version von *jailbreakme* [26]. Der Hacker nutze zwei Exploits um via einer PDF Datei, die im Safari Browser (bis iOS Version 4.3.3) geöffnet werden konnte, dass Apple Gerät zu jailbreaken. Eine Sicherheitslücke im *Freetype Type 1 font parser* ermöglichte

dann die Ausführung von Code, der daraufhin eine Lücke im Kernel nutzte, um die Codesignierung zu umgehen, Rootrechte zu erhalten und den Jailbreak zu installieren [27]. Diese Sicherheitslücke stellt natürlich eine große Gefahr dar und wurde auch kurz nach der Bekanntgabe durch einen Patch wieder geschlossen. Es ermöglichte aber auch für kurze Zeit dem Benutzter ohne spezielle Software und genauer Kenntnisse sein iDevice einfach via ***swipe*** [26] im Browser zu jailbreaken.

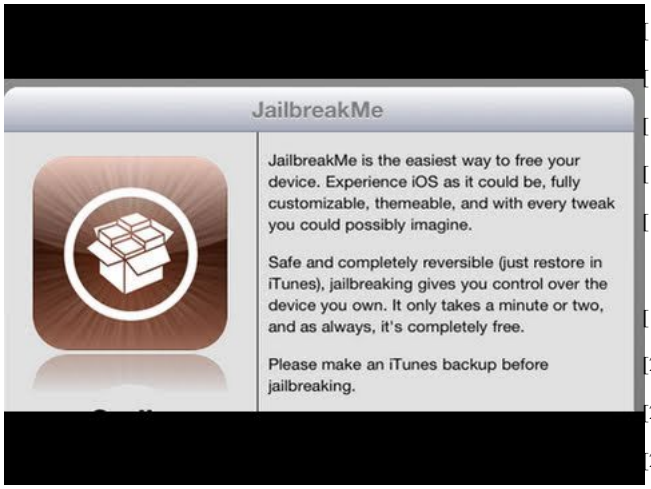


Fig 6. Jailbreak bis iOS 4.3.4 via PDF exploit [26]

H. Schlussfolgerung

Viele der Sicherheitsfeatures, die bei Laptops schon eingesetzt werden, sind auf Smartphones noch nicht vorhanden. Dazu gehören: Virenscanner, Firewalls und vor allem ein fehlendes durchgehendes und wirksames Konzept zur Umsetzung von Datenverschlüsselung. Die mobilen Betriebssysteme sind zwar von Grund auf sicherer konzipiert als normale Betriebssysteme, bieten aber noch zu viele potentielle Angriffsmöglichkeiten. Dieser Problematik Herr zu werden ist eine schwierige Aufgabe, da die mobilen Geräte gleichzeitg sicher, aber dennoch für den normalen Benutzer verständlich und bequem nutzbar sein müssen. Das größte Problem stellt immer noch der Benutzer selbst dar. Sind die Sicherheitsrichtlinien zu streng und bietet dem Nutzer kaum Anpassungsmöglichkeiten, findet dieser alternative Wege (z.B.: Jailbreak, alternativen Quellen zur Installation von Anwendungen, etc.).

LITERATUR

[1] Gartner, "Gartner says sales of mobile devices grew 5.6 percent in third quarter of 2011; smartphone sales increased 42 percent," p. 1, nov 2011. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1848514

[2] C. Eckert, *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. Oldenbourg Wissenschaftsverlag, 2009.

[3] Golem, "Nokia setzt auf windows phone 7," Feb 2011. [Online]. Available: http://www.golem.de/1102/81342.html

[4] Frank, "Android verdoppelt marktanteil in einem jahr auf jetzt 52,52011. [Online]. Available: http://www.android-schweiz.ch/2011-11-17/ android-verdoppelt-marktanteil-in-einem-jahr-auf-jetzt-525/

[5] F. Baeumer, *Android - Grundlagen, Anwendungsentwicklung und moegliche Verdienstformen*. GRIN, 2011, vol. 1.

[6] Google, *Android developers*. [Online]. Available: http://developer.android.com/guide/basics/what-is-android.html

[7] Apple, "ios overview," http://developer.apple.com/library/ios/navigation/.

[8] ——, "The ios environment," http://developer.apple.com/library/ios/documentation/ipho

[9] ——, "ios security," http://images.apple.com/iphone/business/docs/iOS$_S ecurity.pdf$.

[10] Gartner, "Gartner says android to command nearly half of worldwide smartphone operating system market by year-end 2012," April 2011. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1622614

[11] Microsoft, *Security for Windows Phone*, December 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/ff402533(v=vs.92).aspx

[12] BlackBerry, "Blackberry security," 2012. [Online]. Available: http://uk.blackberry.com/ataglance/security/

[13] B. fr Sicherheit in der Informationstechnik, "Mobile engeraete und mobile applikationen: Sicherheitsgefaerdungen und schutzmanahmen," 2009.

[14] Google, *Security and Permissions*, 2011. [Online]. Available: Security and Permissions

[15] D. Bumeyer, "Android sicherheit," 2010. [Online]. Available: http://www1.hgi.rub.de/spring/content/spring6$_1 2_a bstract_b ussmeyer.pdf$

[16] Google, *Android Security*, 2011. [Online]. Available: http://developer.android.com/guide/topics/security/security.html

[17] ——, "Registrierung," http://support.google.com/androidmarket/developer/bin/answer.py?hl=de&answer=113468, 2011.

[18] Symantec, "A window into mobile device security," 2011. [Online]. Available: http://www.symantec.com/content/en/us/about/media/pdfs/symc$_m obile_d evice_s ecurity_j une 2011.pdf?om_e xt_c id = biz_s ocmed_t witter_f acebook_m arketwire_l inkedin_2 011_J un_w orldwide_m obilesecurity$

[19] Apple, *App Store Review Guidline*, 2011. [Online]. Available: http://developer.apple.com/appstore/guidelines.html

[20] BlackBerry, *BlackBerry Security*. [Online]. Available: http://de.blackberry.com/ataglance/security/it$_p olicy.jsp$

[21] ——, "Guidelines," 2011. [Online]. Available: https://appworld.blackberry.com/isvportal/guidelines.do

[22] Microsoft, "Exchange activesync considerations when using windows phone 7 clients," 2011. [Online]. Available: http://social.technet.microsoft.com/wiki/contents/articles/exchange-activesync-considerations-when-using-windows-phone-7-clients.aspx

[23] ——, "Application certification requirements for windows phone," 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/hh184843(v=VS.92).aspx

[24] G. Gruman, "Mobile management: How iphone, android, windows phone 7, and the rest stack up," 2010. [Online]. Available: http://www.infoworld.com/d/mobilize/mobile-management-how-iphone-android-windows-phone-7-and-the-rest-stack-184?page=0,2

[25] F-Secure, "Trojan: Android/bgserrv.a," http://www.f-secure.com/weblog/archives/00002116.html, March 2011.

[26] J. F. comex, "Jailbreakme," http://www.jailbreakme.com/, 2011.

[27] S. E. Lab, "Analysis of the jailbreakme v3 font exploit," http://esec-lab.sogeti.com/post/Analysis-of-the-jailbreakme-v3-font-exploit, July 2011.