Fachstudie

Version 1.0

Thomas Ferber, Paul Jansa, Johannes Dilli

erstellt am: 30. Juni 2011
letzte Änderung: 5. Februar 2012

Inhaltsverzeichnis

1 Einleitung

1.1 Zweck des Dokuments

Das Dokument dient zur Durchführung der Fachstudie „Sicherheit mobiler Endgeräte“ und dokumentiert deren Fortschritt und erarbeitete Ergebnisse. Es dient den Studenten gegenüber dem Institut als Nachweis der erbrachten Leistung.

1.2 Ausschreibung

1.2.1 Privacy & Security – Wie machen das Apple, Google und Co.?

Moderne Smartphones, wie beispielsweise das iPhone 4 oder das NexusOne, werden schon lange nicht mehr nur zum Telefonieren verwendet. Dank der darin verbauten hochperformanten CPU und einem, durch hochauflösende und große Touchscreens ermöglichten, anwenderfreundlichen Eingabekonzept, dienen sie den Nutzern als mobile Computer für die Hosentasche. So lassen sich mit ihnen E-Mails verfassen und versenden, Termine verwalten, Navigationsdienste nutzen, Musik hören oder Fotos aufnehmen, um nur einige Beispiele zu nennen.

Bedingt durch immer günstiger werdende Mobilfunkverträge, sind die Nutzer nicht nur „Always-On“ (d.h. die Geräte sind dauerhaft eingeschaltet), sondern auch „Always-Online“(d.h. es besteht eine dauerhafte Internetverbindung). Zusätzlich beinhalten diese mobilen Geräte eine Vielzahl an weiteren technischen Features, wie GPS-Sensoren zur metergenauen Ortung der Nutzer, Bluetooth zur schnellen und einfachen Verbindung von diversen Geräten oder einer Kamera, mit der sowohl Fotos als auch Videos aufgezeichnet werden können. Dadurch werden sehr viele verschiedene Kontextdaten bestimmt und gespeichert, aus deren Kombination sich weitere Rückschlüsse über die aktuelle Situation des Nutzers ableiten lassen.

Dies kann für den Nutzer sehr hilfreich und angenehm sein – man denke beispielsweise an eine App, die auf Grund der aktuellen Situation eines Nutzers selbstständig entscheidet, wie mit eingehenden Anrufen im Augenblick umzugehen ist (Beispiel: Während einer Konferenz soll nur der Vibrationsalarm aktiviert werden, sonst soll ein normaler Klingelton genutzt werden). Allerdings kann bösartige Software die Kontrolle über diese Informationen ergreifen und dadurch einen Nutzer ausspionieren. Immer wieder tauchen Berichte auf, dass diese hochsensiblen Daten in die Hände von Unbefugten gefallen sind. Auch lässt sich ein sprunghafter Anstieg von Virensoftware für diese mobilen Plattformen feststellen. Daher verwundert es kaum, dass laut aktuellen Umfragen 98% aller Smartphone-Nutzer es für unumgänglich erachten, dass sie volle Transparenz über die Verwendung ihrer persönlichen Daten erhalten.

Ziel dieser Fachstudie soll es sein, zunächst einen umfassenden Überblick über die möglichen Techniken zur Sicherstellung der Privatsphäre und der Sicherheit für mobile Geräte zu liefern. Zur Demonstration des praktischen Nutzens dieser Techniken sollen jeweils Beispiele gefunden werden, wer diese Technik zu dem jeweiligen Zweck einsetzt. Auch muss der Nutzen dieser Technik kritisch bewertet werden.

In einem zweiten Schritt sollen neben diesen Beispielen aus der Praxis auch aktuelle Forschungsergebnisse zu diesem Thema gesucht und analysiert werden. Für beide Aufgabenbereiche sollen auch Implementierungsaspekte berücksichtigt werden, d.h. welche Auswirkungen die jeweilige Technik auf die Programmierung beziehungsweise Nutzung von Apps hat oder welche Funktionen dabei zur Verfügung stehen müssen.

In einem dritten Schritt sollen unterschiedliche Bewertungskriterien für diese Sicherheitsmaßnahmen zusammengestellt und der Nutzen dieser Kriterien im Umfeld von mobiler Privacy & Security ermittelt werden. Abschließend sollen die ermittelten Techniken anhand des Kriterienkatalogs erneut bewertet werden.

1.3 Beteiligte

An dieser Fachstudie sind die im Folgenden genannten Personen beteiligt.

1.3.1 Mitarbeiter des IPVS

Name	Prof. Dr. Bernhard Mitschang
Funktion	Prüfer
E-Mail	Bernhard.Mitschang@ipvs.uni-stuttgart.de
Raum	2.357

Name	Dipl.-Inf. Christoph Stach
Funktion	Betreuer
E-Mail	Christoph.Stach@ipvs.uni-stuttgart.de
Raum	2.360

1.3.2 Autoren

Name	Thomas Ferber
Mat.-Nr.	2349138
E-Mail	ferberts@studi.informatik.uni-stuttgart.de
Studiengang	SWT im 9. Semester

Name	Paul Jansa
Mat.-Nr.	2206417
E-Mail	jansapl@studi.informatik.uni-stuttgart.de
Studiengang	SWT im 8. Semester

Name	Johannes Dilli
Mat.-Nr.	2416294
E-Mail	dillijs@studi.informatik.uni-stuttgart.de
Studiengang	SWT im 8. Semester

1.4 Durchführung

Das Dokument entsteht im Rahmen einer Fachstudie im Studiengang Softwaretechnik an der Universität Stuttgart. Der vorgesehene Umsetzungszeitraum beträgt 6 Monate. Die Fachstudie wird zur besseren Orientierung für die Teilnehmer in Meilensteine unterteilt. Dies gewährt dem Betreuer Einsicht in den Fortschritt des Projektes. Die Fachstudie endet mit einer Abschlusspräsentation und der Ausgabe eines nicht benoteten Scheins.

1.4.1 Meilensteine

Die Meilensteine orientieren sich an den Kapiteln des Dokumentes. Die Reihenfolge der Bearbeitung wurde so gewählt, dass ein sinnvoller Aufbau der Arbeit gewährleistet ist. Die angegebenen Daten dienen zur groben Orientierung bei der Durchführung der Arbeit.

Datum	Meilenstein
August 11	Kapitel 1, Einleitung, Gliederung und Meilensteine
August 11	Kapitel 3, Vorstellung der Plattformen
September 11	Kapitel 4, Security
November 11	Kapitel 5, Privacy
Dezember 11	Kapitel 6, Bewertung
Dezember 11	Kapitel 7, Fazit und Ausblick
Januar 11	Kapitel 8, Sicherheitstipps, Korrektur
Februar 12	Abgabe, Präsentation

1.4.2 Geplanter Umfang

Der angestrebte Gesamtumfang der Fachstudie liegt zwischen fünfzig und sechzig Seiten. Der Umfang der einzelnen Kapitel orientiert sich am Schwerpunkt der Fachstudie. Die Themen Security und Privacy erhalten hierbei am meisten Beachtung und bilden den Kern des Dokuments.

1.5 Themenüberblick

Smartphones stellen einen neuen Trend in der IT-Branche dar, der aufgrund seiner Marktdurchdringung kaum mehr wegzudenken ist. Laut einer Studie vom Januar 2011 [36] besitzt bereits jeder vierte Deutsche ein Smartphone, Tendenz steigend. Somit wird der Smartphonemarkt auch für Hacker zunehmend interessant. Dies führt dazu, dass Smartphones allmählich denselben Gefahrenquellen unterliegen, die es bei Desktop-Computern bereits seit Jahren gibt. Durch den hohen Anteil an Peripherie wie GPS oder Mikrofon ergeben sich zudem neue Gefahrenquellen. Nicht zuletzt, da die Geräte immer dabei und „always-on“ sind, sofern der Akku es mitmacht.

Sicherheit von Netzwerken ist seit langer Zeit ein Thema in der IT-Branche. Wie lassen sich Netzwerke von außen sichern? Wer hat Zugriff auf die verwalteten Daten? Diese Fragen wurden von IT-Unternehmen umfassend durch Software, Hardware und Dienstleistungen beantwortet. Doch was passiert mit der Sicherheit, wenn Nutzer Ihre Daten plötzlich über eine neue Art von Minicomputern, so genannter Smartphones, nach außen tragen? Was wenn Nutzer plötzlich neue Zugangspunkte oder Netzwerke eröffnen und somit Hackern neue Angriffsmöglichkeiten bieten, womöglich ohne es zu wissen? Was passiert mit den Informationen auf dem eigenen Smartphone, wer hat Zugriff auf diese Daten und betreiben die betroffenen Instanzen Datenschutz oder Datenhandel?

Mit diesen Fragen und der zugehörigen technologischen Hardware beschäftigt sich das vorliegende Dokument. Es beschäftigt sich mit der Frage, ob Nutzer eines Smartphones, beziehungsweise einer bestimmten mobilen Hardwareplattform sicher sind, und was mit Ihren Daten alles geschieht und geschehen kann. Nicht zuletzt wird auch aufgezeigt, was die Hersteller alles für Sicherheit und Datenschutz tun und was der Nutzer selbst zu seinem eigenen Schutz tun kann.

1.6 Definition Smartphone

Das vorliegende Dokument bezieht sich auf Betriebssysteme moderner Smartphones. Als Smartphone bezeichnet man moderne Mobiltelefone mit einem Touchscreen zur Eingabe und vielfältigen multimedialen Fähigkeiten. Heutige Smartphones vereinen die Funktionalitäten mehrerer Geräte in einem, darunter Handy, Organizer, Navigationsgerät, MP3-Player, Spielekonsole sowie Tablet-PC. Im Vergleich bieten Smartphones gegenüber älteren Mobilfunkgeräten die folgenden Funktionen und Vorteile:

Früher	**Heute**
Hardwaretasten: Große Tasten, kleines Display	Touchscreen: Mehr Nutzfläche durch Verschmelzung von Display und Eingabe
WAP für langsame Internetverbindungen	Mobiles Internet: Breitbandzugang mit integriertem Web-Browser
Einfacher Kalender	Organizer-Funktionalität
Externe Antenne	Integrierte Antenne
Sperriges Design auf Grund von Hardwarerestriktionen	Flaches, kompaktes Design
Fester Funktionsumfang	Installation von Zusatzfunktionalität durch Apps
Telefonie im Vordergrund	Multimediafunktionalität, audiovisuelle Inhalte

Tabelle 1.1: Smartphones

Smartphones werden zudem mit einem vollwertigen mobilen Betriebssystem betrieben. Diese bieten, analog zu Desktop-Betriebssystemen, neue Stärken und Schwächen gegenüber simpleren Mobilgerätesystemen. Ziel dieser Fachstudie ist es, diese Schwächen in Hinblick auf Sicherheit (Kapitel 4, Security) und Datenschutz (Kapitel 5, Privacy) zu untersuchen.
Die folgenden fünf Betriebssysteme sind in diese Kategorie einzuordnen:

- Android von Google
- iOS von Apple
- Windows Phone 7 von Microsoft
- webOS von HP
- Blackberry OS von RIM

Die gelisteten Systeme werden in Kapitel 3, Vorstellung der Plattformen vorgestellt.

2 Abbildungsverzeichnis

3 Vorstellung der Plattformen

3.1 Android

Android ist ein Betriebssystem für Smartphones, Mobiltelefone und Tablets das von der Open Handset Alliance, die 2007 von Google gegründet wurde, entwickelt wird. Das erste Gerät mit Android erschien am 22. Oktober 2008. Android ist für die Bedienung per Touchscreen ausgelegt.[66]

Hersteller von Geräten dürfen Android nach ihren Wünschen anpassen, so dass eine Vielzahl an verschiedenen Varianten vorhanden sind. Meist unterscheiden sich diese durch eine veränderte Benutzeroberfläche und zusätzliche Programme und Dienste.

Seit der Veröffentlichung von Android 1.0 wurde Android stetig weiterentwickelt. Die aktuelle Version für Smartphones ist 2.3 (Gingerbread) und für Tablet PCs 3.0 (Honeycomb). Während der Entstehung dieser Fachstudie wurde die Version 4.0 (Ice Cream Sandwich) vorgestellt, die die Entwicklungszweige für Smartphones, Tablets und Google TV vereinigt.

3.1.1 Software

Android verwendet den Linuxkernel und weitere Software aus dem Linuxumfeld. Android ist Open Source und steht weitestgehend unter der „Apache Software License, 2.0“. [1]

Android hat eine sehr enge Bindung zu Google. So war es bis April 2009 nicht möglich, ein Android-Gerät ohne Googlekonto zu aktivieren. Jedoch wird für die Nutzung der meisten Googledienste wie Google Mail, Google Calendar und Android Market zwingend ein Google-Konto vorausgesetzt.

Um Apps auszuliefern, stellt Google den Android Market unter `https://market.android.com/` zur Verfügung. Die angebotenen Programme stammen fast ausschließlich von Drittunternehmen und freien Programmierern. Zur Zeit gibt es im Android Market ungefähr 580 000 Anwendungen, die zusammen mehr als 10 Milliarden mal heruntergeladen wurden (Stand Dezember 2011, siehe [10]). Um Apps im Market veröffentlichen zu können, ist ein Entwickler Account nötig, für den Google eine einmalige Registrierungsgebühr von 25 $ erhebt. Die Preise für Apps können von Entwicklern frei festgelegt werden, Google verlangt eine Transaktionsgebühr in Höhe von 30% des Verkaufspreises. Bezahlt werden kann über das von Google angebotene Bezahlsystem „Checkout“, das eine Kredit- oder Debitkarte voraussetzt. In den USA können Kunden von T-Mobile oder AT&T auch per Handyrechnung bezahlen. Eine Bezahlung per „Paypal“ ist in Planung. Seit dem 2. Februar 2011 ist es möglich, Apps direkt von der Webseite des Android Markets auf einem Gerät zu installieren. Hierbei wird die App automatisch an das Gerät gesendet und installiert.

Eine Überprüfung der Apps durch Google vor Veröffentlichung findet nicht statt, jedoch hat Google die Möglichkeit, bereits installierte Apps wieder aus dem Market zu nehmen und, sofern dies notwendig ist, diese von Endgeräten zu löschen. Es ist zudem möglich, Apps als .apk Pakete aus dem Internet zu laden und direkt auf dem Smartphone zu installieren. Somit können auch alternative Quellen, parallel zum Android Market, genutzt werden. Diese bieten häufig andere Zahlungsmöglichkeiten als der offizielle Android Market, haben jedoch nicht die gleiche Vielfalt an Apps.

Verschiedene Dienste können sich als Konto in Android integrieren. Damit übernimmt Android das automatische Synchronisieren der Daten und ermöglicht, Kontakte direkt in das Telefonbuch aufzunehmen. Skype

und Facebook bieten zum Beispiel an, Kontakte zu Synchronisieren, so dass sich diese im Adressbuch des Smartphones wiederfinden.

Ein Androidsystem im Auslieferungszustand liefert folgende Apps mit:

- Browser
- Downloads
- Taschenrechner
- Kamera
- Wecker
- Telefonbuch
- Email
- Fotogalerie
- Nachrichten (SMS und MMS)
- Musik
- Telefon
- Suche
- Einstellungen

Zusätzlich sind auf den meisten Geräten folgende Google Apps enthalten:

- Google Bücher
- Google Mail
- Maps (mit Places und Navigation)
- Market
- News & Wetter
- Kalender

Viele Gerätehersteller liefern noch eigens entwickelte Apps mit, diese unterscheiden sich jedoch von Gerät zu Gerät.

3.1.2 Hardware

In Androidgeräten sind normalerweise ARM-Prozessoren verbaut, es gibt jedoch auch Portierungen auf die x86-Architektur. Es werden die ARM Architekturen v6 und v7 unterstützt. Mit ARMv7 sind auch Dual-Core Modelle möglich.

Androidgeräte werden für gewöhnlich über einen Touchscreen bedient. Zusätzlich gibt es mindestens 3 Tasten: „Home“, „Menü“ und „Zurück“. Weiter Tasten, wie zum Beispiel „Suchen“, „Anruf annehmen“, „Auflegen“, „Lauter“, „Leiser“, sind optional und je nach Hersteller vorhanden. Die Funktionen sind auch ohne Tasten über die Bedienoberfläche von Android erreichbar. Mit „Home“ gelangt man von einer App zurück zu den Startbildschirmen, von denen meistens 3 bis 5 vorhanden sind und jeweils den Bildschirm füllen. Auf diesen kann man Verknüpfungen zu Apps anlegen oder Widgets anzeigen lassen. Mit der Androidversion 3.0 (nur für Tablets) und Android 4.0, wurden virtuelle Tasten eingeführt, die dauerhaft auf dem Touchscreen angezeigt werden, so dass auf physische Tasten verzichtet werden kann.

Androidgeräte können Verbindungen per Mobilfunk (GSM + EDGE, CDMA, UMTS + HSDPA, LTE), WLAN 802.11 b/g/n, Bluetooth 2.1 und Near Field Communication aufnehmen und unterstützen folgende Sensoren:

- Beschleunigungssensoren
- Magnetfeldsensoren
- Temperatursensoren
- Annäherungssensoren
- Drucksensoren
- Lichtsensoren
- Neigungssensoren (Gyroskope)

Bilder und Videos können über eine oder zwei Kameras aufgenommen werden, zur Beleuchtung wird ein Blitz unterstützt. Alle Komponenten zur Verbindungsaufnahme, sowie Sensoren und Kameras, sind optional, die Entscheidung diese einzubauen liegt beim Hersteller des Endgerätes.

Android unterstützt den Anschluss von Geräten per USB oder Bluetooth. So kann beispielsweise eine Tastatur oder Maus zur Steuerung des Gerätes angeschlossen werden.

3.1.3 Entwicklung

Für die Entwicklung von Apps wird das Android SDK benötigt. Dieses stellt die Schnittstellen von Android für Apps zur Verfügung. Android Programme werden für gewöhnlich in Java entwickelt und auf dem Gerät mit Hilfe der „Dalvik Virtual Machine“ ausgeführt. Zusätzlich können Apps in C und C++ mit Hilfe des NDK (Native Development Kit) entwickelt werden, so dass diese als native Anwendungen ohne virtuelle Maschine ausgeführt werden können.

Um Programme einfach testen zu können, liefert Google mit dem Android SDK einen Emulator (Abbildung 3.1) mit, um virtuelle Androidgeräte zu simulieren. Somit wird für die Entwicklung von Apps nicht zwingend ein Androidgerät benötigt.

Als Entwicklungsumgebung empfiehlt Google Eclipse (Abbildung 3.2), wofür ein eigenes Plugin existiert, das die Entwicklung von Android-Apps erleichtert. Unter Anderem bietet es nach dem Kompilieren der App an, diese direkt auf dem Emulator oder einem, an den Computer angeschlossenen, Gerät zu installieren und zu starten. Des Weiteren gibt es einen Editor, der es erleichtert, graphische Oberflächen zu erstellen. Außerdem können viele Grundeinstellungen über eine einfache Oberfläche vorgenommen werden, anstatt diese direkt in eine XML-Datei zuschreiben.

[20]

3.2 Apple iOS

Apple stieg am 29. Juni 2007 mit dem Verkauf der ersten iPhone Generation in das Smartphone Geschäft ein. Als Innovation wurde das kapazitive Display vorgestellt, welches eine einfache Bedienung über Multitouch-Gesten ermöglicht, sowie die Möglichkeit, Zusatzsoftware in Form von Apps auf das Handy zu laden. Nicht zuletzt überzeugte das iPhone durch sein simples aber elegantes Design.

Seither wurde das iPhone mehrere Male überholt und befindet sich derzeit in der vierten Hardwaregeneration. Parallel hierzu stellte Apple den neuen iPod Touch vor, welcher ebenfalls auf der Hardware des iPhone basierte und sich des selben innovativen Displays bediente, jedoch ohne Telefoniefunktionen. Als weitere Nische entdeckte Apple den Tablet Bereich mit dem iPad für sich. Das iPad ist vergleichbar mit dem iPhone, verfügt jedoch über größere Abmessungen und befindet sich derzeit in der zweiten Hardwaregeneration.

Abbildung 3.1: Android Emulator

3.2.1 Software

iOS, früher iPhone OS, stellt das Betriebssystem aller mobilen Post-PC-Geräte von Apple dar. Es wird auf dem iPhone, dem iPod Touch, dem iPad sowie der neuesten Generation des Apple TV eingesetzt. Das System wird von Apple kontinuierlich weiterentwickelt und befindet sich derzeit in der Version 4.3.4. Version 5.0 wird derzeit entwickelt und als Betaversion angeboten. Sie soll unter anderem die Anbindung an Apples neuen Synchronisationsdienst iCloud ermöglichen, welcher sich zurzeit im Aufbau befindet, sowie einen kostenlosen Nachrichtendienst iMessage einführen. Des Weiteren bietet es eine Anbindung an den iTunes Store, was ein Hauptgrund von Apple für die Entwicklung eines eigenen Smartphones darstellt, sowie den darin enthaltenen App Store, welcher die Erweiterung des Systems durch so genannte Apps ermöglicht. Apple wollte sich die Namen „App“ und „App Store“ markenrechtlich sichern lassen und hat bereits mehrere Klagen gegen Mitbewerber ausgesprochen, die ebenfalls einen eigenen Plattform mit dem Namen „App Store“ errichten wollten, unter anderem gegen das Shoppingportal Amazon.[18]

Technisch gesehen stellt iOS ein portiertes und an den verwendeten ARM-Prozessor angepasstes Derivat von Mac OS X dar. Die Realisierung von Software erfolgt als Apps, einfache Applikationen die über das Cocoa Touch Framework auf die Funktionen des jeweiligen Gerätes zugreifen können. Funktionen für Telefonie, Surfen, Kalenderverwaltung sowie Senden und Empfangen von Kurznachrichten und E-Mails werden über mitgelieferte Standardapps realisiert. Diese können hierbei wie alle anderen Apps aktualisiert, jedoch nicht vom System entfernt werden. Ebenfalls wichtiger Bestandteil des Basissystems sind Apps für die Simulation

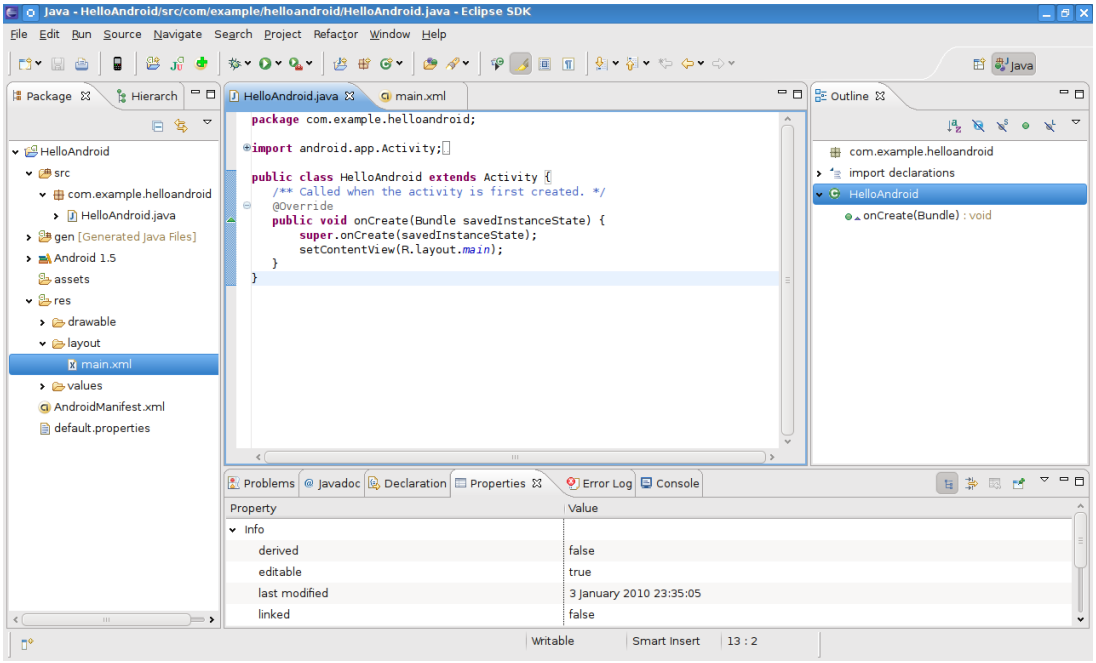

Abbildung 3.2: Android: Eclipse [20]

eines MP3-Players zum Abspielen von Musik und Videos, welche ebenfalls die Grundlage für den iPod Touch bilden.

Die Besonderheit von iOS liegt im Bedienkonzept über den kapazitiven Bildschirm. Der Benutzer kann über Multitouch-Gesten den jeweils angezeigten Inhalt manipulieren. Beispielsweise können Bilder mit zwei Fingern gedreht oder vergrößert werden. Das Aufrufen von Funktionen sowie die Texteingabe funktionieren ebenso über das berühren der jeweiligen Elemente. Die Simulation eines Mauszeigers entfällt somit komplett. Für die Texteingabe erscheint situationsabhängig eine virtuelle Tastatur am unteren Bildschirmrand. Die einzige Hardwaretaste unterhalb des Bildschirms dient zum Schließen der derzeitigen Anwendung und der Rückkehr zum Startbildschirm. Dieses intuitive Bedienkonzept machte das iPhone 2007 zur Innovation und wurde mittlerweile mehrfach kopiert, zum Beispiel von Googles Betriebssystem Android. Der Startbildschirm besteht aus einer Auflistung aller installierten Apps, die durch wischen durchgegangen werden können. Am oberen Bildschirmrand des Startbildschirms befindet sich eine schmale Statuszeile, auf der Informationen wie Empfangssignalstärke, Ladezustand des Akkus sowie Uhrzeit jederzeit ablesbar sind. Die Leiste kann zur Vergrößerung der Bildschirmfläche von Apps ausgeblendet werden. Ebenfalls erwähnenswert ist die Anpassung des Bildschirminhalts bei Drehen des Gerätes, was über die eingebauten Lagesensoren ermöglicht wird.

iOS erlaubt seit Version 4 auch das von PC-Systemen bekannte Multitasking. Hierbei wird eine App beim Schließen nicht komplett aus dem Speicher verdrängt und kann bei erneutem Aufruf schneller und vom letzten Zustand aus aufgerufen werden. Dieses Konzept ist für den schnellen Wechsel sowie Hintergrunddienste notwendig, kann aber die Laufzeit des Akkus beeinträchtigen.

Es folgt eine Auflistung der Grundfunktionen/Apps die auf dem aktuellen iOS für das iPhone 4 vorinstalliert sind. Diese können mit über 500 000 weiteren Apps aus dem App Store ergänzt werden (Stand Dezember 2011[8]).

- Telefon
- FaceTime (ab 4. Generation)
- Nachrichten

- Kalender
- Fotos
- Kamera
- Youtube
- Aktien
- Karten
- iPod
- Mail
- Safari
- Wetter
- Kontakte
- Notizen
- Uhr, Rechner, Sprachaufzeichung, Kompass
- iTunes
- App Store
- Game Center
- Einstellungen

3.2.2 Hardware

Das iPhone in der ersten Generation wird seit dem 29. Juni 2007 in den USA verkauft. Seither ist es international verfügbar und hat 4 Generationen durchlaufen, was in etwa der Anzahl an Jahren seit Markteinführung entspricht, und wurde hierbei jedes Mal in Sachen Ausstattung und Leistung verbessert. Das iPhone 4 unterstützt zur Kommunikation über Mobilfunk GSM und UMTS. Die Datenübertragung geschieht über HSDPA mit bis zu 7,2 Mbit/s Downstream, sowie HSUPA mit bis zu 5,8 Mbit/s Upstream. Als Rückfallebene dient EDGE über das GSM Netz. Für Personal Area Networks verwendet das iPhone 4 Bluetooth 2.1 inklusive Enhanced Data Rate Protokoll. Als Recheneinheit kommt Apples A4 zum Einsatz, welcher von Samsung produziert wird und CPU und GPU auf einem Chip vereinigt. Der verbaute Arbeitsspeicher beträgt 512MB, der Medienspeicher 16GB, 32GB oder 64GB.

Der Bildschirm des iPhone 4 besitzt eine Auflösung von 960 × 640 Pixel auf einer Diagonalen mit 3,5 Inch. Es verfügt somit über eine Pixeldichte von 326 Pixel pro Inch und wird deshab von Apple als Retina-Display bezeichnet, da das menschliche Auge bei normalen Betrachtungsabstand die Pixel nicht mehr erkennen kann. Es bietet besondere Vorteile zur scharfen Darstellung von Schriften und Grafiken. Für Multimediaanwendungen verfügt das iPhone 4 über 2 Kameras: Eine 5 Megapixel Kamera mit LED-Blitz, welche in der Lage ist, HD-Videos aufzunehmen. Außerdem befindet sich auf der Vorderseite eine VGA-Kamera, die die Nutzung von Videotelefonie via Apples Facetime erlaubt. Zusätzlich zu den Fotosensoren bietet das iPhone 4 ein Beschleunigungssensor, sowie ein 3-achsiges Gyroskop, das es ermöglicht, Rotationen in allen drei Raumachsen zu erfassen. Dieses ist vorrangig für multimediale Anwendungen, wie zum Beispiel Spielen, vorgesehen.

Im Gegensatz zu anderen vorgestellten Smartphone-Plattformen wird die Hardware von iOS basierten Geräten von Apple selbst hergestellt und vertrieben. Die verbaute Hardware ist somit einheitlich und lediglich von Generation zu Generation marginal unterschiedlich. Es existieren drei unterschiedliche Gerätetypen mit dem Betriebssystem iOS: Das iPhone, als Smartphone ausgelegt, das iPad, als Tablet PC, sowie dem iPod Touch, einem High-End MP3-Player. iPhone und iPod Touch unterscheiden sich lediglich in dem Fehlen einer Telefonfunktion sowie GPS. Auf allen Geräten befindet sich iOS, aktuell in der Version 4.3.4, welches die Installation und Ausführung von Apps auf allen drei Geräten ermöglicht.

3.2.3 Entwicklung

Die Entwicklung von Apps für das iPhone oder das iPad ist grundsätzlich nur auf einem Mac möglich. Des Weiteren ist eine Mitgliedschaft in Apples iOS Developer Program notwendig. Für die Programmierung und den Test im Simulator genügt eine freie Mitgliedschaft. Sobald jedoch auf einem Gerät getestet werden, oder die App im App Store veröffentlicht werden soll, muss eine kostenpflichtige Mitgliedschaft in Höhe von 99 USD pro Jahr abgeschlossen werden. Um Apps intern an mehrere Geräte zu verteilen ohne den App Store zu nutzen muss eine iOS Enterprise Developer Program Mitgliedschaft in Höhe von 299 USD pro Jahr abgeschlossen werden.

Für die Entwicklung von Homepages unterstützt iOS alle modernen offenen Standards, darunter HTML5, CSS und JavaScript. Hierbei arbeitet Apple selber an der Umsetzung mit und fördert den Einsatz durch die Veröffentlichung des hauseigenen WebKit, das in vielen gängigen Browsern Einsatz findet, beispielsweise Safari und Google Chrome.

Zum Leid einiger Entwickler verzichtet Apple jedoch auf die Unterstützung von Flash für die Entwicklung von Webseiten, Apps oder das Einbetten von Videos wie es auf Youtube der Fall ist. Apple stützt sich hierbei auf mehrere Argumente. Man möchte keine Drittherstellerplattform für die App-Entwicklung zwischen der eigenen Hardware und den Endkunden haben, da dies die Aktualisierung des Basissystems behindern würde. Außerdem sei Flash nicht für die Bedienung über ein Touchdisplay ausgelegt und habe mehrere Sicherheitslücken zu beklagen, die in der Vergangenheit aufgetreten sind. Man möchte mit Flash keine potentielle Gefahrenquelle eröffnen und so das eigene System angreifbar machen. Dies trägt einen Teil zur Sicherheit der iOS Plattform bei.

Objective-C

Die Programmiersprache unter Mac OS X sowie iOS ist Objective-C. Sie stellt eine echte Obermenge von C dar und erweitert C um objektorientierte Sprachkonstrukte. Das bedeutet, dass jedes C-Programm auch von einem Objective C-Compiler kompiliert werden kann. Objective-C ist die primäre Sprache der Cocoa & Cocoa Touch API. Die objektorientierten Erweiterungen sind an Smalltalk angelehnt und von der prozeduralen C-Syntax strikt getrennt.

Cocoa Touch

Die Entwicklerschnittstelle Cocoa Touch ist der Einstiegspunkt für Entwickler in iOS. Cocoa Touch basiert auf der Cocoa API für OS X und bietet grundlegende Funktionen in Frameworks wie dem Foundation Kit Framework und dem UIKit Framework an. Es werden auch viele multimediale Frameworks wie Map Kit oder iAd mitgeliefert, die die Nutzung deren Dienste stark vereinfachen. Cocoa Touch dient als Abstraktionsschicht über den Medien- und Applikationsdiensten, Kernfunktionen (Core Services) bis hin zum Kernsystem von iOS.

Xcode

Xcode ist Apples integrierte Entwicklungsumgebung für die Programmierung für OS X sowie iOS. Um die Entwicklungsumgebung Xcode nutzen zu können, ist ein Intel-basierter Mac mit aktuellem OS X notwendig. Die vorherrschende Programmiersprache ist Objective-C, wobei ebenfalls weitere Sprachen wie C/C++ oder Java unterstützt werden. Xcode bietet die Möglichkeit den entwickelten Quellcode in einem iPhone Simulator zu simulieren. Somit ist es möglich, die Software vor der Bereitstellung auf dem Gerät zu testen. Allerdings simuliert der iPhone Simulator hierbei nur das Betriebssystem iOS, nicht jedoch die Hardware des iPhone. Deshalb ist ein Test auf dem Zielgerät unerlässlich, weshalb eine kostenpflichtige Apple Developer License zur ernsthaften Entwicklung notwendig ist.

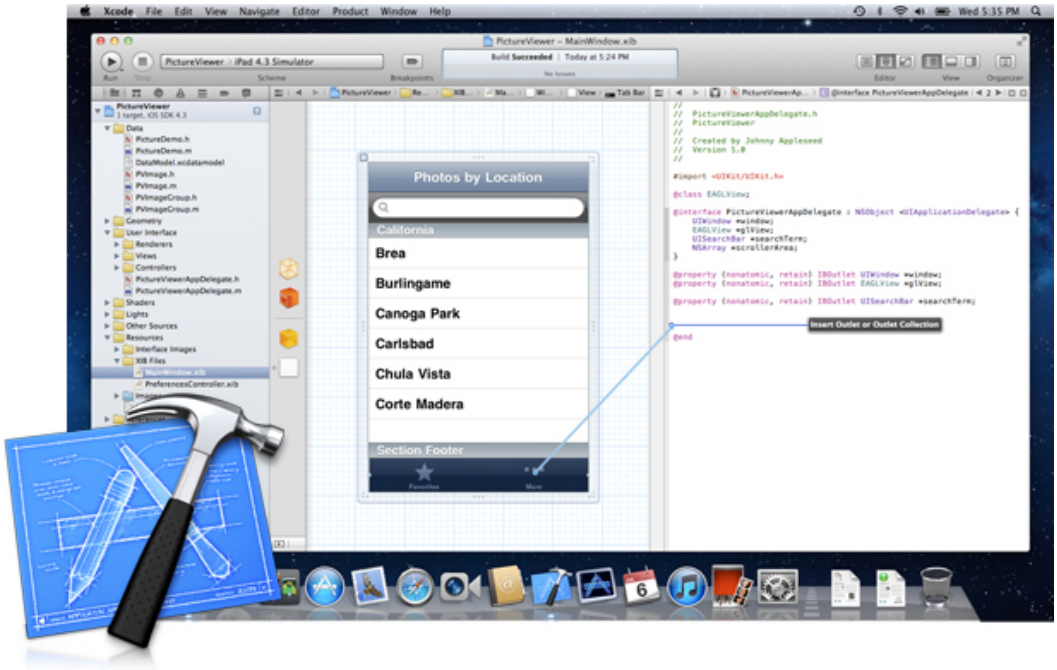

Abbildung 3.3: Xcode IDE Übersicht

Zudem bietet Xcode unter dem Namen Instruments mehrere Tools zur Unterstützung des Entwicklers an. Diese dienen unter anderem der Erfassung von Speicherlecks, der Simulation von Speicherwarnungen oder der Suche nach Performance-Problemen.

Einstellen einer App in den App Store/Verbreitung der App

Bevor eine App im App Store zur Verfügung gestellt wird, prüft Apple diese auf ihre Konformität mit den App Store Review und Human Interface Richtlinien, auf Programmfehlern sowie auf eventuell vorhandenen Schadcode. Werden die Richtlinien nicht erfüllt, so bekommt der Einsender die Möglichkeit die Beanstandungen zu korrigieren und die App erneut hochzuladen. Dieser Prozess kann beliebig oft wiederholt werden. Der Entwickler wird hierbei über den aktuellen Status seiner App-Einsendung informiert.

Das App Review von Apple hat somit einen sicherheitstechnischen Aspekt, da mindestens eine prüfende Instanz zwischen dem Veröffentlichen und der Freigabe stehen. Man kann davon ausgehen, dass zwar nicht jeder Fehler oder schädlicher Eingriff gefunden wird, jedoch eine große Anzahl durch Automatismen abgedeckt und gefunden werden kann. Ebenso hat Apple die Möglichkeit, Apps im Nachhinein zu sperren, welche negativ aufgefallen sind oder Probleme bereiten. Die Sicherheit einer App im App Store ist somit potentiell höher als eine frei verfügbare App bei anderen Plattformen.

Nachteilig ist zu nennen, dass Apple dieses Verfahren auch dazu nutzt, Geschäftsmodelle zu unterbinden, die Ihnen als Konkurrenz erscheinen (iPod) oder schlichtweg nicht zusagen (SevenSnap). Somit ist die Freiheit der Entwickler beim Nutzen der Plattform eingeschränkt. Ebenso kann Apple im Nachhinein Apps verbieten und sperren, welche Ihnen negativ aufgefallen sind, was die Rechte der Benutzer einschränken kann.

Um eine App hochzuladen muss diese zudem mit dem eigenen Entwicklerzertifikat signiert sein. Andernfalls lässt sich die App weder auf einem Gerät ausführen noch in den Store hochladen. Dies garantiert die Echtheit der Software und soll eine weitere Sicherheitsmaßnahme im Einstellungsprozess darstellen.

3.3 Windows Phone 7

Windows Phone 7 ist ein Betriebsystem für Mobiltelefone von Microsoft und ist die Fortsetzung der Software Windows Mobile. Windows Phone 7 ist seit dem 21. Oktober 2010 in Deutschland erhältlich. Ende August 2010 wurde die Entwicklung offiziell abgeschlossen und an die Gerätehersteller und Partner von Microsoft ausgeliefert. Die größte Neuerung von WP7 zu Windows Mobile ist, dass die Bedienung nicht mehr mit einem Eingabestift erfolgt, sondern mit den Fingern als Multi-Touch konzipiert ist.

3.3.1 Software

Windows Phone 7 Applikationen werden mit C# und VB.net in den Technologien Silverlight und XNA entwickelt. Silverlight basiert auf einer reduzierten Version des .NET-Frameworks. Neben der offenen W3C-Webplattform (u.a. Ajax) konkurriert Silverlight mit OpenLaszlo, Adobe Flash/Adobe Flex und JavaFX. Dadurch funktionieren die für Windows Mobile entwickelten Anwendungen nicht mehr.

Für Applikationen stellt Microsoft genau eine offizielle Quelle namens „Marktplace“ zur Verfügung. Zur Registrierung wird die Windows Live ID benötigt, was einen PC oder ähnliches Gerät voraussetzt. Die Apps im Microsofts Marktplace stammen zum größten Teil von Drittanbietern und werden vor der Veröffentlichung von Microsoft zertifiziert. Im Schnitt dauert diese Prozedur 1,6 Tage. Microsoft behält 30% des Verkaufspreises. Im Gegensatz zu anderen Anbietern können Apps vor dem Kauf kostenlos unbegrenzt ausprobiert werden. Eventuelle Funktionseinschränkungen müssen dadurch in Kauf genommen werden. Man kann mittels Handyrechnung oder mit Kreditkarte bezahlen. Der Marktanteil von Windows Phone 7 Geräten liegt bei 6,8%(September 2010[51]), und es gibt ca. 36.000 Entwickler die Apps schreiben. Davon werden ca. 40% veröffentlicht. 63% der Apps erfüllen bereits im ersten Anlauf die strengen Auflagen und Sicherheitsrichtlinien von Microsoft.

Der Homescreen wird in sogenannte „Hubs“ unterteilt. Diese beinhalten die Schwerpunkte Kontakte, Office, Bilder, Social Network, Musik, Video und Spiele. Ändert sich der Status der Anwendungen so sieht der Benutzer direkt auch eine Änderungen der Hubs auf dem Homescreen. Wird Beispielsweise eine Nachricht von Facebook empfangen so ändert sich die Darstellung des Hubs „Kontakte“ und man hat sofort einen Überblick über alle Nachrichten. Der Hub „Spiele“ stellt eine Anbindung zu Microsoft Xbox Live zur Verfügung, welche es dem Benutzer ermöglicht, über das Marketplace, multimediale Artikel zu beziehen und sie gleichzeitig von den verschiedenen Geräten gleichzeitig nutzen. Man kann zum Beispiel ein Videospiel am PC starten, auf dem WP7 weiterspielen und auf der Xbox beenden.

3.3.2 Hardware

Bei der Hardware gibt Microsoft Richtlinien an, an die sich Gerätehersteller halten müssen. Windows Phone 7 Geräte müssen über einen Prozessor mit mindestens 1 GHz sowie Arbeitsspeicher von mindestens 256 MB verfügen. Zudem müssen mindestens 8 GB an Flash-Speicher, ein Kompass, ein GPS Empfänger sowie ein Beschleunigungs- und Lagesensor im Gerät verbaut sein. Des Weiteren muss das Gerät ein kapazitiven Multitouch Bildschirm, eine 5 Megapixel Kamera sowie eine Mobilfunk- und WLAN-Schnittstelle bereit stellen. Eine weitere Besonderheit bei WP7 Geräten ist, dass sich unter dem Touchscreen genau drei Knöpfe befinden müssen. Einer für die Funktion „Zurück“, einer für „Suche“ und der Dritte für den „Homescreen“.

3.3.3 Entwickung

Für Entwickler stellt Microsoft das kostenlose Paket „Windows Phone Developer Tools“ welches die Programme: Microsoft Visual Studio 2010 Express for Windows Phone, einen Emulator, die Silverlight- und XNA Ressourcen und Expression Blend zur Gestaltung der Oberfläche zur Verfügung.

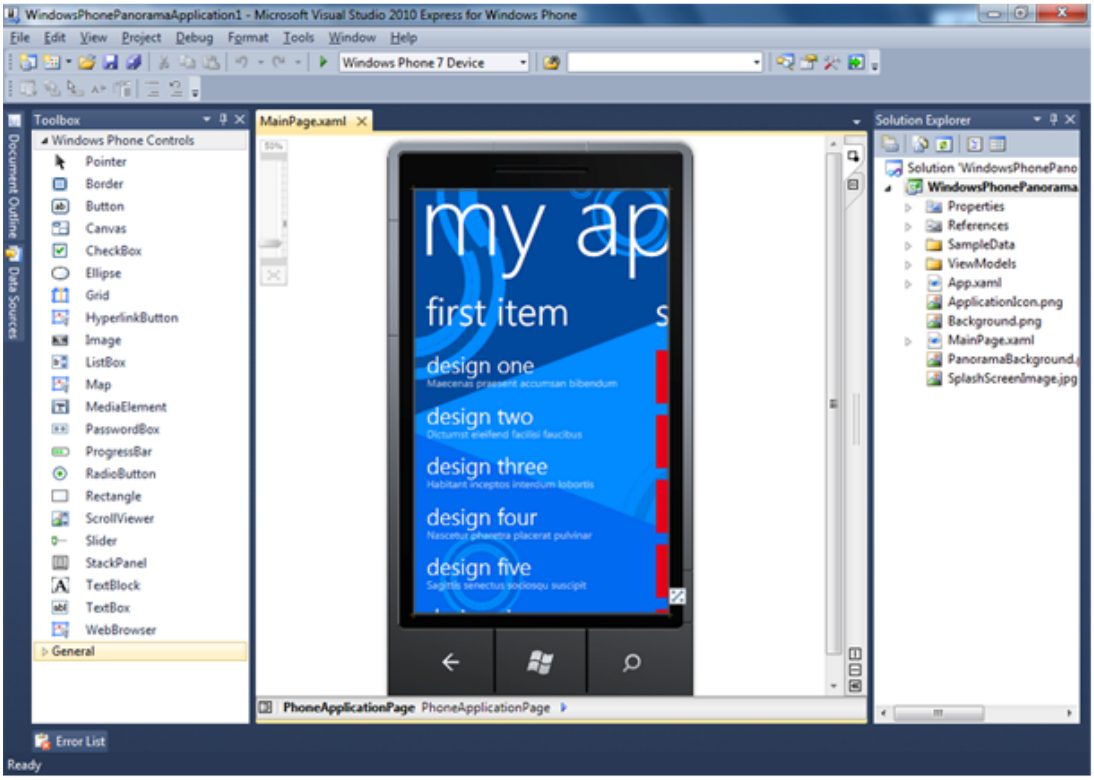

Abbildung 3.4: Visual Studio Express IDE

3.4 webOS

webOS ist ein von HP Palm entwickeltes Betriebsystem für Smartphones und Tablets. Es wurde im Januar 2009 erstmals vorgestellt und ist der Nachfolger von Palm OS. Im Gegensatz zu Palm OS ist es auf die Bedienung per Touchscreen angepasst, unterstützt Multitasking und integriert Web2.0-Technologien.

3.4.1 Software

webOS verwendet, wie Android, den Linuxkernel als Basis. Der Name „web OS“ soll auf eine enge Verbindung zum Internet und dessen Dienste hinweisen. Dazu wird eine Technik/Dienst namens „Synergy“ eingesetzt, um Daten (zum Beispiel E-Mail, Kalender, Kontakte, SMS) verschiedener Dienste zusammenzufassen, per Internet zu synchronisieren und einheitlich anzuzeigen.

Mit dem „HP App Catalog“ stellt HP eine offizielle Quelle für Applikationen bereit. Die Bezahlung erfolgt per Kreditkarte oder über die Handyrechnung. HP Palm behält 30% des Verkaufspreises. Bevor Applikationen in den „App Catalog“ aufgenommen werden, werden diese durch das „HP App Review Team“ überprüft. Dies kann bis zu 2 Wochen dauern. Zusätzlich zum „App Catalog“ gibt es noch die Möglichkeit Applikationen per „Web distribution“ zu veröffentlichen. Hierbei werden die Applikationen nicht von „HP Palm“ überprüft und sind auch nicht im „App Catalog“ auffindbar. Die Applikationen sind nur über eine URL aufrufbar. Die Verteilung der URL bleibt dem Entwickler überlassen.

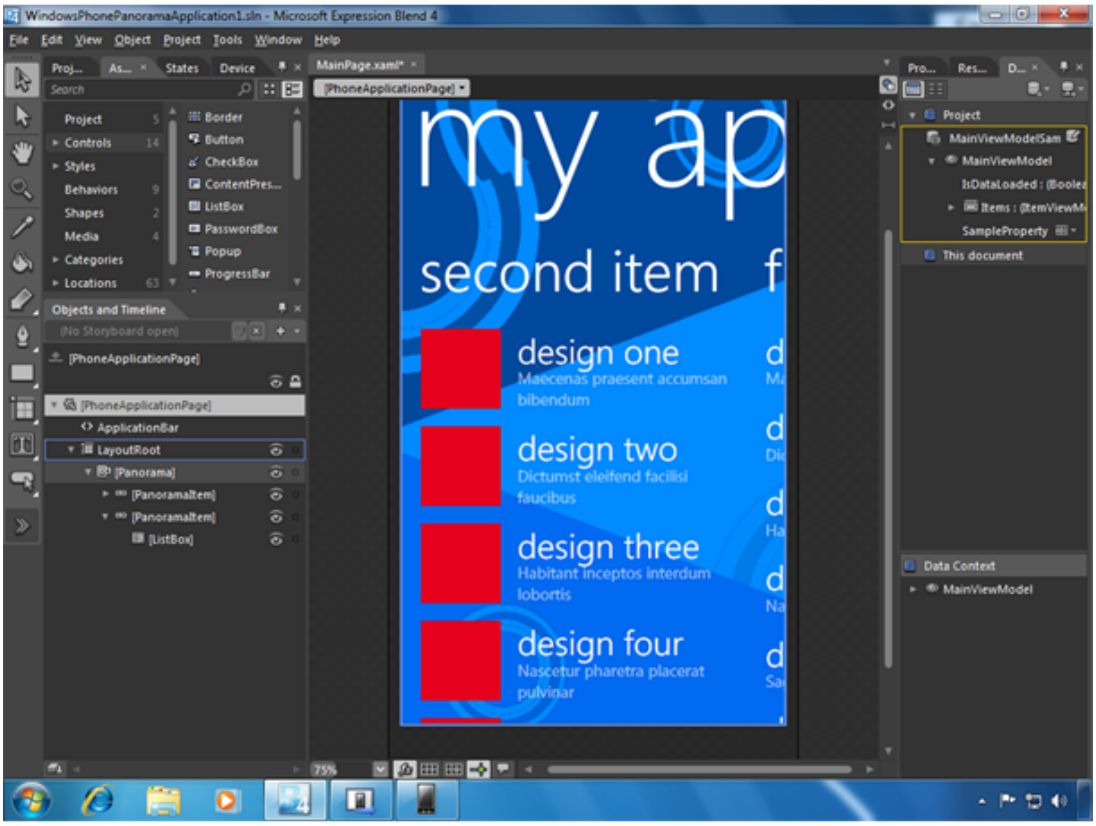

Abbildung 3.5: Expression Blend IDE

3.4.2 Hardware

webOS kann Verbindungen per Mobilfunk (HSDPA, EDGE, GSM, sowie CDMA), WLAN 802.11b/g/n und Bluetooth 2.1 aufbauen. Eine Unterstützung von NFC ist geplant. Folgende Sensoren werden unterstützt: Beschleunigungssensor, GPS, Annäherungssensor, Helligkeitssensor. Bilder und Videos können über eine Kamera aufgenommen werden.

Im Februar 2011 hat HP drei neue Geräte mit webOS vorgestellt:

- HP Pre^3: 3,58 Zoll Display und mit einer Auflösung von 480×800, ausziehbarer QWERTY-/QWERTZ-Tastatur, eine 5 Megapixelkamera mit Blitz, die HD-Videos aufnehmen kann und auf der Vorderseite eine VGA-Kamera. Es gibt eine Variante mit 8 GB und 16GB internem Speicher. Als CPU kommt ein Qualcomm Snapdragon MSM8x55 mit 1.4 GHz zum Einsatz. Das HP Pre^3 wird mit webOS 2.2 ausgeliefert.
- HP Veer: 2,6 Zoll Display und mit einer Auflösung von 320×400, ausziehbarer QWERTY-/QWERTZ-Tastatur, eine 5 Megapixelkamera und 8GB internen Speicher. Als CPU kommt ein Snapdragon MSM7230 mit 800Mhz zum Einsatz. Das HP Veer wird mit webOS 2.1.2 ausgeliefert.
- HP TouchPad, ein Tablet mit 9,7 Zoll Touchscreen und einer Auflösung von 1024×786, 1,3 Megapixelkamera und Qualcomm Snapdragon dual-CPU APQ8060 mit 1.2 GHz. Es gibt eine Variante mit 16GB und mit 32 GB internem Speicher. Online Verbindungen können nur per WLAN hergestellt werden. Ein 3G Variante ist in Planung. Das HP TouchPad ist das erste Gerät mit webOs 3.0.

Das Pre^3 kam Anfang August 2011 auf den Markt.

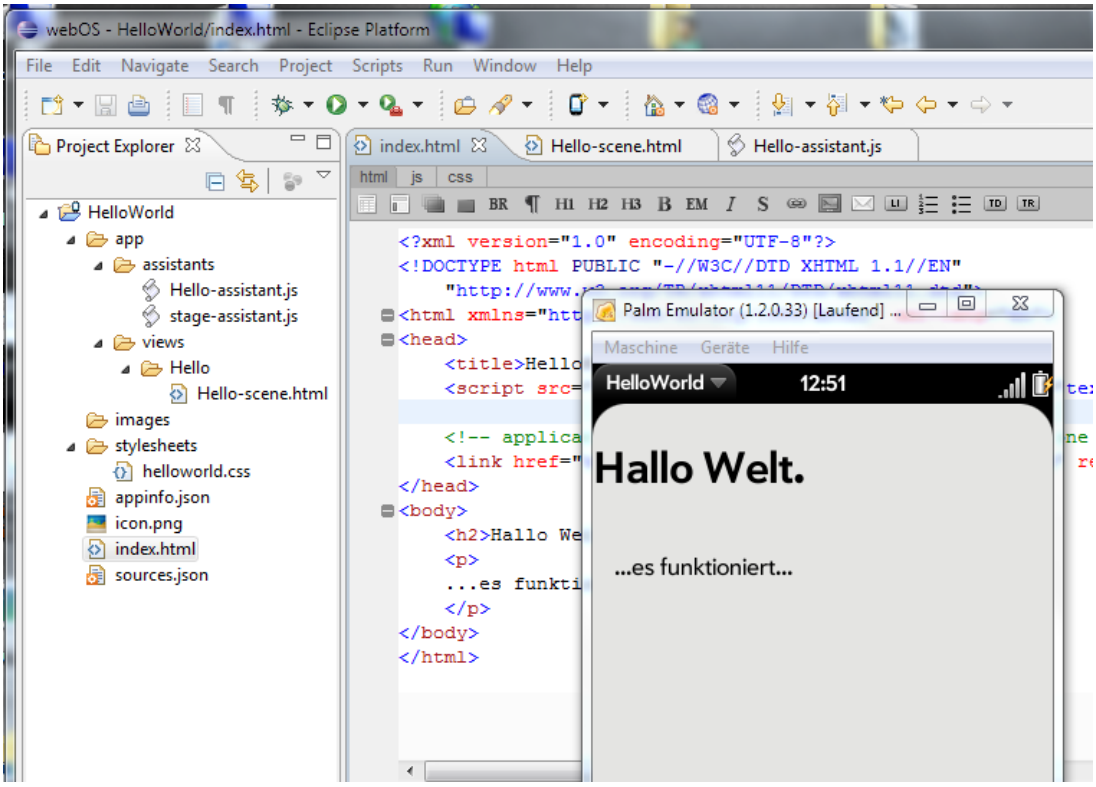

Abbildung 3.6: webOS: Eclipse

3.4.3 Entwicklung

Da webOS ein web-orientiertes Betriebsystem sein soll, werden Apps mit Hilfe von HTML5, CSS und JavaScript entwickelt und mit Hilfe von Googles JavaScript Engine V8 ausgeführt. Zusätzlich gibt es seit März 2010 die Möglichkeit in C/C++ zu entwickeln. Für die Entwicklung stellt HP ein Plugin für Eclipse (Abbildung 3.6) bereit.

Um Apps können entweder direkt auf einem, an den PC angeschlossenen oder auf einem virtuellen Gerät getestet werden

3.4.4 Einstellung der Entwicklung

In einer Pressemitteilung vom 18.08.2011 nimmt HP Stellung zu den Umsatzzahlen des dritten Geschäftsquartals und einer bevorstehenden Umstrukturierung des Konzerns. HP kündigte hierbei an, den Consumer-PC Bereich und die webOS Sparte einzustellen. Hiervon betroffen sind unter anderem alle Touchpads und Smartphones auf Basis von webOS. Als Erklärung nennt HP, die Geräte hätten die Ziele des Konzerns hinsichtlich Umsatz und Verbreitung nicht erreicht. Man möchte sich im Zuge der Umstrukturierung mehr auf die Enterprise-Sparte konzentrieren. HP erwähnte im gleichen Atemzug den britischen Software- und Cloud Spezialisten Autonomy für rund 10 Milliarden US-Dollar zu übernehmen.

3.5 Black Berry OS

BlackBerry (zu deutsch: Brombeere) ist ein Smartphone, welches speziell für das Lesen und Schreiben von E-Mails konzipiert ist. Das kanadische Unternehmen „Research In Motion“, kurz RIM, entwickelte eine Lösung für die drahtlose Kommunikation und Verwaltung von Business Daten (Personal Information Manager,

kurz PIM). Es umfasst eine Client-Server Architektur und ein proprietäres Protokoll zwischen Server und Client. RIM vertreibt für seine Dienste fast ausschließlich eigene Geräte. Geräte von RIM werden auch BES-Geräte genannt, da sie auf einen eigenen Server namens „BlackBerry Enterprise Server“ zugreifen. Hierbei werden nur kleine Datenpakete an das Smartphone versendet, die nicht größer als 2 kB sind. So wird gewährleistet, das der Mobilfunkvertrag, welcher auch speziell an das Blackberry angepasst ist, keine hohen Kosten verursacht. Nachrichten, Kalendereinträge oder Firmendaten gelangen, wie bei einer SMS, als Push-Dienst auf das Endgerät.

Sind E-Mails größer als 2kB, so kann der Benutzer die E-Mail vollständig vom Server herunterladen und selbst bestimmen, welche E-Mails komplett angezeigt werden sollen. Das Besondere daran ist, dass die Anforderung der größeren Datenmengen ohne Verzögerung geschieht und selbst größere Daten, wie zum Beispiel PDFs, schnell geöffnet werden. Dies geschieht, in dem der BES-Server die Daten in textorientierte Pakete umwandelt und dann bündelweise zum Endgerät schickt. RIM-Geräte können diese Datenpakete nicht nur mit einem Passwort sichern, sondern die gesamte Übertragung verschlüsseln. Eine weitere wichtige Funktion von BlackBerrys nennt sich Mobile Data System (MDS). Es wurde im vierten Quartal 2006 eingeführt und erlaubt es, mit der BES-Übertragungstechnik auch andere Daten aus dem Firmennetzwerk, zum Beispiel aus Datenbanken oder ERP-Systemen, zu laden. So lassen sich Bestellvorgänge auslösen, Lagerbestände abfragen oder Kundendaten ändern.

3.5.1 Software

Das Betriebssystem ist in C++ programmiert und bietet eine Java Laufzeitumgebung (J2ME – MIDP), mit speziellen Schnittstellen zum Betrieb von Anwendungsprogrammen. Für Geräte der 8000er Serie gibt es das Betriebssystem 5.0 in der aktuellsten Version. Geräte der 9000er Serie können mittlerweile mit dem neusten Betriebssystem „BlackBerry 7 OS“ betrieben werden. Im Oktober 2011 wurde die Gerätesoftware „BlackBerry 10“ vorgestellt, die 2012 auf den Markt kommen soll und sowohl für Smartphones, als auch Tablet-PCs erhältlich sein wird. Der Kern des Betriebssystems stammt von QNX [50], deren proprietäres Echtzeitbetriebssystem in vielen eingebetteten Systemen zum Einsatz kommt und zum Beispiel in der Anlagensteuerung eingesetzt wird. Laut RIM soll auch HTML5 [64] unterstützt werden. Apps können in der Programmiersprache C und C++ entwickelt werden. Nach Medienberichten soll es außerdem möglich sein, Android-Apps auf Blackberry 10 auszuführen.

3.5.2 Hardware

BlackBerry Geräte werden meist mit einer Hand bedient und verfügen neben einer QWERTZ-Tastatur über den BlackBerry typischen Trackball, mit dem man durch das ganze Betriebssystem navigieren kann. Zusätzlich gibt es noch eine „Zurück“-Taste, welche in Daumennähe liegt. Neuere Smartphones der Firma RIM haben mittlerweile eine Multi-Touch Bedienoberfläche, die sich mit den Fingern steuern lässt. Statt einem Trackball, welcher oft durch Verschmutzung blockierte, wurde ein Trackpad, wie man sie von Notebooks her kennt, eingebaut. Neben den Geräten der Herstellerfirma RIM, welche die volle BES-Technologie unterstützen, gibt es noch weitere Geräte, die nur einen Teil der Funktionalität anbieten. BlackBerry-Connect Geräte, wie zum Beispiel der Nokia N90 Communicator, unterstützen ab der Firmware 4.0 3DES- und AES-Verschlüsselung und können zudem Daten wie E-Mail, Kontakte und Kalender sofort auf dem Gerät anzeigen. Allerdings wird dabei nicht der Push-Dienst von BlackBerry verwendet, das Gerät greift statt dessen in einem vordefinierten Intervall auf den Server zu. Dies kann zu einem erhöhten Datenverkehr führen, gegenüber vergleichbaren BlackBerry Geräten von RIM.

BlackBerry „Built-In“ verwendet für die Verschlüsselung das 3DES Verfahren und unterstützt E-Mail, Kalender, Kontakte, Browser, Aufgaben und Notizen. Das einzige Gerät, welches bisher „Built-In“ unterstützt, ist das Siemens SK65.

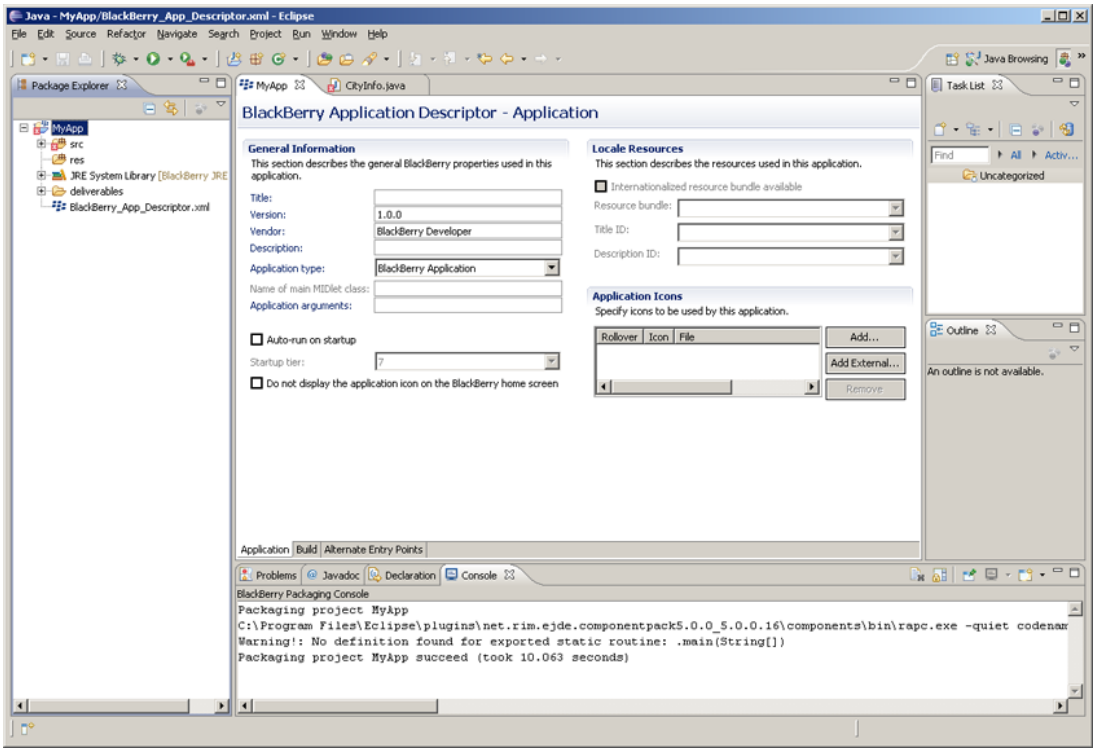

Abbildung 3.7: BlackBerry: Eclipse Plugin

RIM Geräte sind grundsätzlich mit einer Kamera, GPS, WLAN und Bluetooth ausgestattet. Der interne Speicher kann meist über eine Speicherkarte erweitert werden.

3.5.3 Entwicklung

Zur Entwicklung von Apps setzt RIM auf das, im Vergleich zu den aktuellen Java-Versionen, abgespeckte Java ME (Micro Edition). Der Sprachumfang entspricht etwa Java SE 1.3, allerdings stehen verschiedene zusätzliche Bibliotheken zur Verfügung. Moderne Java-Konzepte wie Generics werden nicht unterstützt, das erschwert die Umsetzung von bereits existierendem Java-Code. Der Entwickler kann in seiner gewohnten Entwicklungsumgebung wie Eclipse oder Microsoft Visual Studio arbeiten, oder in der herkömmlichen BlackBerry Java Entwicklungsumgebung (BlackBerry JDE). Letztere lässt sich auch von der offiziellen Webseite `http://de.blackberry.com/developers/` herunterladen. Passende Richtlinien, sowie hilfreiche Foren, lassen sich genauso auf dieser Seite finden, als auch einige Video Tutorials.

4 Security

4.1 Definition: Was ist Security?

Als Security bezeichnet man die Eigenschaft von informationsverarbeitenden und -lagernden Systemen, die Vertraulichkeit, Verfügbarkeit und Integrität sicher zu stellen. Security dient dem Schutz vor Gefahren und Bedrohungen, der Vermeidung von Schäden und der Minimierung von Risiken. Der Begriff bezieht sich oft auf eine globale Informationssicherheit, bei der die Zahl der möglichen schädlichen Szenarien summarisch reduziert ist, oder der Aufwand zur Kompromittierung, das heißt das Daten manipuliert sein könnten, für den Angreifer in einem ungünstigen Verhältnis zum erwarteten Informationsgewinn steht. Die Schutzziele von Security definieren sich auf den folgenden Teilbereichen:

Datensicherheit Bezieht sich auf alle relevanten Informationen einer Organisation oder eines Unternehmens, einschließlich personenbezogener Daten. Darunter fallen die Unterpunkte:

- **Vertraulichkeit** Daten dürfen lediglich von autorisierten Benutzern gelesen und modifiziert werden, dies gilt sowohl beim Zugriff auf gespeicherte Daten, wie auch während der Datenübertragung.
- **Integrität** Daten dürfen nicht unbemerkt verändert werden. Es müssen alle Änderungen nachvollziehbar sein.
- **Verfügbarkeit** Verhinderung von Systemausfällen. Der Zugriff auf Daten muss innerhalb eines vereinbarten Zeitrahmens gewährleistet werden.

Authentizität Echtheit und Glaubwürdigkeit einer Person, oder eines Dienstes, müssen überprüfbar sein.

IT-Sicherheit Einrichtung und Aufrechterhaltung geeigneter betrieblicher und technischer Maßnahmen, um die Einhaltung der Schutzziele bei IT-gestützter Verarbeitung von Informationen zu gewährleisten.

Datenschutz (auch Privacy) Hierbei geht es um den Schutz persönlicher Daten, die laut Bundesgesetzbuch unter das Datenschutzrecht fallen. Geschützt werden muss dabei die Privatsphäre des Einzelnen, um dessen Anonymität zu wahren. Vor allem im Internet nimmt dieser Punkt aufgrund wachsendem Datenhandel und sozialen Netzwerken eine hohe Bedeutung an und wird deshalb in Kapitel 5, Privacy genauer betrachtet.

4.2 Motivation

In einem Sicherheitsreport der IBM vom Oktober 2011 [24] wurden 12 Milliarden Sicherheitsvorfälle seit Beginn des Jahres untersucht. Ergebnis: Immer mehr Angriffe zielen auf mobile Plattformen ab. Ursachen sind, unter Anderem, die langsame Reaktion der Hersteller auf Sicherheitslücken, sowie die Tatsache, dass sich immer mehr private Smartphones in Firmennetzwerken befinden („Bring your own device“). Somit stellen Smartphones Einstiegspunkte für Hacker in, bis dato, sichere Firmennetzwerke dar. Schadcode oder Malware (Unterabschnitt 4.3.1, Malware) wird derzeit hauptsächlich über App-Märkte von Drittanbietern verbreitet. Bekannte Angriffsmuster sind der Diebstahl von Identitätsdaten über Phishing-Attacken (Unterabschnitt 5.3.3, Phishing), oder Betrug durch die Verbreitung von gebührenpflichtigen SMS-Premiumdiensten (Unterabschnitt 4.3.2, Dialer). Die IBM-Forscher prognostizieren bis Ende des Jahres eine Verdopplung solcher mobilen Angriffe. Neben diesen scheinbar harmlosen Angriffen, gibt es jedoch auch gezielte Angriffe auf

Personen der Führungsetage, die als „Whaling“ bezeichnet werden. Diese sind auf eine Verdreifachung des Prozentsatzes kritischer Schwachstellen zurückzuführen und zählen zu den „Advanced Persistent Threats“. Kombinationen aus Bespitzelung und Phishing-Aktionen sollen den Verbrechern hier zum Erfolg helfen. Täter mit politischen Hintergründen werden im Report ebenso verzeichnet. Diese erfolgen meist durch die Verbreitung von Inhalten über anonymisierte Proxy-Netzwerke, sind jedoch nicht auf mobile Anwender zielgerichtet. Somit ist es notwendig, sich mit dem Thema „Mobile Sicherheit“ auseinanderzusetzen, um Mobilgeräte ausreichend vor drohenden Gefahren (Abschnitt 4.3, Gefahren) zu schützen.

4.3 Gefahren

4.3.1 Malware

Bei Malware handelt es sich um Software, welche dem Nutzer potentiell Schaden zufügen kann. Unter dem Begriff lässt sich eine große Bandbreite gefährlicher Programme zusammenfassen, beispielsweise auch Viren und Trojaner, wie in Kapitel Unterabschnitt 5.3.7, Trojaner beschrieben. Malware wird entweder verwendet, um dem Nutzer über kostenpflichtige Dienste Geld zu entlocken, oder um sensible Daten zu erspähen, welche dem Täter ebenfalls Geld einbringen können, zum Beispiel durch Verkauf an Interessenten wie Werbeagenturen. Mögliche Varianten des Datendiebstahls werden in Kapitel 5, Privacy beschrieben.

Auf Smartphones handelt es sich bei Schadsoftware in der Regel um Apps, die über inoffizielle Wege auf das Gerät gelangt sind, oder die Sicherheitsvorkehrungen der offiziellen App-Märkte umgehen konnten. Informationen zu den angebotenen Märkten finden sich in Abschnitt 4.4, Sicherheit von Apps, Informationen zum Schutz vor feindseligen Apps in Abschnitt 4.10, Sicherheitssoftware für Smartphones.

4.3.2 Dialer

Zu den Gefahren vernetzter Computer zählen, unter anderem, so genannte Dialer. Diese wählen teure Telefonnummern über die eigene Verbindung, um dem Nutzer unbemerkt sein Geld zu entlocken. Entdeckt der Nutzer den Diebstahl auf seiner Abrechnung, ist es in den meisten Fällen nicht mehr möglich, das Geld zurückzufordern.

Doch Dialer waren lange Zeit kein Thema mehr: Seit der Einführung von Flatrates, mit dauerhaften Verbindungen ohne Einwahlmöglichkeit, verhindern diese den Missbrauch der Telefonleitung für teure Telefonate. Somit verloren Dialer seit der breitflächigen Verfügbarkeit von DSL ihre Popularität. Auf modernen Smartphones sind diese Dienste jedoch wieder ein auferstanden, allerdings in Form von teuren Premium-SMS-Diensten. Diese können beispielsweise von Trojanern verschickt werden, um dem Nutzer unbemerkt zu schaden, siehe auch Unterabschnitt 5.3.7, Trojaner. Apps tarnen sich hierbei in Form von Spielen, Hintergrundbildern oder anderen scheinbar nützlichen Anwendungen. Diese schadhaften Apps sind im Android Market leider häufiger vertreten, als in Apples App Store. Dies ist auf die gründlichere Prüfung der Apps durch Apple zurückzuführen, welche das Einschleusen von schadhaften Apps deutlich verringert, siehe Abschnitt 4.4, Sicherheit von Apps. Google zieht bei Entdeckung einer betrügerischen App jedoch die Notbremse und löscht diese aus dem Market, sowie von betroffenen Endgeräten. [14]

Um dem zu entgegnen, sollten Android Nutzer die Erwägung einer Antivirensoftware, zum Schutz vor Dialern, in Erwägung ziehen. Ein Überblick findet sich in Abschnitt 4.10, Sicherheitssoftware für Smartphones.

4.4 Sicherheit von Apps

Dieses Kapitel geht auf die Sicherheit bei der Ausführung und Entwicklung von Apps ein. Es werden Systemmechanismen der einzelnen Systeme vorgestellt, die eine Ausführung von schadhaftem Code verhindern sollen. Zudem werden die Bezugsquellen von Software und deren Sicherheit betrachtet.

4.4.1 Android

Im Gegensatz zu anderen Plattformen, gibt es für Android verschiedene Bezugsquellen für Apps. Android Apps liegen als APK (Application Package File)-Pakete vor. Diese können direkt auf dem System installiert werden. Dies kann allerdings vom Bereitsteller des Betriebssystems (meist Hersteller oder Netzbetreiber) verhindert werden. Die größte Verbreitung hat der offizielle „Android Market“ von Google. Dieser ist auf allen gängigen Android Geräten als App vorinstalliert. Er verzeichnet laut [14] zum Stand Dezember 2011 über 300.000 Apps.

Google stellt durch die Zahlung einer Registrierungsgebühr die Identität der Entwickler sicher. Um eine App einem Entwickler eindeutig zuordnen zu können, muss diese vom Entwickler signiert werden. Unsignierte Apps können nicht installiert werden. Google stellt sehr geringe Richtlinien an die Entwickler (`http://www.android.com/de/developer-distribution-agreement.html`). Diese dienen weitgehend dazu, Schadsoftware und missbräuchlichen Gebrauch des Markets zu verhindern. Eine Überprüfung von Apps vor der Veröffentlichung findet nicht statt. Google setzt darauf, dass Nutzer bösartige Software erkennen und melden. Daraufhin hat Google die Möglichkeit, diese aus dem Market und von betroffenen Endgeräten zu löschen.

Android Apps haben nur eingeschränkten Zugriff auf das System. Dazu verwendet Android die Benutzerverwaltung des Linux-Kernels: Jede App läuft unter einem eigenem Linux-Benutzer, so dass diese nur auf den ihr zugewiesenen Speicher (sowohl Arbeitsspeicher, als auch dauerhaften Speicher) zugreifen kann. Um auf Daten außerhalb dieses Speicherbereiches zuzugreifen, müssen extra Berechtigungen vom System angefordert werden (zum Beispiel Zugriff auf die Kamera), dadurch wird missbräuchlicher Zugriff von Apps auf Nutzerdaten verhindert. Diese Berechtigungen werden dem Nutzer vor der Installation der App, wie in Grafik Abbildung 4.1 dargestellt, angezeigt.

Es ist möglich, dass Apps mehr Rechte anfordern, als für den eigentlich Gebrauch nötig wären. Ein Beispiel hierfür könnte eine App sein, die das aktuelle Wetter anzeigt. Hierzu werden Rechte zum Zugriff auf das Internet und vielleicht auf den aktuellen Standort benötigt. Fordert die App aber das Recht, um auf das Adressbuch zuzugreifen, so sollte der Benutzer misstrauisch werden und diese App melden, so dass sie überprüft wird. Hat sich nach der Überprüfung herausgestellt, dass es sich bei einer App um Schadsoftware handelt, so hat Google die Möglichkeit diese App automatisch von Geräten zu entfernen, um so die Nutzer zu schützen. Allerdings geschieht dies ohne eine Benachrichtigung des Nutzers. Zuletzt nutzte Google dieses Privileg im Dezember 2011. Es wurden auf Hinweis vom Sicherheitsexperten Lookout 27 schädliche Apps aus dem Market entfernt, die, unter Anderem, kostenpflichtige Premium-SMS verschickten und so ihren Nutzern schadeten. Die Apps tarnten sich hierbei als Horoskop, Bildschirmhintergrund oder Spiel. [14]

Um zusätzliche Sicherheit zu bieten, bietet der Android Market nach [39] verschiedene Filtereinstellungen:

Alle Stufen Anwendungen dieser Kategorie dürfen weder Daten zum Standort der Nutzer erfassen, noch unangemessene Inhalte aufweisen. Die Anwendungen dürfen die Nutzerinhalte nicht freigeben und keine sozialen Funktionen enthalten.

Niedrige Stufe Anwendungen dieser Kategorie können leichte Gewalt in Cartoons oder andere, möglicherweise anstößige Inhalte enthalten. Die Anwendungen erfassen unter Umständen Daten zum Standort der Nutzer, um Standort-spezifische Informationen bereit zu stellen, oder die Nutzererfahrung anderweitig zu verbessern. Die Daten dürfen jedoch nicht an andere Nutzer weitergeleitet werden. Die Anwendungen können soziale Funktionen enthalten, dürfen jedoch in erster Linie nicht dazu dienen, dass Nutzer sich gegenseitig finden und miteinander kommunizieren können.

Anwendungen dieser Kategorie können sexuelle Anspielungen, intensive Fantasiegewalt oder realistische Gewalt, Obszönitäten und derben Humor, Anspielungen auf Tabak-, Alkohol- und Drogenkonsum sowie simuliertes Glücksspiel enthalten. Die Anwendungen können Daten zum Standort der Nutzer erfassen, um diese mit dem Einverständnis des Nutzers weiterzuleiten oder zu veröffentlichen.

Abbildung 4.1: Installation von Apps

Hohe Stufe Anwendungen dieser Kategorie können in größerem Umfang sexuelle und anzügliche Inhalte, grafische Gewalt, soziale Funktionen, simuliertes Glücksspiel sowie wirkungsvolle Anspielungen auf Tabak-, Alkohol- und Drogenkonsum enthalten. Die Anwendungen können Daten zum Standort der Nutzer erfassen, um diese mit dem Einverständnis des Nutzers weiterzuleiten oder zu veröffentlichen.

Neben dem Android Market gibt es noch verschiedene andere Quellen für Apps. Die bekannteste Alternative ist der Amazon App Store. Im Gegensatz zum Android Market, gibt es strengere Richtlinien für Apps (`https://developer.amazon.com/help/developer-conditions.html`) und es findet eine Vorabprüfung statt.

Anstatt Apps über einen Market zu beziehen, gibt es auch noch die Möglichkeit Apps direkt auf dem Gerät zu installieren. Dies muss jedoch explizit auf dem Gerät aktiviert werden. Einige Hersteller und Netzbetreiber haben dies aus Sicherheitsgründen auf ihren Geräten deaktiviert. Bei direkt installierten Apps ist der Nutzer selbst für die Sicherheit verantwortlich. Um eine direkte Installation zu ermöglichen, muss die Einstellung „Unbekannte Herkunft“ unter *Einstellungen/Apps* aktiviert sein.

4.4.2 iOS

Apps für Apples Smartphone Plattform iOS unterstehen einer Reihe von Sicherheitsvorkehrungen. Diese sollen dafür Sorgen, dass Apples hohe Sicherheits- und Qualitätsstandards auch bei Apps durch Dritthersteller durchgesetzt werden.

iOS Apps haben generell nur beschränkten Zugriff auf das Dateisystem und die Prozesse von iOS. Dies wird durch eine Sandbox Umgebung erreicht, ein dedizierter Speicherbereich, in dem eine App arbeiten darf. Zudem existiert auf ARM-Prozessoren eine Datenausführungsverhinderung namens eXecute Never, um die Ausführung von Code in geschützten Speicherbereichen zu unterbinden. Ist für einen Speicherbereich der eXecute Never-Flag gesetzt, so darf an dieser Stelle kein Zugriff durch Code erfolgen. Der Zugriff auf außenliegende Funktionen somit ist nur über die von Apple angebotenen Schnittstellen des Foundation Frameworks und Cocoa Touch möglich. Eine Einbindung von fremden Frameworks ist grundsätzlich untersagt. Es ist jedoch möglich, Drittanbieter-Funktionen über die Einbindung von Quelltext zur Verfügung zu stellen. Diese arbeiten jedoch in der selben Sandbox Umgebung wie die App selbst. Somit ist es Apple möglich, den eingebundenen Code während des Reviews ebenso einem Test zu unterziehen und auf Schadcode zu prüfen. Der Aufruf einer außenstehenden App sowie die Installation einer App durch Programmcode ist ebenso untersagt, um Schadsoftware vorzubeugen. Weiterhin ist die Programmierung, sowie die Bereitstellung von iOS Apps nur mit der hauseigenen Entwicklungsumgebung Xcode möglich. Hierzu benötigt man als Entwickler mindestens einen Intel-basierten Mac mit Snow Leopard.

Um Apps für das iPhone programmieren zu können müssen Entwickler dem Apple Developer Program angehören Diese personengebundene Lizenz kostet 99$ pro Jahr und erlaubt das Testen auf einem Gerät sowie das Bereitstellen einer App im App Store. Dieser ist die einzige Möglichkeit Apps für iOS Geräte zu vertreiben und zu erhalten und wird von Apple streng kontrolliert. Jede App muss vor der Bereitstellung mit der Developer Lizenz digital signiert werden. Somit ist sichergestellt, dass keine unbekannten Apps in den App Store gelangen können, sowie eine Rückverfolgung des Entwicklers möglich zu machen.

Vor der Verbreitung einer App über den App Store unterzieht Apple diese einem ausführlichen Review. Dies dauert in der Regel 2 Wochen. Apple prüft hierbei die App auf unzulässige Programmaufrufe und Schadcode und meldet dies dem Entwickler über das Portal iTunes Connect zurück. Gibt es Beandstandungen so muss nachgebessert werden, gefolgt von einem erneuten Review. Die Richtlinien für dieses Review beziehungsweise der Erstellung einer App bietet Apple auf ihrer Developer Webseite zur Einsicht an. Apple behält sich zudem die Möglichkeit vor, Apps nachträglich im App Store zu sperren oder durch eine Fernsteuerfunktion auf Endgeräten zu löschen. Somit können im Nachhinein gefundene Schädlinge unschädlich gemacht werden.

Sichere Entwicklung

Die folgenden Sicherheitstipps sollten laut [58] berücksichtigt werden, wenn Apps mit vertraulichen Informationen umgehen, oder im Unternehmen eingesetzt werden sollen. Die Quellensammlung des Artikels findet sich unter [32].

Apple bietet zum Schutz des Speichers eine Speicherverwürfelung namens Adress Space Layout Randomization (ASLR). Diese wurde mit iOS Version 4.3 eingeführt erschwert durch zufällige dynamische Speicheradressierung die Ausnutzung von Softwareschwachstellen. Dies sollte während der Entwicklung bei der Wahl der Zielplattform berücksichtigt werden. Dies sollte möglichst aktuell aber nicht kleiner 4.3 gewählt werden. Die Wahl der Zielplattform stellt somit einen Kompromiss zwischen Sicherheit und Kompatibilität dar. Apple bietet zwei Varianten von ASLR an: limitiertes ASLR, mit teilweise statischen Speicheradressen, sowie vollständiges ASLR, mit komplett zufälligen Speicherbereichen. Voraussetzung für vollständiges ASLR ist die Erstellung der Applikation als Position Independent Executable (PIE) durch Xcode. Mit iOS 5 als Zielplattform stellt dies den Standard dar. Unter iOS 4.3 muss hingegen die Linker-Option *-fPIE* bei der App-Erstellung gewählt werden. Da in Objective-C auch low-level C-Funktionen verwendet werden können, welche zu Speicherüberläufen neigen, sollten diese vermieden werden. Dies können von Hackern ausgenutzt

werden, um Schadcode in das System zu schleusen. Hierzu zählen Stringfunktionen wie *printf*, *strcpy* oder *strcat*. Es wird empfohlen möglichst high level zu arbeiten und die angebotenen Funktionen der Cocoa API zu verwenden.
Die Validierung von Benutzereingaben sollte auch bei der App-Entwicklung berücksichtigt werden. Bei der Nutzung von SQLite-Datenbanken sollten beispielsweise Prepared Statements und Escape Strings eingesetzt werden, um SQL-Injections zu vermeiden. Bei einer SQL-Injection wird absichtlich ein SQL-Statement wie „DROP TABLE“ verwendet, um die App zu sabotieren.
Bei Zugriffe auf Web-Services und Internet-Inhalten bieten die Klassen NSURLRequest, NSURLConneciton und NSURLDownload eine verschlüsselte Kommunikation auf Basis von https. Es ist darauf zu achten, dass die Zertifikatsprüfung, die als Antwort zurückgeliefert wird, nicht übergangen wird. Dies ist oft in der Entwicklungsphase und bei Tests der Fall.

4.4.3 Windows Phone 7

Microsofts Marketplace ist die einzige Bezugsquelle an Apps für das Windows Phone 7. Für Entwickler von Apps stellt Microsoft einige Hürden bereit, die erst einmal überwunden werden müssen. Für die Einstellung der Apps in Marketplace muss sich der Entwickler einen Windows Live Account zulegen. Da Marketplace in den USA gehosted wird, ist es erforderlich, diverse Formulare auszufüllen und eine vom Einwohnermeldeamt beglaubigte Kopie des Reisepasses oder des Personalausweises mitzuschicken. Mit dieser ersten Sicherheitsmaßnahme wird verhindert, das anonyme Entwickler schadhafte Software entwickeln können. Als nächstes führt Microsoft nach Einstellen einer App eine Zertifizierung durch. Dies kann in der Regel 5 Tage dauern. Erst dann wird die App zum Download freigegeben. Microsoft gibt klare Richtlinien [42] zur Entwicklung von Apps. Nicht alle Richtlinien sind zwingend erforderlich. Zum Beispiel soll sich das Design an den Metro-Style anlehnen, was jedoch nicht notwendig ist, um die Applikation zu vertreiben. Andere Richtlinien, wie zum Beispiel die maximale Speichernutzung von 90MB sind Pflicht.
Windows Phone 7 Apps laufen in ihrer eigenen Sandbox und können somit nicht auf andere Anwendungen zugreifen. So kann zum Beispiel der neu entwickelte mobile Internet Explorer keine schadhafte Software auf Internet Seiten ausführen, dadurch wird das Risiko beseitigt, sich Malware auf dem Smartphone zu installieren. Zudem besitzt eine App ihren eigenen Speicherbereich und andere Apps haben ebenso keinen Zugriff auf die Daten wie die App selbst keine Möglichkeit hat, ihre Daten an einem andern Ort zu speichern als in ihrer eigenen Umgebung.

4.4.4 webOS

Für webOS gibt es den offiziellen HP App Catalog. Alle im HP App Catalog gelisteten Apps sind offiziell von HP auf Konformität gegenüber den Richtlinien (`http://www.hpwebos.com/us/company/app-tc.html`) des HP App Catalogs überprüft. Apps lassen sich leicht außerhalb des App Catalogs vermarkten, indem der Link zur App verteilt wird. Allerdings landet der Benutzer beim Aufruf des Links wieder im offiziellen App Catalog. Über Links lassen sich auch Vorabversionen von Apps verteilen, die noch nicht offiziellen gelistet werden.

Zusätzlich gibt es die Möglichkeit ungetestete Anwendungen zu installieren. So zum Beispiel Apps, die nicht im offiziellen HP App Catalog enthalten sind. Dazu muss allerdings auf dem Gerät der Developermodus aktiviert werden und eine Software auf dem Rechner installiert werden. Damit lässt sich auch der alternative Market „Preware“ installieren. Dieser enthält viele Programme und Hacks, die es nicht in den offiziellen App Catalog schaffen.

Alle webOS-Apps werden in getrennten Umgebung ausgeführt. Somit wird der unberechtigte Zugriff auf Daten und Dienste anderer Apps verhindert. Die Kommunikation von Apps untereinander kann nur über bereitgestellte APIs und Dienste erfolgen.

4.4.5 Blackberry

Für BlackBerry gibt es wie bei Android Geräten verschiedene Bezugsquellen. Zu nennen sind die zwei größten Plattformen „MobiHand App Store“ und „BlackBerry Appworld“.

Seit dem 1. April 2009 führte die Firma RIM den Onlineshop namens „BlackBerry Appworld“ ein [57]. Zunächst war dieser Dienst nur in der USA, Kanada und Großbritannien verfügbar, aber mittlerweile ist er auch in Deutschland erreichbar. Die Software steht zum kostenlosen Download auf der offiziellen BlackBerry Website zur Verfügung und kann nur auf Geräten mit Trackball oder Touchoberfläche installiert werden. Zusätzlich muss eine aktuelle Firmware ab der Version 4.2 oder höher installiert sein um diesen Dienst zu nutzen. Die Bezahlung erfolgt ausschließlich per PayPal und anders als bei anderen Plattformen behält BlackBerry nur 20% des erzielten Verkaufspreises ein. Neben vielen kostenlosen Tools liegt die unterste Preisgrenze bei 2,99 USD. Am 13. Juli 2011 meldete die Firma „Research In Motion“ 1 Milliarde Downloads. Was den Umfang an Angeboten und der Downloadzahlen angeht, hinkt aber RIM weit hinter seinen Konkurrenten hinterher.

Der „MobiHand App Store“ [45] ist kein offizieller App Store von RIM, war allerdings schon früher auf dem Markt als die „BlackBerry Appworld“. Der MobiHand App Store kann direkt mit dem Gerät heruntergeladen werden. Alternativ kann man sich den Link per E-Mail schicken lassen. Die App unterstützt alle BlackBerry-Modelle – mit oder ohne Touchscreen. Für die Nutzung ist ein Konto bei MobiHand nötig. Neben einzelnen Programmen ist MobiHand vor allem auch eine Fundgrube für BlackBerry-Themes. Bezahlt werden die Apps per PayPal oder mit diversen Kreditkarten.

Es gibt noch weitere Bezugsquellen wie BBB:OTA, CrackBerry oder Handago, die aber im Prinzip die gleichen Möglichkeiten bieten wie der MobiHand App Store.

4.5 Sicherheit drahtgebundener Verbindungen

Die folgenden zwei Kapitel beschäftigen sich mit der Sicherheit von eingehenden und ausgehenden Verbindungen, die mit einem Smartphone aufgebaut werden können. Die grundsätzliche Sicherheit dieser Verbindungen beruht auf den Sicherheitsmechanismen, die im verwendeten Übertragungsstandard festgelegt sind, und sind somit für Smartphones aller Plattformen gleich. Ausnahmen wie vereinzelte Sicherheitslücken in der Umsetzung beziehungsweise Implementierung dieser Richtlinien auf der jeweiligen Plattform werden hier nicht explizit betrachtet.

4.5.1 Universal Serial Bus (USB)

USB steht für Universal Serial Bus und stellt die einzige drahtgebundene Schnittstelle moderner Smartphones dar. Dies begründet sich vor allem in der kompakten Bauform sowie der hohen erzielten Mobilität. Der USB dient dem Nutzer zur Stromversorgung und dem Datenaustausch über ein Datenkabel (USB-Kabel). Ein separater Stromanschluss ist somit überflüssig, da die 5 Volt Spannung des USB-Anschlusses zur Ladung der relativ kleinen verbauten Akkus ausreichend ist. Anschlüsse, die dem USB-Standard genügen, gibt es in den verschiedensten Formen und Größen. Der Einfachheit halber einigte sich die Smartphoneindustrie auf den platzsparenden Micro-USB Anschluss, welcher über ein Adapterkabel an den PC oder an ein USB-Netzteil angeschlossen werden kann.

Um Daten per USB zu übertragen gibt es grundsätzlich zwei Ansätze:

USB-Massenspeicher (MSC) Das Gerät gibt sich bei Verbindung mit dem Rechner als Massenspeicher aus, somit kann direkt auf die Daten zugegriffen werden. Hierbei ist kein Zugriff auf Daten von Apps oder des Systems möglich, es kann nur auf einen begrenzten Speicherbereich zugegriffen werden (zum Beispiel auf die SD-Karte oder den internen Speicher). Als Treiber kommen meistens die Standard

Systemtreiber für Massenspeicher zum Einsatz, manchmal werden jedoch zusätzliche Treiber für das Gerät benötigt. Der Zugriff auf das Gerät ist nur möglich, wenn der Speicher auf dem Gerät zuvor explizit freigegeben wurde. Diese Variante kommt vor allem auf Android-basierten Geräten, webOS Geräten und BlackBerry OS Geräten zum Einsatz, um eine hohe Kompatibilität und Freiheit zu gewährleisten. Allerdings steigt mit dieser Freiheit auch die Gefahr schädliche Dateien auf das Gerät zu laden, weshalb der Benutzer genau wissen sollte, welche Daten er transferiert.

Zusatzsoftware über das Media Transfer Protokoll Auf die Daten des Geräts kann nur über zusätzliche Software und Treiber zugegriffen werden. Diese übernimmt für gewöhnlich die automatische Synchronisation der Daten und verhindert, dass beliebige, eventuell schädliche Dateien auf das Gerät verschoben werden können oder die interne Dateistruktur des Gerätes verändert wird. Folgende Systeme nutzen diese Art der Übertragung: iOS über iTunes, webOS, Phone 7 über Zune und BlackBerry über den BlackBerry Desktop Manager. Manche Hersteller von Android-Geräten setzen ebenfalls auf eine kontrollierte Schnittstelle, um unter anderem Firmwareupdates zu vereinfachen, wie beispielsweise Samsung mit ihrer Software Kies.

Für jede Plattform gibt es diverse Möglichkeiten auf einen der nicht verfügbaren Modi umzusteigen. Beispielsweise kann ein iPhone mittels Jailbreak vom iTunes-Zwang befreit werden.

Des weiteren bieten einige Plattformen einen Modus für Entwickler, der mehr Rechte gewährt und einen tieferen Eingriff in das System ermöglicht. Dieser muss explizit auf dem Gerät freigeschaltet werden. Android bietet beispielsweise die Möglichkeit, einen USB Debuggingmodus einzuschalten. Mit eingeschaltetem Debuggingmodus ist es möglich, Apps ohne Bestätigung auf dem Gerät zu installieren. Dies erleichtert das Entwickeln und Testen von Apps. Ist diese Funktion jedoch im Normalbetrieb aktiviert, stellt diese ein Sicherheitsrisiko dar. Wird das Gerät entwendet, so können Angreifer leichter Schadsoftware einschleusen und auf Systemfunktionen zugreifen.

4.6 Sicherheit drahtloser Verbindungen

Aufgrund der nötigen Mobilität finden sich auf Smartphones heutzutage eine hohe Anzahl an drahtlosen Verbindungen. Diese ermöglichen den schnellen und bequemen Beitritt zu den verschiedensten Netzwerken – seien es private Kleinstnetzwerke hergestellt über Bluetooth, Firmennetzwerke per WLAN oder das mobile Internet über den Tarifanbieter.

Alle diese Netzwerke kommunizieren per Funk über ein öffentliches Medium – die Luft. Der Schutz eines physischen Leiters fehlt komplett. Somit sind sie alle grundsätzlich anfällig für folgende Arten von Störungen und Angriffen:

- **Abhörbarkeit:** Ungewollter Mitschnitt von Daten durch Dritte
- **Unerlaubter Zugang:** Eindringen eines Angreifers oder Spions in ein sensibles Netz
- **Übertragungsstörungen:** Gewollte Maßnahmen zur Störung bestimmter Funkkanäle

Meistens handelt es sich bei Straftaten wie Datendiebstahl um die beiden erst genannten Kategorien. Dies liegt auch daran, dass eine Störung für einen Täter selten einen hohen Nutzen hat. Zudem möchte der Täter möglichst unbemerkt an seine Beute kommen, gewollte Störungen erregen jedoch Aufsehen.

4.6.1 Bluetooth

Bluetooth zählt zu den Personal Area Networks (PAN) und dient dem Aufbau eines Ad-Hoc Netzwerkes mit geringer Reichweite von mehreren Metern. Es soll als USB-Ersatz dienen und ermöglicht die kabellose, automatische Verbindung von Smartphones mit einem Computer, Laptop oder anderem Mobilgerät. Als Anwendungsfälle lassen sich beispielsweise Bluetooth-Headsets, Dateiaustausch oder Verbindungsbrücken

für lokale Netzwerke nennen. Durch den Ad-Hoc Charakter ist Bluetooth vor allem anfällig für geklonte Geräteidentitäten. Die Informationen könnten beispielsweise über einen Man-In-The-Middle Angriff, siehe Unterabschnitt 5.3.4, Man In The Middle, in Erfahrung gebracht werden. Als einfachster Schutz kann somit die relativ geringe Reichweite von 10 Metern verwendet werden, um potentiellen Angreifern fern zu bleiben. Des Weiteren sollten unbenutzte Bluetooth-Module ausgeschaltet werden. Dies verhindert zudem den Missbrauch von Bluetooth zur Aufzeichnung von Benutzerprofilen, welche auf der eindeutigen Geräte-ID beruhen und den Datenschutz des Nutzers gefährden. Das BSI gibt in [17] die generelle Schutzmaßnahmen zur Verwendung von Bluetooth in den Versionen 1.1 und 1.2 an. Ähnliche Maßnahmen wurden ebenso in dem aktuelleren Artikel [63] beschrieben und deshalb hier festgehalten:

Kein Standby Wird das Bluetooth-Modul nicht verwendet, so sollte es deaktiviert werden.

Menschenmassen meiden In sehr belebten Plätzen sollte auf die Verwendung von Bluetooth verzichtet werden.

Sichere PIN Zu Verbindungszwecken sollte eine möglichst sichere PINmit mindestens 8 Zeichen gewählt werden.

Erneute Authentifizierung Wird bei laufender Verbindung eine erneute Authentifizierung angefordert, so kann dies ein Zeichen für einen Man-In-The-Middle Angriff durch eine gewollte Störung sein. Der Verbindungsaufbau sollte deshalb sicherheitshalber abgebrochen werden.

Bekannte Geräte Ein Verbindungsaufbau sollte nur mit bekannten Geräten erfolgen.

Empfangsbereich sichern Stellen sie sicher, dass sich keine fremden oder unbekannten Geräte im Empfangsbereich aufhalten. Dies kann beispielsweise über einen ausreichenden Radius, oder private Bereiche gewährleistet werden.

Automatische Verbindungen meiden Unterbinden sie wenn möglich den automatischen Verbindungsaufbau mit bekannten Geräten.

4.6.2 WLAN

Der IEEE Standard 802.11 beschreibt in seinen Ausführungen a,b,g und n den Aufbau von Wireless Local Area Networks mit verschiedenen Bandbreiten, auch bekannt als WLAN. Zur Sicherung des gemeinsam genutzten Funkkanales existieren die folgenden drei Verschlüsselungsverfahren:

- Wired Equivalent Privacy (WEP), 1998
- Wi-Fi Protected Access (WPA), 2003
- Wi-Fi Protected Access 2 (WPA2), 2004

Alle drei Verfahren werden von heutigen Smartphones unterstützt. Das BSI bietet auf [6] einen Überblick über die Verschlüsselungsverfahren. Diese wurden in Tabelle 4.1 zum Vergleich festgehalten.

Des Weiteren existieren für 802.11 auch verschiedene Mechanismen zum Schutz von WLAN-Hotspots, unter Anderem:

- Das Verbergen der SSID bei einer Broadcast-Anfrage
- Filterung der MAC-Adressen von Teilnehmern

Beide Ansätze zielen darauf ab, einen vorhandenen Zugangspunkt vor unbefugten Dritten zu schützen. Da es sich bei Smartphones jedoch in der Regel um WLAN-Clients handelt, werden diese Punkte nicht weiter ausgeführt. Es sei jedoch erwähnt, dass die Filterung von MAC-Adressen nur einen hinlänglichen Zugangsschutz bietet, da ein Angreifer die MAC-Adressen von authentisierten Clients abfangen und seine eigene mit diesen ersetzen kann.

Standard	WEP	WPA	WPA2
Verschlüsselungs-Algorithmus	RC4	RC4	AES
Schlüssellänge	40 / 104 Bit	128 Bit (64 Bit bei der Authentisierung)	128 Bit
Schlüssel	statisch	dynamisch (PSK)	dynamisch (PMK)
Initialisierungsvektor	24 Bit	48 Bit	48 Bit
Datenintegrität	CRC-32	MICHAEL	CCMP

Tabelle 4.1: Verschlüsselungsstandards

Wired Equivalent Privacy

Das älteste Verschlüsselungsverfahren ist WEP, welches im Standard 802.11 beschrieben ist. WEP basiert auf dem veralteten Temporary Key Integrity Protocol, kurz TKIP, und nutzt zur Verschlüsselung den Algorithmus RC4. Es bietet laut [6] keinen ausreichenden Schutz und kann von Hackern durch Brute-Force Ansätze leicht geknackt werden. Das Bundesamt für Sicherheit in der Informationstechnik, kurz BSI, rät von der Verwendung von WEP ab. Somit sollten Smartphone Nutzer Netze mit schwacher WEP-Verschlüsselung meiden. Verwendet ein Hotspot WEP, so rät das BSI dem Betreiber, die maximale Schlüssellänge zu verwenden, und den Zugangsschlüssel regelmäßig, mindestens einmal täglich, auszutauschen. Wegen der Sicherheitsmängel von TKIP, dürfen Access Points laut [15] dieses ab 2011 nicht mehr unterstützen. Dies gilt ab 2012 für alle WLAN-Geräte.

Wi-Fi Protected Access

Auf Grund der gravierenden Sicherheitsmängel von WEP, wurde 2003 das WPA-Verfahren entwickelt. Es galt als Übergangslösung zum Standard 802.11i, der sich noch in der Entwicklung befand, und sollte die groben Sicherheitslücken von WEP beheben. Daher basiert es, ebenso wie WEP, noch auf TKIP und nutzt RC4 zur Verschlüsselung. Allerdings wurde WPA mit einer verbesserten Schlüsselberechnung versehen. Da das Verfahren TKIP nutzt, wird es jedoch nicht mehr für den sicheren Einsatz empfohlen.

Wi-Fi Protected Access 2

WPA2 basiert als erstes Verfahren auf dem Sicherheitsstandard 802.11i, der Schlüssel der Größe 128 Bit verwendet. Diese lassen sich selbst durch heutige Computer nicht in ausreichend kurzer Zeit durch Brute-Force-Ansätze knacken. Zudem basiert es auf dem Advanced Encryption Standard, kurz AES. Mit der Weiterentwicklung WPA2 ist somit ein ausreichendes Maß an Sicherheit für Verbindungen gegeben. Wie bei allen Schlüssel-abhängigen Sicherheitsmechanismen gilt, dass der Pre-Shared-Key, kurz PSK, möglichst lang und mit hoher Zeichenvarianz versehen werden sollte, um größtmögliche Sicherheit zu gewährleisten. Mit der Abschaffung von TKIP dürfen Access Points ab 2014 nur noch WPA2-AES anbieten.

Somit sollte jeder Nutzer eines Smartphones, der sich in ein WLAN-Netz einbucht, auf die WPA2 Verschlüsselung achten. Vor allem eigene und Firmennetzwerke sollten mit diesem Standard gesichert werden. Nutzt er ein schwach verschlüsseltes, oder gar offenes öffentliches Netz, so sollte die Verwendung einer verschlüsselten Verbindung über HTTPS zum Einsatz kommen, oder das Netz gemieden werden.

4.6.3 Mobilfunk

Heutige Mobilfunknetze in Europa setzen nach [13] auf die digitalen Übertragungsstandards GSM (Global System for Mobile Communications) und UMTS (Universal Mobile Telecommunications System). Letzteres

wurde weitestgehend ausgebaut, um den mobilen Datentransfer zu beschleunigen und so die Datendienste von GSM in unterstützten Gebieten zu ersetzen, sowie Sprachübertragungsengpässe in GSM zu beseitigen. GSM dient jedoch als Rückfallebene, falls keine schnelleren Verbindungen verfügbar sind. Man spricht bei GSM von einem Netz der zweiten Generation, dem ersten digitalen Funknetz. UMTS stellt ein Netz der dritten Generation dar. Neuste Netze basierend auf LTE (Long Term Evolution), die sich derzeit noch im Ausbau befinden, werden als vierte Generation bezeichnet. Im Folgenden werden deshalb jedoch nur die Netze der zweiten und dritten Generation, sowie deren Übertragungsstandards, betrachtet.

Datenübertragungsstandards

Es existieren pro Netz verschiedenste Standards zur Paket-orientierten Datenübertragung, um ein breitbandiges mobiles Netz zu schaffen, welches auch als Metropolitan Area Network (MAN) bezeichnet wird. Hierzu zählen, unter Anderem, GPRS und EDGE, basierend auf GSM, sowie HSDPA, basierend auf UMTS. GPRS, das immer noch als Rückfallebene in weniger gut abgedeckten Gebieten verwendet wird, wird hierbei als weniger sicher eingestuft, als Pendants im UMTS-Netz. Laut [52] wurde die Verschlüsselung von GPRS bereits durch eine Firma mit Sitz in Berlin geknackt. Der Firma ist es gelungen, GPRS-Daten in einem Umkreis von 5 Kilometern abzufangen und somit auch unverschlüsselten E-Mail-Verkehr abzuhören. In diesem Zusammenhang wurde der Verschlüsselungsalgorithmus GPRS-A5 kritisiert, der in Deutschland Anwendung findet. In anderen Ländern, wie beispielsweise Italien, würden jedoch noch schwächere Verschlüsselungen verwendet, oder ganz darauf verzichtet.

Somit stellen besonders die älteren, leistungsschwächeren Datenübertragungsstandards ein potentielles Sicherheitsrisiko dar. Smartphones lassen jedoch nur selten eine Priorisierung der verwendeten Dienste zu – es wird, je nach Empfangsqualität, das Protokoll verwendet, welches die beste Verfügbarkeit garantiert. Anwender des mobilen Webs sollten deshalb darauf achten, Internetaktivitäten über das HTTPS-Protokoll durchzuführen, durch den der komplette Datenverkehr, unabhängig vom verwendeten Netzwerk, verschlüsselt wird. Dies gilt insbesondere für das Versenden vertraulicher Informationen, zum Beispiel bei E-Mails oder dem Online-Banking.

Aus Sicht des Datenschutzes ist das Surfen über UMTS und HDSPA sicherer, als die Nutzung eines W-LAN Hotspots, da ein Zwangsproxie vor den eigentlichen Datenverkehr geschaltet wird, durch den alle Daten getunnelt werden. Die Position des Nutzers ist somit nur vage zu ermitteln, eine genaue Ortung zum Missbrauch dieser Daten somit schwierig, siehe auf Abschnitt 5.5, Sicherheit von Ortungsdiensten.

IMSI-Catcher

Eine weitere Schwachstelle im GSM-Netz wurde durch so genannte IMSI-Catcher ausgenutzt. Hierbei setzt sich der Hacker zwischen Mobilteilnehmer und Netzanbieter und gibt sich gegenüber Ersterem als Basistation des Netzanbieters aus. Es handelt sich somit um eine Man-In-The-Middle Attacke, siehe Unterabschnitt 5.3.4, Man In The Middle. Hierdurch konnte der Angreifer sensible Daten der SIM-Karte ausspähen, unter Anderem die IMSI (International Mobile Subscriber Identity), durch die eine Kopie der SIM-Karte für eigene Zwecke möglich wurde. Deshalb wurde bei der Entwicklung von UMTS ein weiteres, so genanntes Authentication Token, kurz AUTN, eingeführt. Somit kann das Handy noch vor den Authentisierungs- und Verschlüsselungsprozeduren prüfen, ob es sich tatsächlich um den angegebenen Netzbetreiber handelt, um anschließend die Teilnehmerauthentisierung zu starten. Der geheime Teilnehmer-Identitätsschlüssel ist bei UMTS nur im Authentication Center sowie auf der SIM-Karte hinterlegt und wird nicht transportiert, was ein Ausspähen verhindert. UMTS ist somit in diesem Punkt sicherer als GSM und sollte bei aktiven Internetverbindungen bevorzugt werden. [65]

Neue Angriffe auf GSM-Handys

Im Dezember 2011 wurden auf dem 28. Chaos Communication Congress (28C3) von Sicherheitsforschern neue Angriffsmöglichkeiten auf das GSM-Netz vorgestellt. Der Angriff beruht auf den, im Vorjahr vorgestellten Schwachstellen, des noch eingesetzten A5/1-Verschlüsselungsalgorithmus. Über diese lassen sich Gespräche minutenschnell entschlüsseln und mitschneiden. Hierzu müssen die Temporary Mobile Subscriber Identity (TMSI), sowie der verwendete geheime Schlüssel bekannt sein. In einem weiteren Schritt sei es nun möglich, ein GSM-Handy auf Softwarebasis zu simulieren und den Versand teurer Premium-SMS, sowie die Nutzung kostenpflichtiger Telefonate, zu initiieren. Somit seien vereinzelte Handy-Rechnungen erklärbar, die aus bislang unbekannten Gründen Kontakt zu Premium-Diensten auf den Karibischen Inseln aufgenommen hätten, und so Schäden im dreistelligen Bereich verursachten. Der Angriff ermögliche ebenfalls das Abhören der Mailbox des Opfers, sofern dessen Standort bekannt ist.

Sicherheitsexperte Karsten Nohl fordert laut [37] somit nachdrücklich die Umsetzung des Verschlüsselungsstandards A5/3, der die Gefahr von großflächigen Angriffen reduzieren soll. Nach der Implementierung durch die Mobilfunkbetreiber seien Handy-Hersteller hierbei der entscheidende Faktor, da diese den Algorithmus in ihren Geräten ebenso bereitstellen müssen. Laut Nohl verzögere sich dies jedoch durch einen einzelnen Hersteller, dessen Geräte den Algorithmus bis dato noch nicht beherrschen und somit weitere Tests blockieren.

4.6.4 Near Field Communication

Der international anerkannte Übertragungsstandard Near Field Communication(NFC) dient zum kontaktlosen Austausch von Daten über kurze Distanzen. NFC gehört somit zu den Short-Range Wireless Networks. NFC baut im Gegensatz zu Bluetooth oder WLAN eine Point-to-point-Verbindung auf. Es können also nur zwei Geräte miteinander kommunizieren.

Folgende Dienste können beispielsweise über NFC realisiert werden:

- Austausch von Dateien
- elektronische Visitenkarte
- mobiles Bezahlen
- mobile Multiplayerspiele

Eine Verbindung zwischen zwei Geräten wird aufgebaut, indem diese aneinader gehalten werden. Da NFC nur über eine kurze Distanz funktioniert, gibt es keine Nutzerinteraktion am Gerät, um eine Verbindung zu bestätigen. Die reine Nähe zweier Geräte wird als Zustimmung des Nutzers, eine Verbindung zum anderen Gerät aufbauen zu wollen, gewertet. NFC bietet eine maximale Bitrate von 424 kbit/s und ist somit langsamer als Bluetooth. Die Verbindung über NFC kann sowohl zwischen zwei aktiven Geräten (zum Beispiel Austausch von Kontaktdaten zwischen zwei Telefonen) oder zwischen einer aktiven und einer passiven Komponente (ähnlich RFID) erfolgen. Die Geräte können sich gegenseitig authentifizieren. NFC kann auch dazu dienen, schnellere Datenverbindungen (Bluetooth, WLAN) zu initiieren. Dadurch werden die Vorteile beider Verbindungen kombiniert. NFC bietet einen schnellen und einfachen Verbindungsaufbau und Bluetooth bzw. WLAN bieten eine schnelle Datenverbindung.

Das Konzept ist vorwiegend in Japan verbreitet und wird zum Bezahlen kleinerer Beträge verwendet.

4.7 Sicherheit in Unternehmen

Viele Unternehmen gehen derzeit nach und nach dazu über, ihren Mitarbeitern anstatt eines herkömmlichen Mobilfunkgerätes ein Smartphone bereitzustellen. Doch nicht alle Unternehmen setzen firmeneigene Geräte

ein. In vielen Betrieben heisst es: „Bring your own Device“. Was aus Arbeitnehmergesichtspunkten natürlich gut ist, da man nur ein Gerät mit sich führen muss und keine Umstellung auf neue Geräte stattfindet, bringt aus Sicht des Arbeitgebers jedoch Probleme im Hinblick auf Datenschutz und IT-Sicherheit in Unternehmen mit sich. Es ist schwieriger, die Sicherheitsmaßnahmen umzusetzen und auf dem Smartphone zu installieren. Da ein Smartphone ein Hybrid aus Mobilfunkgerät und PC darstellt, ergeben sich insoweit zum Teil die gleichen Probleme, genannt sei an dieser Stelle beispielsweise die private Internetnutzung mittels betrieblicher PCs oder Smartphones. Gleichwohl ergeben sich auch weitere Herausforderungen, welche über den Betrieb herkömmlicher Mobilfunktelefone hinausgehen. Wichtigstes Sicherheitsmanko ist zum Beispiel der E-Mailverkehr. Hier werden die meisten Unternehmensinformationen verschickt oder empfangen. Darum müssen solche Geräte, da sie sich meistens auch im privaten Besitz der Mitarbeiter befinden, sehr genau an die unternehmensspezifischen Sicherheitsanforderungen halten. Das große Problem daran ist, dass die Unternehmen einen einheitlichen Standard einführen müssen, damit jeder Mitarbeiter seine Daten gleichermaßen schützen muss.

Laut einer Umfrage der Sicherheitsfirma CheckPoint[48] sind Apples iOS Geräte mit 30% am häufigsten in Firmennetzwerken vertreten. Gefolgt von BlackBerry OS mit einem Anteil von 29%, Android mit 21% und Windows Mobile/Phone mit 18%. Die befragten IT-Mitarbeiter stufen Android als das System mit dem größten Sicherheitsrisiko ein und halten unabhängig vom Betriebssystem die Nutzer für eine größere Bedrohung der Sicherheit als Hacker.[28]

Im folgenden werden hier die verschiedenen Plattformen beschrieben:

4.7.1 Android

Mit Android 2.2 führte Google das Android Device Administration API ein. Dieses erlaubt es, Anwendungen zu erstellen, die folgende Sicherheitsrichtlinien für das Gerät vorschreiben:

- Bildschirmsperre mit Passwort
- Passwortlänge, minimale Anzahl an Zahlen, Groß-/Kleinbuchstaben
- Gültigkeit von Passwörtern
- Passworthistorie, die verhindert, dass immer wieder das selbe Passwort verwendet wird
- Anzahl an Fehleingaben des Passworts, bevor das Gerät zurückgesetzt wird
- Sperrung des Geräts nach einer bestimmten Zeit
- Verschlüsselung der Daten

Zusätzlich ist es möglich, das Gerät aus der Ferne zurückzusetzen. Dies bedeutet, dass alle Daten auf dem Gerät gelöscht werden und es sich wieder im Auslieferungszustand befindet.

Um Androidgeräte in Netzwerke einzubinden werden verschiedene VPN-Dienste (PPTP mit Shared Secret, L2TP und L2TP/IPSec, entweder mit Zertifikat oder Shared Secret), die Verbindung per SSL und TTL, sowie die Verbindung zu verschlüsselten WLANs per WPA und WPA2 (Enterprise) verwendet.

4.7.2 iOS

Es gibt eine große Anzahl an Apps im App-Store für das iPhone, welche für die Benutzung im Unternehmen vorgesehen sind. Aber auch mit den vorinstallierten Apps auf dem iPhone lässt sich das Gerät für den Einsatz im Unternehmen einrichten. Die vier Sicherheitskonzepte des iPhones sind:

Gerätesicherheit Methoden, die die unbefugte Nutzung des Geräts verhindern

Datensicherheit Schutz gespeicherter Daten bei Diebstahl oder Verlust des Geräts

Netzwerksicherheit Netzwerkprotokolle und Datenverschlüsselungen bei der Übertragung

Programmsicherheit Sichere Plattformgrundlage mit iOS

Gerätesicherheit

iOS verwendet einen eindeutigen, von jedem Benutzer festgelegten Code, um einen sicheren Schlüssel zur Datenverschlüsselung zu generieren, durch den E-Mails und vertrauliche Programmdaten auf dem Gerät zusätzlich geschützt werden. Zusätzlich bietet das iPhone sichere Methoden zur Konfiguration des Gerätes in einer Unternehmensumgebung, in der bestimmte Einstellungen, Richtlinien und Einschränkungen erzwungen werden. Ein Gerätecode verhindert, dass Unbefugte auf die Daten auf dem iPhone zugreifen, oder auf andere Weise Zugriff auf das Gerät erhalten. Die wichtigsten werden hier kurz aufgeführt:

- Zwingende Eingabe eines Gerätecodes
- Einfachen Wert zulassen
- Alphanumerischen Wert zulassen
- Mindestlänge für den Code
- Mindestanzahl komplexer Zeichen
- Maximale Geltungsdauer des Codes
- Automatische Sperrung
- Frist für Gerätesperre
- Maximale Anzahl fehlgeschlagener Versuche

Richtlinienumsetzung Die oben beschriebenen Richtlinien können auf dem iPhone auf mehrere Arten festgelegt werden. Richtlinien können als Teil eines Konfigurationsprofils für den Benutzer installiert werden. Ein Profil kann so definiert werden, dass das Löschen nur mit einem Administratorkennwort möglich ist, oder dass es auf dem Gerät gesperrt ist und nicht ohne vollständiges Löschen aller Geräteinhalte entfernt werden kann. Zusätzlich können Codeeinstellungen mithilfe von Mobile Device Management-Lösungen, kurz MDM, durch die Richtlinien im Push-Modus direkt an das Gerät gesendet werden. Auf diese Weise können Geräte über ein Netzwerk konfiguriert werden, und Richtlinien können ohne Interaktion durch den jeweiligen Benutzer umgesetzt und aktualisiert werden. Zur manuellen Konfiguration von Richtlinien bietet Apple das „iPhone Configuration Utility“ zum freien Download an.

Sichere Gerätekonfiguration Konfigurationsprofile werden in XML-Dateien gespeichert. Sie enthalten Gerätesicherheitsrichtlinien und -einschränkungen, VPN-Konfigurationsdaten, WLAN-Einstellungen, E-Mail- und Kalender-Accounts, sowie Authentifizierungsdaten. Die Möglichkeit, Coderichtlinien zusammen mit Geräteeinstellungen in einem Konfigurationsprofil festzulegen, stellt sicher, dass die Geräte im Unternehmen korrekt und gemäß den Sicherheitsstandards des Unternehmens konfiguriert sind. Da sich Konfigurationsprofile sowohl verschlüsseln als auch sperren lassen, können die Einstellungen nicht entfernt, verändert oder weitergegeben werden. Konfigurationsprofile können signiert und verschlüsselt werden. Das Signieren stellt sicher, das die so umgesetzten Einstellungen nicht verändert werden können. Das Verschlüsseln schützt die Inhalte und erlaubt die Installation nur auf dem vorgesehenen Gerät. Konfigurationsprofile werden über CMS (Cryptographic Message Syntax, RFC 3852) mit 3DES- und AES-128-Unterstützung verschlüsselt. Bei der Erstübertragung verschlüsselter Konfigurationsprofile müssen diese per USB-Synchronisation mit dem iPhone Configuration Utility oder per kabelloser Registrierung übertragen werden. Zusätzlich kann die Verteilung verschlüsselter Konfigurationsprofile per E-Mail Anhang, über eine Website, auf die nur Benutzer Zugriff haben, oder mithilfe der Push-Funktion von Mobile Device Management-Lösungen erfolgen.

Geräteeinschränkungen Geräteeinschränkungen bestimmen, auf welche iPhone Funktionen der Benutzer auf dem Gerät zugreifen kann. Normalerweise handelt es sich dabei um netzwerkfähige Programme wie Safari, YouTube und den iTunes Store, aber auch Funktionen wie die Installation von Programmen oder die Verwendung der Kamera können eingeschränkt werden. Mit Geräteeinstellungen kann das Gerät für den jeweiligen Bedarf gemäß der Unternehmensvorgaben eingestellt werden. Zusätzlich zum Festlegen von Einschränkungen und Richtlinien, kann die IT-Abteilung das iTunes Desktop-Programm konfigurieren und steuern. Hierzu gehört eine Zugriffssperre für ungeeignete und anstößige Inhalte, das Festlegen von Netzwerkdiensten, auf die der Benutzer innerhalb von iTunes zugreifen darf und das Erlauben von Softwareaktualisierungen.

Datensicherheit

Zusätzlich zur Datenverschlüsselung während der Übertragung unterstützt das iPhone für verbesserten Datenschutz die Hardwareverschlüsselung, für auf dem Gerät gespeicherte Daten und die zusätzliche Verschlüsselung von E-Mail und Programmdaten. Das iPhone 3GS und höher bieten eine hardwarebasierte Verschlüsselung. Zum Schutz aller Daten auf dem iPhone wird eine 256-Bit-AES Verschlüsselung verwendet. Die Verschlüsselung ist dauerhaft aktiviert und kann nicht durch den Benutzer deaktiviert werden. Basierend auf den Funktionen der Hardwareverschlüsselung des iPhones 3GS und neuerer Geräte, lassen sich auf dem Gerät gespeicherte E-Mails und Anhänge mit den in iOS4 integrierten Datenschutzfunktionen schützen. Hierfür wird der eindeutige Gerätecode jedes Benutzers, zusammen mit der Hardwareverschlüsselung des iPhones, verwendet, um einen sicheren Schlüssel zu generieren. Dieser Schlüssel verhindert den Datenzugriff, wenn das Gerät gesperrt ist, und stellt sicher, dass vertrauliche Daten vor Unbefugten geschützt sind. Die Daten werden hierbei in der so genannten Keychain abgelegt, siehe auch Unterabschnitt 5.2.3, Sicherheit der iOS Keychain.

Fernlöschen Das iPhone unterstützt die Funktion „Fernlöschen“. Wenn das Gerät verloren geht oder gestohlen wird, kann der Administrator oder Gerätebesitzer mit der Funktion „Fernlöschen“ alle Daten löschen und das Gerät deaktivieren. Siehe hierzu Abschnitt 5.7, Diebstahlschutz.

Lokales Löschen Geräte können auch für ein automatisches lokales Löschen nach mehreren falschen Codeeingaben konfiguriert werden. Diese Maßnahme bietet Schutz bei Überfällen, mit dem Ziel, Zugriff auf das Gerät zu erhalten. Ist ein Code festgelegt, kann der Benutzer das lokale Löschen direkt in den Einstellungen des iPhones aktivieren. Brute-Force Angriffe werden so verhindert, siehe Unterabschnitt 5.3.1, Brute-Force-Angriffe.

Netzwerksicherheit

Das iPhone kann über die Unterstützung von Cisco IPSec, L2TP und PPTP in eine Vielzahl gängiger VPN-Technologien integriert werden. Zusätzlich unterstützt das iPhone SSl-VPN über Programme von Juniper und Cisco. Die Unterstützung für diese Protokolle stellt sicher, dass das höchste IP-basierte Verschlüsselungsniveau für die Übertragung vertraulicher Informationen eingehalten wird.

SSL/TLS Das iPhone unterstützt SSL v3 und Transport Layer Security (TLS v1), den Sicherheitsstandard für das Internet. Safari, Kalender, Mail und weitere Internetprogramme starten diese Verfahren automatisch, um einen verschlüsselten Kommunikationskanal zwischen iPhone und Unternehmensdiensten zu bieten.

WPA/WPA2 Das iPhone unterstützt WPA2 Enterprise für den authentifizierten Zugriff auf das WLAN des Unternehmens. WPA2 Enterprise verwendet die 128-Bit-AES Verschlüsselung und bietet damit einen hohen Sicherheitsstandard. Durch die Unterstützung für 802.1X lässt sich das iPhone außerdem in eine Vielzahl von RADIUS-Authentifizierungsumgebungen integrieren.

Programmsicherheit

Der Sandbox Ansatz für den Programmlaufzeitschutz und die verbindliche Programmsignierung stellen sicher, dass Programme nicht verändert werden können. Die Programme auf dem Gerät werden in einem geschützten Bereich (Sandbox) ausgeführt, damit sie nicht auf von anderen Programmen gespeicherte Daten zugreifen können. Die Systemdateien, Ressourcen und der Kernel, sind vom Programmbereich des Benutzers abgegrenzt. Programmentwickler haben Zugriff auf die Verschlüsselungs-API „Common Crypto“, mit denen sie die Daten ihrer Programme noch besser schützen können, indem sie sie mit bewährten Methoden wie AES verschlüsseln. Zusätzlich bietet das iPhone eine Hardwarebeschleunigung für AES- und SHA1-Verschlüsselung. Auch Programme können die integrierte Hardwareverschlüsselung des iPhones nutzen, um sensible Programmdaten zusätzlich zu schützen. Entwickler können bestimmte Dateien als zu schützend kennzeichnen und so das System anweisen, den Inhalt dieser Dateien durch Verschlüsselung, sowohl für Programme, als auch für Unbefugte, unzugänglich zu machen, wenn das Gerät gesperrt ist. Näheres hierzu in Unterabschnitt 4.4.2, iOS. [5]

4.7.3 Windows Phone 7

Da sich Microsofts Betriebssystem Windows Phone 7 erst kurze Zeit auf dem Markt befindet, gibt es noch keine konkreten Sicherheitsvorkehrungen zum Einsatz in Unternehmen. Laut einem Pressesprecher von Microsoft konzentriere man sich mit Windows Phone 7 zunächst auf den Consumer Markt, da man dies mit der vorherigen Version Windows Mobile 6.5 versäumt habe. Man hoffe darauf, dass die Nutzer des Systems ihre Geräte dann sozusagen in die Firma tragen, und diese dort Verwendung finden.

Mit der neusten Version des Betriebsystems Windows Phone 7 Mango hat Microsoft an Lösungen gedacht, die das Verwenden im Unternehmen erleichtern. So fügte Microsoft dem neustem System „Outlook Mobile“ sowie „Office Mobile“ hinzu, welche zu 100 Prozent mit der Desktop Version kompatibel ist. Eine Synchronisation mit dem lokalen Outlook ist jedoch nicht möglich. Der Benutzer muss den Abgleich über das Internet vornehmen [34]. Dabei überträgt das Windows Phone 7 die Daten verschlüsselt per Secure Socket Layer (SSL), je nach Serververbindung mit 128 oder 256 Bit [12].

Anwendungen wie Exchange, SharePoint oder Lync werden von IT-Spezialisten verwendet um Daten und Kollegen miteinander zu vernetzen. Windows Phone 7 ist ein EAS-zertifiziertes (Exchange ActiveSync) Smartphone. Daher kann die IT-Abteilung einfach Richtlinien festlegen und verwalten, um Windows Phones direkt über Exchange oder über eine Reihe von Verwaltungstools von Drittanbietern zu verwalten. Alle Anwendungen werden in einem verwalteten Code (mit Microsoft Silverlight/XNA) entwickelt. Auf diese Weise werden Sicherheitsrisiken reduziert. Geschäftsanwendungen können die Unternehmensauthentifizierung nutzen und Daten und Anmeldeinformationen verschlüsseln, um Geschäftsinformationen zu schützen. Microsoft Rights Management Services verhindert den Zugriff auf Dokumente und E-Mails durch nicht berechtigte Personen. Als Benutzer können Daten klassifiziert und auf diese Weise geschützt werden [43].
Zudem hat man die Ortung des Geräts standardmäßig mit integriert damit gestohlene, verlorene oder nur verlegte Geräte sich wieder finden und gegebenenfalls sperren oder löschen können. Dies kann über jeden Rechner geschehen der mit dem Internet verbunden ist und man Zugriff auf die Seite: [47] hat.

4.7.4 webOS

Unter webOS werden die folgenden Sicherheitsaspekte für Unternehmen betrachtet:

Sichere Kommunikation webOS unterstützt die Übertragung von Daten per WLAN und Bluetooth. Für WLAN werden sowohl WPA mit TKIP-Authentifizierung, als auch WPA2 mit AES- und EAP-Authentifizierung. Wobei alle verbreiteten EAS-Typen unterstüzt werden: EAP-TLS, PEAP v1/v2,

EAP-TTLS, EAP-FAST und LEAP. webOS unterstützt die Sicherheitsfunktionen von Bluetooth 2.1. Daten können per Bluetooth erst ausgetauscht werden, wenn der Partner autorisiert und somit bekannt ist.

lokaler Datenschutz webOS-Geräte werden durch eine Authentifizierung beim Einschalten des Gerätes vor unberechtigtem Zugriff geschützt. Das Gerät bleibt solange gesperrt, bis ein gültiges Kennwort eingegeben wird. Die Kennwörter können entweder auf dem Gerät konfiguriert werden oder vom Unternehmen festgelegt werden. Hierbei kann die Mindestlänge, die Verwendung von Ziffern und Buchstaben, die Maximalzahl an fehlgeschlagenen Kennwortversuchen, sowie nach welcher Zeit der Inaktivität der Bildschirm gesperrt wird, festgelegt werden.

Anwendungsumgebung Durch die Abschottung der Apps in getrennte Umgebungen werden die Daten einer Anwendung vor unberechtigtem Zugriff geschützt.

Fernlöschung Geht ein Gerät verloren oder wird gestohlen, kann es vom Besitzer oder Administrator des Unternehmens aus der Ferne gelöscht werden. Das Gerät kann so eingestellt werden, dass wenn die Maximalzahl an fehlgeschlagenen Kennwortversuchen erreicht ist, das Gerät sich automatisch löscht.

Datensicherung Die Daten eines webOS-Gerätes können regelmäßig in die Cloud gesichert werden. So dass bei einem Verlust des Gerätes die Daten auf ein anderes Gerät wieder aufgespielt werden können und so die Ausfallzeit gering gehalten wird.

[46]

4.7.5 BlackBerry

Unternehmen die BlackBerry Geräte einsetzten, vor allem die Orginalgeräte der Firma RIM können und werden meist von einem Systemadministrator „Over the Air“ fern verwaltet. So ist es dem Administrator möglich, den Geräten neue Sicherheitsmechanismen oder Software aufzuspielen. Die wichtigste Sicherheitskomponente bei den BlackBerry Geräten ist die AES-Verschlüsselung (bei älteren Geräten ist es die 3DES Verschlüsselung) des gesamten Datenverkehrs. Dieser Schlüssel ist 256 Bit lang und wird bei der ersten Aktivierung des Geräts per Zufallsgenerator erstellt und ist genau dreißig Tage lang gültig und muss dann wieder neu erstellt werden. Der IT-Administrator hat nun die Möglichkeit diesen Schlüssel neu zu generieren. Auch der Benutzer kann durch einen Knopf diese Schlüsselgenerierung erzwingen. Der Schlüssel für die Datenübertragung wird nicht mit dem Masterkey verschlüsselt, sondern ein auf dem Masterkey beruhenden Sessionkey erstellt. Zu dem kann man den E-Mailverkehr mit dem Einsatz von S/MIME oder PGP verschlüsseln. Geht das Gerät verloren, kann der Systemadministrator das Gerät sperren und gegebenenfalls fern löschen falls das Gerät nicht wieder gefunden wird und die Daten nicht in die falschen Hände geraten dürfen. Auch der Geräteinhalt und nicht nur die Übertragung kann vom Administrator verschlüsselt werden. Hat der Benutzer kein Gerätepasswort erstellt, der das Benutzen des Geräts verhindert, kann auch dies über „Over the Air“ geschehen und erzwungen werden. Es stehen mehr als 400 verschiedene zentrale Einstellungen, den sogenannten „Policies“ für den Administrator bereit um das Gerät sicher vor Zugriffen zu schützen. Es können Sicherheitseinstellungen hinzugefügt und Geräteeigenschaften abgeschaltet werden. Die wichtigsten werden hier kurz erläutert:

1. Das erzwingen von sicheren Passwörtern mit einer Zeichenlänge bis 24 Zeichen mit Kleinbuchstaben, Großbuchstaben, Zahlen und Sonderzeichen.
2. Einschalten einer Passworthistorie, die n letzte Passwörter speichert und damit verhindert das gleiche Passwörter erneut gesetzt werden.
3. Sperren von bestimmten Passwörtern
4. Sperrung des Geräts nach einer bestimmten Zeit
5. Sperren und Löschen des Geräteinhalts nach einer bestimmten Fehleingabe des Gerätepassworts.

6. „Periodic Challenge“, bedeutet das nach einer bestimmten Zeit die Kennworteingabe erneut erfolgt auch wenn mit dem Gerät gearbeitet wird.
7. Das Sperren von SMS/MMS, Instant Messaging oder anderen E-Maildiensten damit nicht über diese Dienste vertrauliche Daten kommuniziert werden
8. Bei neueren Geräten kann auch die Kamera, Multimediale Dienste oder die externe Speicherkarte die sich im Gerät befindet, deaktiviert werden

All das ist aber nur möglich, wenn sich das Gerät in Reichweite eines Funknetzes(GPRS/UMTS) befindet.

4.8 Sicherheit von Geräten

Dieses Kapitel beschäftigt sich mit der Sicherheit von Geräten. Insbesondere möchten wir hier auf Verfahren eingehen, die die eingebauten Sicherheitssysteme aushebeln und dem Nutzer somit vielfältige neue Möglichkeiten bieten. Dies wird allgemein als Rooting bezeichnet, da der Nutzer die Root-Rechte am System erhält. Da dies vom Hersteller der Geräte nicht vorgesehen ist, geben wir keine genaue Anleitung. Rooting-Verfahren sind jedoch relativ populär, deshalb möchten wir in diesem Kapitel kurz darauf eingehen und auf entstehende Probleme und Sicherheitslücken hinweisen.

4.8.1 Rooten von Android-Geräten

Da Android auf Linux basiert, ist es prinzipiell möglich, Programme als Administrator mit Root-Rechten auszuführen. In der Standardkonfiguration ist dies jedoch nicht vorgesehen. Durch Lücken im System, oder durch einen offenen Bootloader, lässt sich diese Beschränkung umgehen. Danach lassen sich ausgewählte Programme mit root-Rechten ausführen. Durch root-Rechte lassen sich praktisch alle Sicherheitsvorrichtungen umgehen, wie zum Beispiel die Sandbox für Programme. Für gewöhnlich wird das System beim Rooten so angepasst, dass der Benutzer der Nutzung von root-Rechten einer App zunächst zustimmen muss, wenn sie diese anfordert.

Eine Anpassung des Systems ist möglich, da der Quellcode von Android frei verfügbar ist. Dadurch ist es auch möglich, das gesamte System nach belieben zu verändern und eine angepasste Version von Android zu installieren. Diese so genannten Custom ROMs bieten den Vorteil, dass Sicherheitsupdates häufig schneller als direkt vom Hersteller verfügbar sind (siehe Abschnitt 4.9, Systemupdates). Der Nachteil ist jedoch die fehlende Qualitätssicherung, so dass sich auch vermehrt Fehler einschleichen können.

4.8.2 Rooten von iOS-Geräten: Jailbreak

Bei iOS-Geräten wird das Rooten eines Gerätes als Jailbreak, zu deutsch Gefängnisausbruch, bezeichnet. Nach einem erfolgreichen Jailbreak ist es möglich, jede beliebige unsignierte App auf dem Gerät zu installieren. Somit können auch Apps, die nicht im App Store vorhanden sind, installiert werden. Als freie Alternative dient hierzu der Cydia Store, der über die Cydia App angesprochen werden kann und signierte wie unsignierte Apps kostenfrei anbietet. Zusätzlich werden verschiedene Restriktionen freigeschaltet, unter Anderem die Beschränkung auf System-eigene Hintergrundbilder. In der Vergangenheit wurden vorwiegend Sicherheitslücken im PDF-Reader von iOS ausgenutzt, um eigenen Schadcode zu injizieren und das Gerät auf diese Weise freizuschalten. Ein iPhone Jailbreak ist in den USA nach einem Gerichtsbeschluss vom 11. Juni 2010 des US Copyright Office [40] offiziell erlaubt.

4.8.3 Vorteile

Das Rooten, beziehungsweise der Jailbreak eines Gerätes eröffnet laut [21], durch die hoheitlichen Zugriffsrechte auf Systemdateien, die folgenden Möglichkeiten:

- Systemdateien können modifiziert werden. Somit können Zusatzfunktionen integriert werden, beispielsweise ein Energiemanager in der Statusleiste.
- Vorinstallierte, sowie System-Applikationen, können modifiziert und entfernt werden.
- Es können eigene Themes zum Ändern der Oberfläche verwendet werden.
- Apps, die Root-Rechte voraussetzen, können verwendet werden.
- Ein Backup des kompletten Systems ist nur mit Root-Rechten möglich.
- Apps, deren Daten, sowie der Cache können komplett auf die SD-Karte ausgelagert werden. (Nur Android)
- Das Aufspielen so genannter Custom-ROMs wird möglich. Diese stellen komplett modifizierte Android-Systeme dar, die meist auf bestimmte Kriterien hin optimiert wurden, zum Beispiel Leistung, Akkulaufzeit oder zusätzliche Einstellmöglichkeiten und Features bieten.
- Ist der Update-Support des Herstellers sehr langsam oder bereits eingestellt, so kann ein Custom-ROM auf Basis einer aktuellen Android Version eingespielt werden. Somit können im alten System bestehende Sicherheitslücken geschlossen werden.

4.8.4 Nachteile und Gefahren

Als offensichtlichster Nachteil ist der Verfall der Herstellergarantie zu nennen. Da das Rooten und Modifizieren von Systemdateien zu Schäden am Gerät führen kann, übernehmen Hersteller keine Verantwortung und grenzen modifizierte Geräte aus ihrer Garantieleistung aus. Grundsätzlich kann die Originalsoftware jedoch problemlos wiederhergestellt werden. Ein Totalausfall des Gerätes beim Root-Vorgang ist in sehr seltenen Fällen ebenso möglich, somit geschieht der Vorgang stets auf eigene Gefahr. Im Englischen wird ein solches Gerät als „bricked“ bezeichnet.

Ein gejailbreaktes iPhone eröffnet zudem neue Gefahrenquellen. Da der root-Benutzer freigeschaltet wird, kann ein Angreifer diesen nutzen, um Schadcode auszuführen und die Sicherheitsmechanismen des Systems zu umgehen. Des Weiteren ist auf einem gejailbreakten iPhone der SSH-Dienst aktiviert, welcher es ermöglicht, Systembefehle auszuführen. Da nach einem Jailbreak die Passwörter für die Benutzer „root“ und „mobile“ bei jedem Gerät auf *alpine* gesetzt werden, ist es dringend notwendig, diese umgehend zu ändern. Ansonsten kann ein Angreifer diese Lücke über drahtlose Netzwerke ausnutzen, um einen Angriff zu starten und so beispielsweise einen Trojaner einzuschleusen. Hierzu bedient er sich lediglich des Standard-Passwortes und eines SSH-Clients und kann so ungehindert Dateien des Systems manipulieren, ohne dass der Benutzer etwas hiervon bemerkt.
2009 wurde diese Lücke bereits von einem niederländischen Hacker ausgenutzt, um Besitzern eines gejailbreakten iPhone eine SMS Nachricht zu schicken, mit dem Inhalt, dass ihr iPhone gehackt wurde und er die Kontrolle über ihre Daten hätte, siehe Grafik 4.2. Die Opfer wurden anschließend dazu aufgefordert, einen Betrag von 5$ auf ein Paypal-Konto zu überweisen, um die Manipulation rückgängig zu machen. Auf Grund moralischer Bedenken zog der Täter seine Aufforderung jedoch zurück und veröffentlichte die Informationen zur Reparatur im Netz. [16]
Ein weiterer Trojaner nutzte ebenfalls das Standard-Passwort und den SSH-Dienst, um ein Bot-Netzwerk aus infizierten iPhones aufzubauen.

Bei einem gerooteten Android-System besteht dieses Problem nicht. Es wird zwar der Superuser mit User-ID 0 freigeschaltet, jedoch wird beim Root-Vorgang in der Regel die App „Superuser“ installiert, die den Zugriff auf Root-Rechte regelt. Sie fragt den Nutzer bei jeder neuen App, die Root-Rechte anfordert, nach dessen Zustimmung und legt eine Whitelist mit Apps an, welche vollen Zugriff auf das System erhalten.

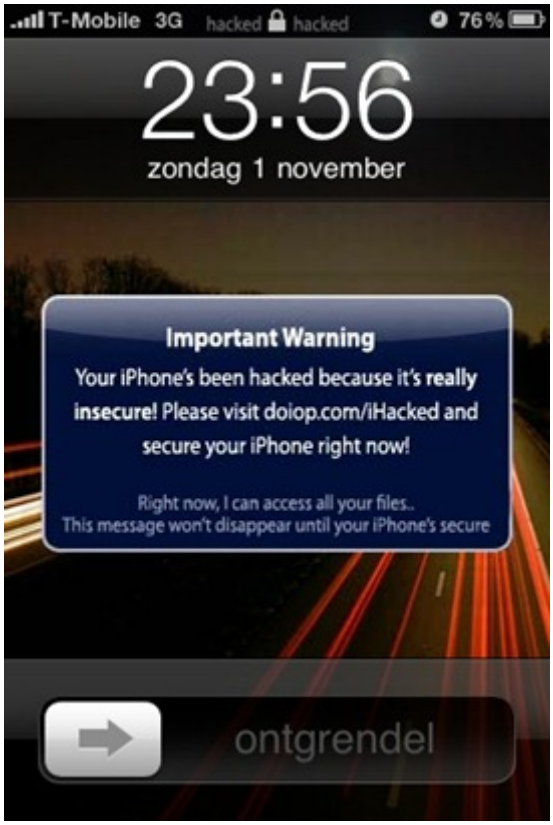

Abbildung 4.2: Hacker-Nachricht nach SSH-Angriff

4.8.5 Schutzmaßnahmen

Nach einem iPhone-Jailbreak sollten unbedingt die Passwörter der Nutzer „root“ und „mobile“ geändert werden, da diese standardmäßig auf „alpine“ gesetzt sind und so SSH-Angriffe von außen ermöglichen. Die folgende Anleitung beschreibt die Änderung beider Passwörter des OpenSSH-Dienstes, begonnen mit dem Passwort des Nutzers „mobile“:

1. Installation von MobileTerminal aus dem Cydia Store. Anschließend iOS neu starten.
2. MobileTerminal starten und Kommando *passwd* eingeben.
3. Nun kann das neue Passwort samt Bestätigung eingegeben werden. Hierzu wird zunächst das alte Passwort abgefragt, welches *alpine* lauten sollte (selten auch *dottie*).

Um das Passwort des Nutzers „root“ zu ändern kann analog vorgegangen werden. Jedoch muss man sich vor Beginn der Prozedur als „root“-Nutzer anmelden. Hierzu muss vor dem einloggen *login* eingegeben werden, danach der gewünschte Benutzer *root* mit Passwort *alpine*. Anschließend müssen die Schritte 1.-3. wiederholt werden. [16]

Alternativ kann OpenSSH bis zum nächsten Systemneustart auch deaktiviert werden. Hierzu sind folgende Schritte nötig:

1. Starten von Mobile Terminal.
2. Eingabe von *launchctl* zur Verwaltung von Diensten.
3. Eingabe von *stop com.openssh.sshd* zum Stopppen von OpenSSH.

Möchte man ganz auf Nummer Sicher gehen, kann OpenSSH auch über die App Cydia deinstalliert werden. Hierzu navigiert man in Cydia zu *Verwalten/Pakete/OpenSSH/Verändern/Entfernen*. [55]

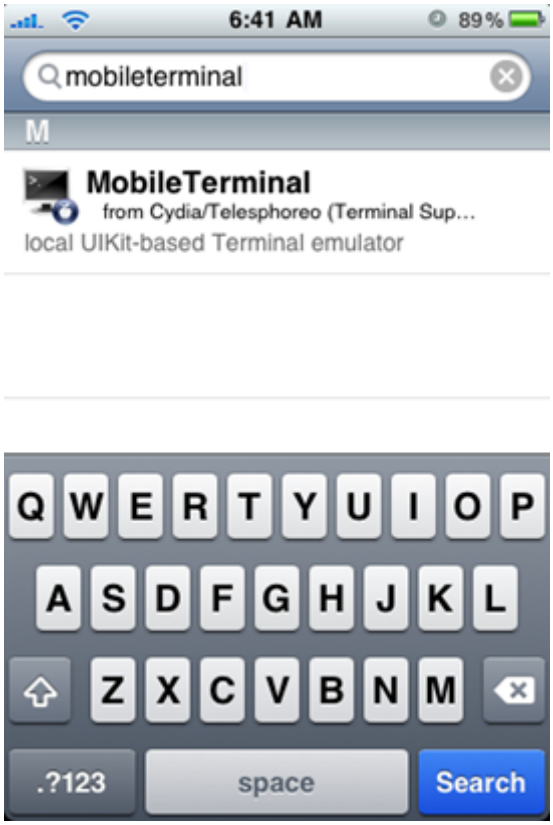

Abbildung 4.3: Root Passwörter ändern, 1

Nach dem Rooten eines Android-Gerätes gibt es keine besonderen Sicherheitsmaßnahmen, die man beachten muss. Man kann sein Gerät wie gewohnt weiter verwenden. Es gilt weiterhin der Grundsatz, dass man nur Software installieren soll, die man kennt und der man vertraut. Insbesondere, wenn es sich dabei um Software handelt, die Root-Rechte benötigt. Man sollte sich bei jeder Software, die Root-Rechte anfordert, im Klaren darüber sein, dass diese Zugriff auf das gesamte System hat. Handelt es sich dabei um Schadsoftware, so kann sie nicht nur alle Daten auslesen, sonder diese auch beliebig modifizieren.

4.9 Systemupdates

Das folgende Kapitel beschäftigt sich mit einer Betrachtung der Zeitspannen, die für Updatevorgänge relevant sind. Dies ist interessant, da Sicherheitslücken meist mit einem Update auf eine neuere Version behoben werden, sofern sie dem Softwarehersteller bekannt sind. Interessant sind hierbei die Zeitspanne zwischen dem Bekanntwerden einer Sicherheitslücke und deren Behebung (Reaktionszeit), sowie die Zeitspanne bis zur Bereitstellung für das Endgerät (Bereitstellungszeit). Ebenso betrachtet wird die Zeitspanne zwischen der Veröffentlichung zweier Updates (Update-Häufigkeit) sowie der Support-Zeitraum für Endgeräte (Update-Support). Da die beobachteten Zeitspannen für die einzelnen Plattformen stark auseinander gehen, können diese zur Bewertung der Sicherheit einer Plattform hinzugezogen werden. Ein wichtiges Kriterium für die Entstehung dieser Differenzen, ist die Verantwortlichkeit der Hardwareherstellung. Es wird deshalb bei der Bereitstellungszeit und dem Update-Support zwischen Systemherstellern und reinen Softwareherstellern unterschieden.

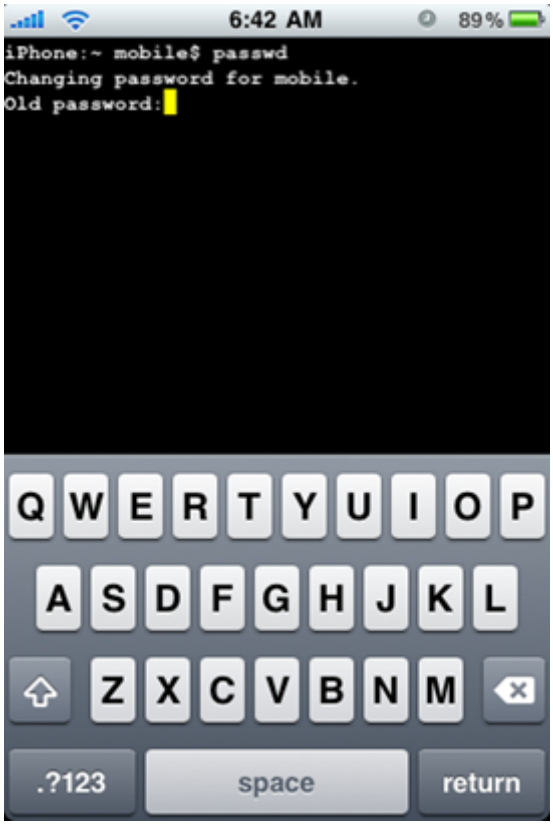

Abbildung 4.4: Root Passwörter ändern, 2

4.9.1 Reaktionszeit

Die Reaktionszeiten der Softwarehersteller auf ein Update sind nahezu nicht messbar. Dies resultiert aus der Tatsache, dass nicht jedes Update einer bestimmten Sicherheitslücke zuzuordnen ist. Somit wird dieser Aspekt nicht zur Sicherheitsbewertung herangezogen. Es ist jedoch auffällig, dass bestimmten Sicherheitslücken eine höhere Priorität zugewiesen wird, als anderen. Im Beispiel von Apple galt dies diversen Lücken im PDF-Betrachter von iOS. Diese ließen sich nämlich dazu nutzen, mit Hilfe einer manipulierten PDF-Datei einen Jailbreak auf dem Gerät durchzuführen. Um diesem Mechanismus einen Riegel vorzuschieben, und natürlich um die Sicherheit des Systems wiederherzustellen, veröffentlichte Apple innerhalb kurzer Zeit kleine Updates für ihr System iOS. Die letzte bekannte Sicherheitslücke im PDF Betrachter wurde mit dem Update von Version 4.3.3 auf Version 4.3.4 geschlossen, was alte Jailbreak Mechanismen unbrauchbar machte, siehe auch Abschnitt 4.8, Sicherheit von Geräten.

4.9.2 Bereitstellungszeiten

Reine Softwarehersteller

Unter dieser Kategorie fallen Hersteller, die nur das mobile Betriebssystem bereit stellen. Die Endgeräte werden somit von unterschiedlichen Herstellern entwickelt und mit dem Betriebssystem bespielt. Dies führt dazu, dass das System bei jedem Update durch den Gerätehersteller angepasst werden muss, indem Treiber, Apps und Bedienoberfläche ergänzt werden. Das so entstehende System muss anschließend noch vom Gerätehersteller getestet werden. Handelt es sich um Provider-spezifische Geräteversionen, so unterzieht dieser die Geräte anschließend ebenfalls einem Test. Bei Android zertifiziert Google die so getesteten Versionen meistens nachträglich, was nochmals zu einer Verzögerung führt. Treten bei einem der genannten Tests Pro-

bleme auf, so muss der Prozess erneut durchgeführt werden. Dies führt dazu, dass die Bereitstellung eines Updates auf dem Gerät erst mehrere Monate nach einem offiziellen Update des Betriebssystemherstellers erfolgt, oder überhaupt nicht. Benutzer werden dann mit veralteten Betriebssystemen und Sicherheitslücken zurückgelassen.

Zur Kategorie gehören die Unternehmen Google, mit dem Betriebssystem Android, sowie Microsoft, mit Windows Phone 7. Google stellt ungefähr halbjährlich neue Android Versionen, mit neuen Features, zur Verfügung. Sicherheitsupdates erscheinen regelmäßig zwischen diesen Versionen. Google versorgt die Gerätehersteller hierbei nur mit ihrer Software, die Implementierung der Treiber sowie das Rollout erfolgt dezentral. Microsoft geht mit ihrem Betriebssystem Windows Phone 7 einen ähnlichen Weg, sammelt jedoch die Updates nach Fertigstellung durch die Gerätehersteller wieder ein und verteilt diese zentral. Dies entlastet die Hersteller teilweise, setzt sie aber gleichzeitig seitens Microsoft unter Druck, die Updates tatsächlich für ihre Geräte umzusetzen. Zwischenzeitlich bot Microsoft Nutzern von Windows Phone 7 die Möglichkeit auf einer Webseite namens „Where is my phone update?“ einzusehen, wann ein Update für das eigene Gerät zur Verfügung stehen wird. Leider wurden die Informationen laut [38] Anfang 2012 wieder, ohne Stellungnahme seitens Microsoft, vom Netz genommen.

Leider erscheinen Softwareupdates für Android-Endgeräte unregelmäßig und oft verspätet. Es folgt ein Beispiel für die Umsetzung seitens der Gerätehersteller: So erhielt das HTC Sensation das Update auf Android 2.3.4, das am 29. April 2011 erschienen ist, erst im August 2011, 3 Monate später. Zu diesem Zeitpunkt war bereits Android 2.3.5 verfügbar, welches neue Sicherheitslücken behob. Viele Hersteller verzichten aufgrund der aufwändigen Anpassungen und Tests auch ganz auf Updates. Dies führt dazu, dass viele Geräte veraltete Software mit Sicherheitslücken verwenden. Nutzer können durch das Rooten des Gerätes jedoch aktuelle Custom-Roms, basierend auf einer aktuellen Android Version, einspielen. Dies erfordert jedoch Expertenwissen und birgt Risiken, wie den Verlust des Garantieanspruchs. Näheres hierzu in Abschnitt 4.8, Sicherheit von Geräten.

Ein Blick auf den derzeitigen Anteil der Android-Versionen auf dem Markt bestätigt diese These. Die Statistik 4.5 stammt von Google, herausgegeben im September 2011: Die aktuell neueste Version 2.3.3 (Gingerbread) kommt nur zu 30,7% vor. Die ältesten Versionen 1.5 und 1.6 von 2009 machen nur noch 2,8% aller aktiven Geräte aus. Am meisten verbreitet ist Version 2.2 (Froyo) vom 20. Mai 2010 mit 51,2%, dicht gefolgt von Version 2.1 (Eclair) vom 12. Januar 2010 mit 13,3%. Somit laufen 64,5% aller Android Geräte mit einem System, dass mindestens 15 Monate alt ist. Eine lange Zeitspanne in der Informationstechnik.

Systemhersteller

Zu dieser Kategorie zählen Hersteller, die sowohl die Hardwareplattform, als auch das zugehörige Betriebssystem anbieten. Somit muss bei einem Systemupdate kein zweiter Hersteller zwischengeschaltet werden, durch den sich ein Update verspäten könnte. Dies führt zu drastisch reduzierten Bereitstellungszeiten, da die nötigen Anpassungen und Tests intern durchgeführt werden können. Tests durch unterstützte Provider müssen aber auch hier durchgeführt werden. Trotzdem stehen Updates für Geräte von Systemherstellern generell schneller zur Verfügung, als bei reinen Softwarelieferanten. Als populärster Vertreter dieser Kategorie lässt sich Apple nennen. Das Unternehmen hat sowohl das iPhone, als auch das Betriebssystem iOS, entwickelt und vertrieben. Updates für das System stehen Entwicklern meist sofort, Anwendern nach geringer Verzögerung für alle unterstützten Generationen zur Verfügung.

Ebenfalls in dieser Kategorie befindet sich HP mit seinem eingestellten Betriebssystem webOS und den zugehörigen drei Endgeräten, siehe Abschnitt 3.4, webOS. Als Ausnahme unter dem Betriebssystem Android ist noch die Nexus Serie von Google zu nennen, welche über eine lange Zeit hinweg stets auf dem neuesten Stand gehalten wird. Die Endgeräte werden hierbei von HTC(Nexus One) und Samsung(Nexus S und Galaxy Nexus) gefertigt.

Ein weiterer Vertreter der Systemhersteller ist mittlerweile auch RIM, da kaum noch ein anderer Hersteller die BlackBerry-Technologie einsetzt. Hier sind die Updatezyklen von Version zu Version unterschiedlich.

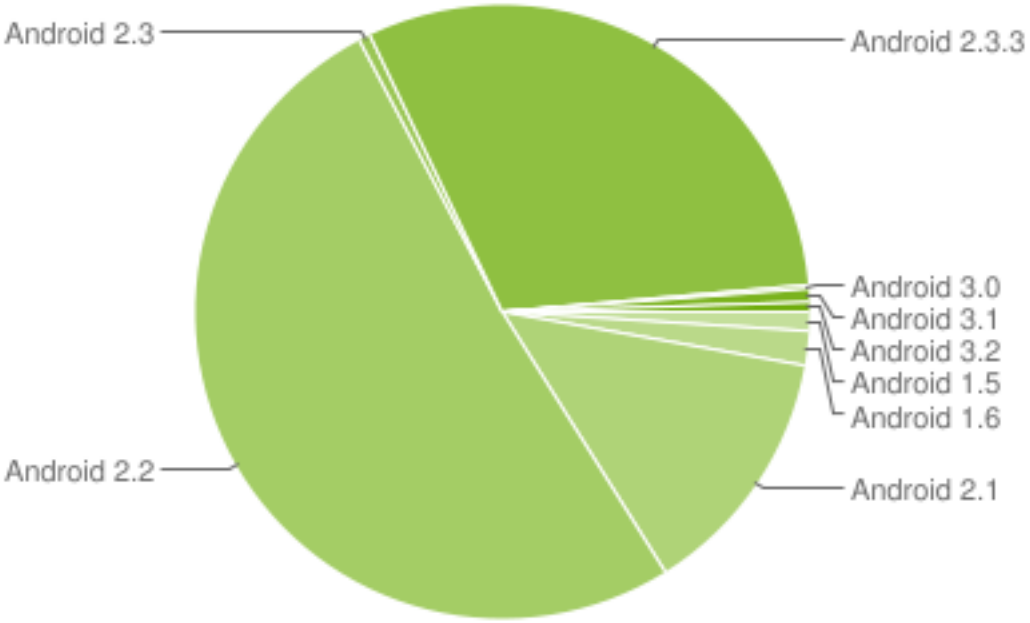

Abbildung 4.5: Android Versionsverteilung September 2011

Neue Versionen erscheinen, wie bei iOS, für Entwickler früher als für Endanwender. Anwender die ihr BlackBerry im Unternehmen einsetzten und mit einem BlackBerry Enterprise Server oder der BlackBerry Professional Software integriert sind, werden von ihrem jeweiligen Systemadministrator über das BES-Netz aktualisiert. Alle anderen Anwender können ihr BlackBerry einfach über das USB-Kabel verbinden und aktualisieren.

Die aktuellste Version für Smartphones der Firma Blackberry ist 7.0. Diese wird für Geräte der Reihe 9000 zur Verfügung gestellt. Vereinzelte Geräte können leider nur auf Version 5.0 oder 6.0 aktualisiert werden. Ältere Geräte der Reihe 8000 müssen sich mit Version 4.5 begnügen. Somit herrscht auch bei Blackberry OS als Systemhersteller eine hohe Versionsdiskrepanz vor, ähnlich wie bei Android. Die Sicherheit älterer Geräte ist somit fraglich. [67]

Vergleich

Abbildung 4.6 zeigt für iPhones und für eine Auswahl aus Androidgeräten beispielhaft das Problem, das sich durch verschiedene Updateverfahren ergibt. Während iOS-Geräte sehr schnell die neusten Updates erhalten, haben viele Android-Geräte bereits beim Kauf veraltete Software, erhalten sehr spät oder gar keine Updates.

Diese Problematik lässt sich so auf alle Hersteller übertragen. Hardwarehersteller hinken meist den Updates des Betriebssystemherstellers hinterher, oder bringen für ältere und billige Geräte keine Updates. Systemhersteller bringen, im Gegensatz dazu, meist Updates für all ihre Geräte heraus. Dabei kann es sein, dass ältere Geräte aus Leistungs- oder Marketinggründen nicht alle neuen Funktionen erhalten, es werden jedoch Sicherheitslücken geschlossen.

4.9.3 Update-Häufigkeit

Updates werden bei Smartphones nicht nur zur Verbesserung der Sicherheit angeboten, sondern auch zur Verbesserung der Akkulaufzeit, der Leistung und um neue Funktionen zu ermöglichen. Smartphone Betriebssysteme werden somit aktiv weiterentwickelt. Apple bietet durchschnittlich jedes Jahr seit Einführung von

Version	Codename	Veröffentlichung
1.0		23. September 2008
1.1		9. Februar 2009
1.5	Cupcake	30. April 2009
1.6	Donut	15. September 2009
2.0	Eclair	26. Oktober 2009
2.0.1	Eclair	3. Dezember 2009
2.1	Eclair	12. Januar 2010
2.2	Froyo	20. Mai 2010
2.2.1	Froyo	18. Januar 2011
2.2.2	Froyo	22. Januar 2011
2.2.3	Froyo	21. November 2011
2.3	Gingerbread	6. Dezember 2010
2.3.3	Gingerbread	9. Februar 2011
2.3.4	Gingerbread	29. April 2011
2.3.5	Gingerbread	25. Juli 2011
2.3.6	Gingerbread	2. September 2011

Tabelle 4.2: Android Updatezyklen September 2011

iOS eine neue Version ihres Betriebssystems an. Dies geht erfahrungsgemäß mit einem öffentlichen Betatest für Entwickler einher, damit diese die Möglichkeit haben, ihre Apps an die neuen Funktionen anzupassen. Kleinere Versionsupdates, erkennbar an der Erhöhung der Nachkommastellen der Versionsnummer, dienen meist zum Schließen offener Sicherheitslücken und Beheben von Fehlern. Diese kommen in kürzeren Abständen von mehreren Wochen zwischen den großen jährlichen Updates auf den Markt. Im Oktober 2011 kam beispielsweise die neueste iOS Version 5.0 heraus, welche als größte Neuerung die Anbindung an den eigenen Clouddienst iCloud [3] schaffen sollte. Abbildung 4.7 zeigt die Updatezyklen des Betriebssystems iOS.

Bei Googles Betriebssystem Android sieht es ähnlich aus, wobei Google eine andere Versionierung verwendet als Apple, siehe Tabelle 4.2.

4.9.4 Update-Support

Reine Softwarehersteller

Geräte von Drittherstellern, welche das Betriebssystem eines Softwareherstellers einsetzen, bieten einen unzuverlässigen Update-Support. Dieser richtet sich ausschließlich nach dem Gerätehersteller und nicht nach dem Softwarehersteller. Grund für den unzuverlässigen Update-Support ist der hohe Aufwand, der durch eine große, heterogene Produktpalette an Endgeräten entsteht. Des Weiteren nutzen Dritthersteller neue Systemversionen, um ihre Geräte besser zu verkaufen. So entstehen für alle Geräte unterschiedliche Support-Zeiträume, die jedoch meist unter einem Jahr betragen. Der Updatesupport beträgt bei Android-Geräten durchschnittlich 9 Monate. Nur wenige, meist hochpreisige Geräte, werden länger als ein Jahr mit Updates versorgt. Als Ausnahme gilt beispielsweise Googles Nexus One, welches über 15 Monate Update-Support erhalten hat und auch weiterhin versorgt wird. Dies kann jedoch zur zweiten Kategorie gezählt werden, welche im Folgenden beschrieben wird.

Systemhersteller

Ebenso wie die Geschwindigkeit der Bereitstellung kürzer ist, ist der Zeitraum, über den Updates angeboten werden, bei Systemlieferanten deutlich länger. Man kann davon ausgehen, dass Updates solcher Herstel-

ler nach dem Erscheinen nahezu umgehend für alle Geräte bereitgestellt werden und einige Generationen überdauern. Apple bietet hierbei für das iPhone 3GS, als einziger Hersteller auf dem Markt, einen Update-Support von über 30 Monaten.

4.10 Sicherheitssoftware für Smartphones

Aufgrund der bisher aufgeführten Bedrohungen und der wachsenden Anzahl an Sicherheitsvorfällen auf mobilen Endgeräten, gibt es bereits die ersten Sicherheits-Apps für Smartphones. Als prominenter Anbieter stellt Kaspersky seine Antiviren- und Sicherheitssoftware als „Mobile Security 9“ für Android, sowie Blackberry, zur Verfügung [35]. Ebenso bietet McAfee ein Sicherheitspaket für Smartphone-Betriebssysteme an, unterstützt zusätzlich aber Apple iOS [41]. Symmantec unterstützt stattdessen mit „Norton Mobile Security Lite“ nur Googles Android [61]. Einige Hersteller unterstützen zusätzlich ältere Systeme, wie Windows Mobile und Symbian, welche jedoch wegen mangelnder Aktualität im Rahmen dieser Fachstudie nicht weiter betrachtet werden. Aufgrund des geringen Alters von Windows Phone 7 gibt es für dieses Betriebssystem noch keine Sicherheitslösungen. Neben einem umfassenden Daten- und Virenschutz, rüsten die Programme fehlende Funktionen des Diebstahlschutzes nach, zu dem unter anderem Fernlöschung und -Sperrung zählen. Android verfügt in seinem Grundzustand als einziges System über keine dieser Mechanismen. Manche Gerätehersteller, wie beispielsweise Samsung, rüsten diesen jedoch mit einem eigenen Portal nach (Samsung Dive).

Zu den angebotenen Funktionen von Sicherheits-Apps gehören im Wesentlichen:

Virenschutz Schutz gegen Viren in Form von infizierten Downloads oder Apps.

Anti-Spyware Schutz gegen Trojaner und gleichwertige Schädlinge in Form von Downloads oder Apps.

Anti-Phishing Schutz gegen Phishing-Attacken zum Ausspähen sensibler Daten.

Schutz der Privatssphäre Inhaltsfilterung, sowie Kindersicherung. Ebenso können bestimmte Nummern über Black- und Whitelists verboten oder erlaubt werden, um Anrufe zu filtern.

Diebstahlschutz Sperrung mobiler Geräte, sowie ferngesteuerte Löschung von Daten auf Mobilgeräten. Zudem können verloren gegangene Geräte lokalisiert und verfolgt werden.

Datenschutz Sicherung und Wiederherstellung mobiler Daten, sowie Verschlüsselung zum Schutz vor unbefugtem Zugriff.

Firewall Schutz des Gerätes vor schädlichen Verbindungen, eingehend (Hacker), sowie ausgehend (Trojaner).

Deinstallationsschutz Schützt die Sicherheitssoftware vor Deinstallation und somit vor Unschädlichmachung durch Dritte.

Hierbei ist zu bemerken, dass nicht alle Funktionen auf allen Systemen gleichermaßen angeboten werden. So bietet Kaspersky beispielsweise die Verschlüsselungsfunktion für Android nicht an. Hintergründe könnten technischer, aber auch rechtlicher Natur sein – Verschlüsselungsmechanismen sind nämlich, vor allem in den USA, bestimmten Gesetzen unterlegen, die eine wahlfreie Einbindung von Verschlüsselungsalgorithmen verbieten.

4.10.1 Test von Android Antivirenprogrammen

AV-Test hat folgende kostenlose Antivirenprogramme für Android getestet:

- Antivirus Free
- BluePoint Antivirus Free
- GuardX Antivirus

- Kinetoo Malware Scan
- LabMSF Antivirus beta
- Zoner Anti-Virus Free
- Privateer Lite

Wie in Abbildung 4.8 zu sehen ist, haben alle getesteten Antivirenprogramme eine niedrige Erkennungsrate, so dass sie den Anwender bislang nicht gut schützen. Dies kommt vor allem von der recht kurzen Zeitspanne, in der die Apps nun auf dem Markt sind. Als schlechter Gewinner dieser Studie unter den Gratis-Apps geht „Zoner Anti-Virus Free“ hervor. Die zum Vergleich getesteten kostenpflichtigen Programme „F-Secure Mobile Security“ und „Kaspersky Mobile Security“ schnitten deutlich besser ab und erkannten nahezu jedes Schadprogramm. [29]

Als bekannter Vertreter von Desktop-Systemen ist zudem Avast! Mobile Security für Android zu nennen, welches jedoch noch keine Tests unterlaufen hat. Dieses ist ebenso frei verfügbar und rüstet auch einen Diebstahlschutz, sowie eine Firewall nach. In persönlichen Tests kam es jedoch zu Leistungseinbußen des Systems. Die Erkennungsrate konnte leider ebenso wenig getestet werden.

Somit ist von kostenlosen Antivirenprogrammen bislang abzuraten, da diese nicht die erhoffte Erkennungsrate und die damit verbundene Sicherheit liefern. Nutzlose Antivirenprogramme können sogar die Sicherheit herabsetzen, da sich der Anwender in Sicherheit wägt und somit eher riskante Anwendungen installiert. Kostenpflichtige Programme von Firmen, mit Erfahrung in der Erkennung von Schadsoftware, sind hingegen auf Grund der aufkommenden mobilen Hacker- und Virenszene empfehlenswert. Dies gilt vor allem für das relativ offene Android, da hier keine App-Prüfung seitens Google statt findet und somit die Gefahr einer Infektion höher ist, als bei der Konkurrenz.

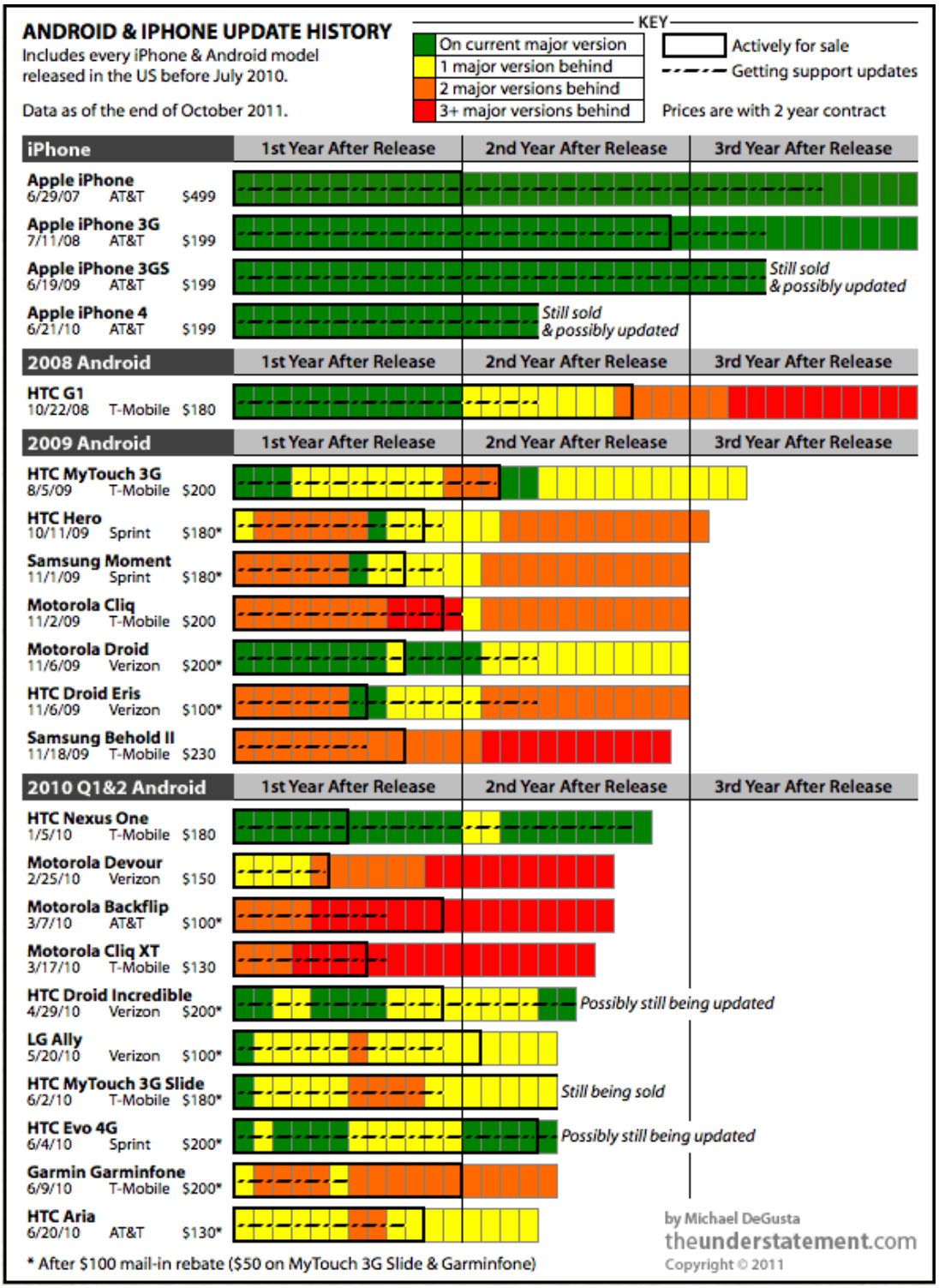

Abbildung 4.6: Android und iOS Updatevergleich [9]

Softwareversion	Vorgestellt	Anzahl der Betas	Veröffentlichung
iOS 2.0	6. März 2008	8	11. Juli 2008
iOS 3.0	17. März 2009	5	17. Juni 2009
iOS 4.0	8. April 2010	4	21. Juni 2010
iOS 5.0	6. Juni 2011	aktuell 5	?

Abbildung 4.7: iOS Updatezyklen September 2011

2. Test results

Name	Vendor	Version	Installation	Rating[2]	Size	Detection	
						Manual Scan	On installation
Antivirus Free	Creative Apps	1.3.1	1.000.000 - 5.000.000	4,5 / 41375	0,4 MB	0 / 172 (0%)	0 / 10 (0%)
http://zrgiu.com/							
BluePoint Antivirus Free	BluePoint Security	4.0.14	10.000 - 50.000	4,2 / 549	3,4 MB	2 / 172 (1%)	1 / 10 (10%)
http://www.bluepointsecurity.com/							
GuardX Antivirus	Qstar	2.3	100.000 - 500.000	4,6 / 2824	1,2 MB	0 / 172 (0%)	0 / 10 (0%)
http://guardx.qstar.org/							
Kinetoo Malware Scan	CPU Media SARL	1.6.9	10.000 - 50.000	4,2 / 184	0,2 MB	11 / 172 (6%)	1 / 10 (10%)
http://kinetoo.com/							
LabMSF Antivirus beta	LabMSF	1.0	1.000 - 5.000	4,3 / 16	1,0 MB	0 / 172 (0%)	0 / 10 (0%)
http://labmsf.com/							
Privateer Lite	Online Vault	2.1.4	1.000 - 5.000	4,5 / 28	1,1 MB	0 / 172 (0%)	1 / 10 (10%)
http://www.privateerlabs.net/privateer-mobile-released							
Zoner AntiVirus Free	ZONER	1.2.4	50.000 - 100.000	4,6 / 1614	0,9 MB	55 / 172 (32%)	8 / 10 (80%)
http://www.zonerantivirus.com							

Commercial products for comparison

Name	Vendor	Version	Installation	Rating	Size
F-Secure Mobile Security[3]	F-Secure	7.1	-	-	4,5 MB
http://www.f-secure.com/de/web/home_de/protection/mobile-security/overview					
Kaspersky Mobile Security	Kaspersky Lab	9.10.77	10.000 - 50.000	4,2 / 992	3,8 MB
http://www.kaspersky.com/kaspersky_mobile_security					

[2] Android Market Rating / Number of ratings
[3] F-Secure Mobile Security is not (yet) available via the Android Market

4

AV-TEST The Independent IT-Security Institute

Abbildung 4.8: Android-Virenscanner im Test

5 Privacy

In diesem Kapitel geht es um Datenschutz. Was bedeutet der Begriff? Wie ist er im Gesetz verankert? Welche Informationen fallen unter den Datenschutz, und wie wird er in Smartphones von den jeweiligen Herstellern gehandhabt? Diese und weitere Fragen werden auf den folgenden Seiten diskutiert und beantwortet.

5.1 Definition: Was ist Privacy?

Privacy befasst sich in erster Linie mit der Privatsphäre jeder einzelnen Person, denn jeder Mensch hat das Recht auf die freie Entfaltung der Persönlichkeit. Der Schutz der Privatsphäre ist im deutschen Grundgesetz aus dem allgemeinen Persönlichkeitsrecht (Art. 2 Abs. 1 i. V. m. Art. 1 Abs. 1) [53] abzuleiten. Seit circa 1980 werden, hauptsächlich im europäischen Raum, mehrere Dimensionen der Privatsphäre beschrieben:

psychologische Privatsphäre: die psychologisch empfundene Intimsphäre

physische Privatsphäre: zum Beispiel die Unversehrtheit der Wohnung

interaktionelle Privatsphäre: Kontrolle über Interaktionen und Kommunikation

informationelle Privatsphäre: Vertraulichkeit von Informationen über eine Person

Das Hauptaugenmerk dieser Fachstudie liegt bei der interaktionellen und informationellen Privatsphäre. Denn heutzutage werden Geschäftsdaten, Bankdaten und -verbindungen, Telefonnummern, E-Mails sowie persönliche Dokumente auf dem Smartphone, häufig ungeschützt, gespeichert. Was früher in verschließbaren Aktenschränken oder Safes abgelegt wurde, ist nun frei zugänglich sobald das Smartphone gestohlen wurde oder verloren geht. Die wichtigste Methode, seine Daten zu schützen, ist sie mittels Passwörtern und Verschlüsselung unzugänglich zu machen. Im Folgenden wird der Schutz durch Passwörter diskutiert, sowie einige Angriffsmuster, die auf den Zugriff auf sensible persönliche Daten abzielen.

5.2 Sicherheit von Passwörtern

Der Zugriffsschutz durch Passwörter existiert schon seit Beginn der Informatik. Ein Passwort dient zur eindeutigen Identifizierung und Authentifizierung zur Kommunikation zwischen zwei Kommunikationspartnern. Hierbei erfragt ein Partner in der Regel Einlass zu einem Service oder einem geschützten Bereich, der durch den anderen gewährt oder verweigert wird. Die Authentizität dieser Anfrage bleibt solange gewahrt, wie das Passwort geheim ist. Bei einem Passwort handelt sich also um ein so genanntes „Shared Secret“, dass beiden Kommunikationspartnern bekannt ist. Wichtig dabei ist, dass ein effizienter, moderner und sicherer Algorithmus verwendet wird, um das Passwort zu verschlüsseln.

5.2.1 Schwachstellen von Passwörtern

Ohne weitere Schutzmaßnahmen kann auf beiden Seiten beim Speichern des Passworts ein Angriff erfolgen, sowie bei der Kommunikation, d.h. der Übermittlung des Passworts. Ist das Passwort erst einmal bekannt, können wahlfrei alle Informationen eingesehen werden, die die Kommunikationspartner schützen wollten. Hierzu zählen persönliche Daten wie Namen, Adressen, Telefonnummern, sowie Zahlungsinformationen, wie Kreditkartendaten oder Kontonummern. All diese persönlichen, zu schützenden Daten fallen unter das Bundesdatenschutzgesetz.

5.2.2 Schutz vor herkömmlichen Angriffen

Als Anwender kann man sich schützen, indem man ein Passwort ausreichender Sicherheit verwendet, um einem potentiellen Angreifer den Angriff über einen Brute-Force-Ansatz zu erschweren, siehe Unterabschnitt 5.3.1, Brute-Force-Angriffe. Ein sicheres Passwort sollte mindestens 8 Stellen lang sein, aus Buchstaben, Ziffern, sowie Sonderzeichen bestehen, und keine persönlichen Daten oder bekannte Wörter enthalten. Fantasiewörter oder zufällig generierte Passwörter bieten hier den höchstmöglichen Schutz, da diese nicht in Wörterbüchern enthalten sind.

Ein offensichtlicher Schutz vor einem Angriff auf einer der beiden Kommunikationsseiten, ist das Vermeiden einer ungeschützten Speicherung des Passwortes. Bei einem Client-Server Verhältnis funktioniert dies am einfachsten, indem sich der Benutzer das Passwort lediglich merkt, nicht aber aufschreibt oder speichert. Der Server muss das Passwort jedoch in einer sicheren Form hinterlegen, um eine spätere Prüfung zu ermöglichen. Das Passwort sollte dabei jedoch auf keinen Fall als Klartext hinterlegt werden. Hierbei gibt es laut [22] die folgenden Ansätze:

- Als Hash Wert
- Als Hash Wert mit Salt (Hinzufügen einer beliebigen Zeichenkette zu dem Passwort)
- Als Hash Wert mit Salt und anschließendem mehrfachen Hashen über sich selbst

Der Sinn des Hash-Werts besteht darin, dass das Passwort grundsätzlich nicht wiederhergestellt werden kann. Dies ist jedoch nicht notwendig, da der Server das Passwort für eine Zugangsprüfung nicht kennen muss. Somit muss nur noch einer Seite, in diesem Fall der Benutzerseite, das Passwort bekannt sein. Für eine Prüfung wird das Passwort auf Seiten des Benutzers nun mit dem selben Algorithmus gehasht und nur dieser Wert übertragen. Als Algorithmus kommt zum Beispiel MD5 oder SHA512 in Frage, wobei letzterer höhere Sicherheit bietet.

Zum Schutz der Übertragung des gehashten Passwortes sollte in jedem Falle eine verschlüsselte Verbindung wie SSL verwendet werden. Ein Man-In-The-Middle Angriff könnte den Hash-Wert ansonsten abfangen und zur erneuten Übertragung verwenden, um somit Zugang zu erhalten. Zudem sollten fälschungssichere Zertifikate verwendet werden, um die Authentizität des Kommunikationspartners sicher zu stellen. Näheres hierzu in Unterabschnitt 5.3.4, Man In The Middle.

Unter iOS können Passwörter in der von Apple angebotenen Keychain hinterlegt werden. Hierbei müssen Passwörter jedoch mit dem Attribut *kSecAttrAccessibleWhenUnlockedThisDeviceOnly* hinterlegt sein, um maximale Sicherheit zu bieten. In diesem Fall werden die Daten verschlüsselt hinterlegt und nur dann entschlüsselt, sofern das Gerät vom Nutzer entsperrt wurde. Genauere Informationen zur Sicherheit der Keychain finden sich im folgenden Kapitel, Unterabschnitt 5.2.3, Sicherheit der iOS Keychain.

5.2.3 Sicherheit der iOS Keychain

Die Keychain ist Apples zentraler Datenspeicher für sensible Benutzerdaten auf iOS-Geräten. Sie ist Teil der Apple Data Protection und ist als relationale Datenbank aufgebaut. Hier werden Accounts mit Benutzernamen und Passwörter für eine Reihe von Diensten hinterlegt: Email, Groupware, VPN, WLAN, Webseiten sowie Zertifikate für Apps von Dritten. Somit birgt sie ein hohes Sicherheitsrisiko, da sie viele sensible Daten aggregiert.

Um dem entgegenzuwirken verschlüsselt Apple die Keychain mit Hilfe von AES256. Leider verwendet Apple hierbei nicht den vom Benutzer frei gewählten Passcode zum Schutz des Gerätes, sondern eine Kombination die sich aus der festen ID des jeweiligen Gerätes zusammensetzt. Somit ist es einem potentiellen Täter möglich, diese Kombination allein aus den Hardwaredaten des Gerätes zu entschlüsseln.

Dies ist jedoch nicht einmal notwendig, wie das Fraunhofer Institut in ihrem Bericht „Lost iPhone? Lost Passwords!“ [31] demonstriert. Der Artikel beschreibt einen Weg, um Teile der Keychain als Klartext auszulesen. Hierbei geht das Institut von einem durch einen Passcode gesperrten iPhone aus, welches durch Diebstahl entwendet wurde. Das Gerät wurde noch nicht per Remote-Wipe gelöscht, der Diebstahl blieb also unentdeckt. Um den Zugriff zu verhindern wird zunächst die SIM-Karte entfernt. Das so präparierte iPhone wird dann mittels USB-Verbindung und einem frei zugänglichen Jailbreak am PC entsperrt und mit einem SSH-Server versehen. Anschließend konnte das Institut ein selbst erstelltes Skript auf dem so entsperrten Gerät laufen lassen, das über interne Schnittstellen Benutzerdaten und Passwörter der Keychain bereit stellt.

Die Keychain wurde hierbei vom Gerät selbst entschlüsselt, da kein benutzerdefinierter Schlüssel hierfür vorgesehen ist, sondern lediglich eine Kombination aus der Geräte-ID. Dies ist möglich, weil Apple hiermit einen Kompromiss aus Sicherheit und Bequemlichkeit realisiert – gespeicherte Passwörter stehen direkt zum Systemstart zur Verfügung, um bekannte Verbindungen wieder aufzubauen.

Der Artikel [30] führt die verwundbaren Versionen für den beschriebenen Angriff auf. Betroffen sind alle aktuellen iOS Versionen, von Version 3.0 bis einschließlich 4.3.5 sowie 5.0. Obwohl die Gefahr immer noch besteht, wurden in iOS 5 weitergehende Sicherheitsmechanismen für die Keychain nachgerüstet. Es bestehen unter anderem Unterschiede in der Anzahl entschlüsselbarer Einträge, ohne den Benutzerpasscode zu kennen oder das Gerät zu entsperren. In iOS 3.x sind beispielsweise alle Einträge ohne zusätzliches Wissen entschlüsselbar.

Nutzt eine App eines Drittherstellers die Keychain, so ist diese nur dann betroffen, wenn für den gespeicherten Wert das Attribut *kSecAttrAccessibleAlways* gesetzt ist. Dies sollte von Entwicklern durchweg vermieden und der Standardwert *kSecAttrAccessibleWhenUnlocked* gesetzt werden.

Ein Jailbreak bietet leider keinen Schutz, sondern verstärkt das Problem nur. Über den inoffiziellen Cydia Store kann beispielsweise eine Trojaner App auf das Handy gelangen, welche dann über genügend Rechte verfügt um alle Keychain Einträge auszulesen.

iOS 5 verbessert die Sicherheit der Keychain, indem Accountnamen und Beschreibungen nur noch als Hash-Werte hinterlegt werden. Dies erschwert die Zuordnung der enthüllten Geheimnisse. Zudem werden die Zugriffsattribute einiger Dienste wie Microsoft Exchange oder VPN verbessert, indem nun der Passcode des Nutzers erforderlich ist.

Um dem Problem entgegenzuwirken empfiehlt das Fraunhofer Institut nachdrücklich auf die neuste Version von iOS 5 zu aktualisieren. Des Weiteren ist es notwendig, einen mindestens 6-stelligen Passcode für das Entsperren des Gerätes zu setzen. Ohne diesen verlieren die Sicherheitsmaßnahmen von iOS 5 ihren Schutz, was den freien Zugriff auf die Einträge ermöglicht. Ein Passcode von nur 4 Zeichen könnte mit Hilfe einer Brute-Force-Methode schneller geknackt werden. Weitere Informationen zur sicheren Programmierung der Keychain finden sich in Unterabschnitt 5.8.2, Apple iOS.

5.3 Angriffsmuster

In diesem Kapitel möchten wir verschiedene Möglichkeiten und Angriffsmuster beschreiben, welche die Sicherheit von Geräten und Daten gefährden können. Zusätzlich versuchen wir Hinweise zu geben, wie man sich vor den jeweiligen Angriffen schützen kann.

5.3.1 Brute-Force-Angriffe

Bei einem Brute-Force-Angriff probiert der Angreifer sämtliche in Frage kommenden Lösungen blind durch. Dies erfordert jedoch bei einem Passwort mittlerer Sicherheit bereits einen erheblichen Zeitaufwand und kann durch entsprechende Sicherheitsmechanismen erschwert werden. Brute-Force-Angriffe werden meist durch die

Nutzung von Wörterbüchern beschleunigt, welche statistisch häufig verwendete Passwörter enthalten oder Wörter aus dem normalen Sprachgebrauch. Auch werden persönliche Informationen wie Geburtsdaten oder Familiennamen gerne zum Test herangezogen. In Unterabschnitt 5.2.2, Schutz vor herkömmlichen Angriffen wird die Verwendung von Hash-Werten zur Verbesserung der Passwortsicherheit vorgeschlagen. Um Brute-Force-Angriffe auf gehashte Passwörter zu beschleunigen, können Rainbow-Tables (siehe Abschnitt 5.3.1, Rainbow-Tables) verwendet werden.

Rainbow-Tables

Anstatt sämtliche Folgen von Passwörtern durchzugehen, für diese jeweils den Hashwert zu berechnen und damit den Angriff durchzuführen, kann das rechenintensive Hashen der Passwörter einmalig durchgeführt werden und anschließend für weitere Angriffe gespeichert werden. Eine Rainbow-Table ist solch eine vorberechnete Tabelle, um eine Hashfunktion umzukehren. Dazu enthält die Tabelle die Hashwerte von bekannten Ausgangswerten. Rainbow-Tables können verwendet werden, um zu einem bekannten Hashwert den Ausgangswert zu berechnen. Für einfache, kurze Ausgangswerte kann dies auf [54] getestet werden.

5.3.2 Auslesen von WLAN-Passwörtern

WLAN-Passwörter werden auf Smartphones gespeichert, um den Zugang automatisiert und bequem zur Verfügung zu stellen. Deshalb sind diese potentielle Zielscheiben von Hackern, um Zugang zu Privat- oder Firmennetzen zu erlangen.

Grundsätzlich kann der Benutzer unter Android sowie iOS diese Daten nicht auslesen. Das Rechtesystem der Plattformen verhindert den Zugriff. Rooted man sein Android Handy, oder benutzt einen Jailbreak auf einem iPhone (siehe Abschnitt 4.8, Sicherheit von Geräten), so sind diese Dateien jedoch für den Superuser des Systems zugänglich. So kann bei Diebstahl eines Android-Phones mittels USB-Debugging und einem Dateimanager auf die Passwortdatei zugegriffen werden. Diese befindet sich unter */data/misc/wifi/wpa_supplicant.conf*. Die Datei kann zudem von einer App ausgelesen werden. Jedoch benötigt die App dafür ebenso Superuser-Rechte, welche vom Anwender genehmigt werden müssen. Da dies bei einem nicht-gerooteten Smartphone nicht möglich ist, verringert das rooten somit die Sicherheit, vor allem wenn ein Diebstahl erfolgt.

Bei iOS sieht es ähnlich aus, jedoch speichert das System die Passwörter laut [56] in der Keychain (siehe Unterabschnitt 5.2.3, Sicherheit der iOS Keychain). Diese liegt als SQLite Datenbank auf jedem Gerät vor. In einem Artikel, inklusive Demonstrationsvideo, vom Fraunhofer Institut [31] wird anschaulich gezeigt, wie man die WLAN-Passwörter eines gestohlenen iPhones innerhalb von sechs Minuten auslesen kann. Hierfür wird das entwendete iPhone zunächst gejailbreaked, um die Codesperre zu umgehen und den Schreibschutz des Systems aufzuheben. Anschließend wird noch eine Zugangssoftware benötigt, um auf das Dateisystem zuzugreifen. Passwörter werden von iOS in der Keychain zwar verschlüsselt hinterlegt, jedoch nutzt Apple hierfür die Geräte-ID des iPhones, wodurch ein Angriff möglich wird. Auf diesem Weg war es dem Fraunhofer Institut möglich, einige der Keychain-Passwörter im Klartext auslesen.

Weitere Maßnahmen gegen Diebstahl finden sich in Abschnitt 5.7, Diebstahlschutz.

5.3.3 Phishing

Der Begriff lehnt sich laut [68] an den englischen Wörtern für Passwort (Password), ernten (Harvesting) und angeln (Fishing) an und bedeutet übersetzt das „Angeln nach Passwörtern“. Hierbei versucht der Täter seinem potentiellen Opfer Daten auf freiwilliger Basis zu entlocken, indem er vorgibt eine vertrauenswürdige Person oder Instanz zu sein. Hierzu werden Nachrichten per E-Mail oder Instant Messaging verschickt, in denen das Opfer aufgefordert wird, einem augenscheinlich vertrauenswürdigen Link zu folgen und sensible Daten wie Benutzernamen und Passwörter zu hinterlegen. Auf Smartphones kommt als zusätzliche

Kontaktmöglichkeit die Nutzung von als nützliche App getarnte Malware ins Spiel, welche als Ersatz für eine herkömmliche Webseite dienen kann. Denkbar sind gefälschte Banking-Apps, die Daten des Benutzers abgreifen. Zudem können durch Täter E-Mails mit einer Aufforderung eines App-Downloads verschickt werden, welche sich später als Trojaner(s. Unterabschnitt 5.3.7, Trojaner) entpuppt. Der Täter gelangt so in den Besitz von Passwörtern oder Kreditkartendaten welche er zum Missbrauch nutzen kann. Die Erfolgsquote hängt hierbei stark von der Authentizität der gefälschten Nachrichten und Webseiten sowie der Naivität des Opfers ab.

Um dem entgegenzuwirken setzen neuere Webbrowser und E-Mail-Programme auf einen Phishing-Filter, der bei einer möglichen Gefahrenquelle Alarm schlägt und den Benutzer so vor dem Betrugsversuch warnt. Des weiteren wurden die Gesetz an diese Form des digitalen Betruges angepasst, um für einen Betrugsfall die rechtliche Grundlage zu schaffen.

Um sich als Nutzer gegen Phishing zu schützen kann man einfache Faustregeln verwenden: Generell sollten keine E-Mails beantwortet werden, die nach vertraulichen Daten fragen, oder das Aufrufen einer Webseite mit unbekanntem Link fordern. Prüfen der E-Mails auf Authentizität, vor allem der Absenderadresse, ist unerlässlich. Des weiteren sollten vom Benutzer keine Apps auf Anfrage installiert werden, schon gar nicht von unbekannten Quellen. Der Weg über den offiziellen App-Markt ist stets der sicherste.

5.3.4 Man In The Middle

Bei einem Man-In-The-Middle Angriff steht der Angreifer entweder physikalisch oder logisch zwischen zwei oder mehreren Kommunikationspartnern. Er erlangt hiermit die vollständige Kontrolle über den Datenverkehr zwischen den Netzwerkteilnehmern und kann die Informationen nach Belieben einsehen und manipulieren. Hierfür muss der Angriff jedoch unbemerkt erfolgen, damit die Parteien ihre Informationen dem jeweils anderen Partner anvertrauen. Dem Client wird ein falscher Server vorgetäuscht und dem Server ein falscher Client. So bemerken beide Parteien nicht, das eine dritte Instanz zwischengeschaltet ist.

Ein Man-In-The-Middle Angriff ist besonders bei Funknetzen wie beispielsweise WLAN oder Bluetooth erfolgversprechend. Bei der Einwahl in öffentlich angebotene, offene WLAN-Netze fehlt jegliche Art von Verschlüsselung. Übertragene Daten können somit problemlos von Dritten mitgeschnitten werden. Bei Bluetooth kann beispielsweise der ausgehandelte Schlüssel zur Kommunikation abgefangen werden um dem Datenverkehr zu lauschen. Des Weiteren ist es seit 2011 problemlos möglich, GSM-Netze zu simulieren und somit Mobilfunkdaten auszuspähen und zu missbrauchen. Hierunter fallen beispielsweise so genannte IMSI-Catcher, siehe Abschnitt 4.6.3, IMSI-Catcher.

Zum Schutz sollten offene Netze grundsätzlich gemieden werden. Es ist zudem darauf zu achten, eine automatische Einwahl in solche Netze zu unterbinden. Am einfachsten gelingt dies durch das Ausschalten des jeweiligen Netzdienstes bei Nichtgebrauch. Sollte doch einmal die Verbindung mit einem offenen Netz notwendig sein, so müssen zusätzliche Verschlüsselungsverfahren wie SSL und HTTPS eingesetzt werden, um den Datenverkehr zu schützen. Wichtig sind hier die Zertifikate der jeweiligen Kommunikationspartner. Diese verifizieren die Authentizität des Gegenüber und schützen so vor Fälschungen und Imitaten. Internet-Browser weisen bei der Nutzung von HTTPS auf potentiell nicht-vertrauenswürdige Zertifikate hin, so dass die Verbindung wahlweise abgebrochen werden kann. Des Weiteren sollten nur ausreichend verschlüsselte Verbindungen verwendet werden. Informationen zu den Schwachstellen der einzelnen Netzwerke und den bevorzugten Verschlüsselungsverfahren finden sich in Abschnitt 4.6, Sicherheit drahtloser Verbindungen.

5.3.5 Cookie-Klau

Ein Angriff kann auch auf zwischengespeicherte Daten erfolgen. Diese werden bei Browsern teilweise in Textdateien namens Cookies hinterlegt und können somit ausgespäht werden. Ein Angreifer erhält so Zugriff auf persönliche Zugangsdaten, die im schlimmsten Fall als Klartext im Cookie hinterlegt sind. Da Smartphones

über gleichwertige Browser verfügen wie aktuelle Desktopsysteme, sind diese ebenso durch diese Methode angreifbar.

Laut [23] wurde bereits eine Lücke im Android Browser entdeckt, über die Cookies ausgelesen werden konnten. Der Angriff wurde über schädlichen Code in einer Android App ausgeführt, wobei die App jedoch in den Android Market geschleust oder dem Benutzer als Paket zum Download angeboten werden müsste. Google reagierte auf die Lücke in Android und schloss sie mit Version 2.3.5.

Am einfachsten schützt man sich vor einem Cookie-Klau, in dem man erstellte Cookies jedes mal nach dem Beenden des Browsers löschen lässt. Des Weiteren gibt es in vielen Browsern die Einstellung, dass sich Cookies nach einer gewissen Zeit, in welcher sie Gültigkeit besitzen, selbstständig löschen. Leider werden Cookies auf Smartphones meist dauerhaft gespeichert, weshalb eine manuelle Kontrolle der Einstellungen empfohlen wird. In diesem Zusammenhang wird auch das Löschen des Browser-Verlaufs empfohlen, denn dieser wird oft ebenso dauerhaft gespeichert und kann Auskunft über das Surfverhalten des Nutzers geben.

5.3.6 Keylogging

Unter Keylogging versteht man das Abgreifen von Tastatureingaben über Hard- und Software. Ein Software-Keylogger muss sich folglich zwischen Tastatur und Betriebssystem schalten, um die Daten erst abhören und anschließend weiterleiten zu können. Bei einem Smartphone ist die Verwendung eines Hardware-Keyloggers auf Grund der abgeschlossenen Hardwareumgebung im Normalfall nicht möglich. Eine Hardware-Tastatur müsste per Kabel (siehe Unterabschnitt 4.5.1, Universal Serial Bus (USB)) oder Funk (siehe Unterabschnitt 4.6.1, Bluetooth) angeschlossen werden, um den Einsatz von Hardware-Keyloggern zu ermöglichen. Dies ist jedoch nicht ohne Zutun des Nutzers möglich. Software-Keylogger können dagegen, je nach Plattform, relativ leicht zum Abhören verwendet werden. So bietet zum Beispiel Android die Möglichkeit, Tastaturen von Dritten zu installieren. Die Tastatur verhält sich dabei wie ein eigenständiges Programm und kann über den Android Market bezogen werden. Da alle Eingaben über die Softwaretastatur erfolgen, kann diese die Daten auch mithören und für weitere Zwecke speichern, beispielsweise die Optimierung von Textvorhersagen und -Korrekturen. Deshalb wird der Nutzer unter Android vor der Gefahr des Mithörens gewarnt, bevor er eine andere als die integrierte Tastatur aktiviert. Unter iOS ist die Nutzung eigener Tastaturen ohne Jailbreak nicht möglich, somit ist der Einsatz eines Software-Keyloggers hier schwieriger.

Einen weiteren Angriffspunkt bieten Sensoren wie Gyroskope: Zwei Forscher der UC Davis in Kalifornien haben die Gyroskop-Schnittstelle von Android dazu benutzt, Muster bei der Eingabe bestimmter Buchstaben zu erkennen. Der entworfene „TouchLogger“ arbeitet laut eigenen Angaben mit einer bis zu 71,5 prozentigen Genauigkeit. Es ist somit möglich, Passwörter von Smartphones über offizielle Schnittstellen von einem unverdächtigen Service abzuhören.

Einen Schutz gegen Software-Keylogger bieten Virenscanner und Anti-Spyware, die bekannte Schädlinge anhand einer Blacklist erkennen und isolieren können. Diese befinden sich auf Smartphones noch in der Anfangsphase. Weitere Informationen hierzu in Abschnitt 4.10, Sicherheitssoftware für Smartphones.

5.3.7 Trojaner

Als Trojaner, oder trojanisches Pferd, werden Programme bezeichnet, die als normale Anwendung getarnt sind und im Hintergrund weitere Funktionen erfüllen, ohne dass der Benutzer dies erfährt. Auch wenn prinzipiell jedes Programm, dass ohne Wissen des Benutzers im Hintergrund andere Tätigkeiten verrichtet, als Trojaner bezeichnet werden kann, so wollen wir uns hier auf Trojaner mit versteckter Schadsoftware beschränken.

Ein Trojaner kann sich nicht von alleine verbreiten, dies unterscheidet ihn von einem Virus. Die Verbreitung geschieht entweder durch weitere Schadsoftware, oder indem der Trojaner als Spiel oder, als kostenfreie Version eines eigentlich kostenpflichtigen Programmes getarnt wird. Zusätzlich ist es auch möglich, dass ein

Trojaner gezielt auf dem Gerät einer bestimmten Person installiert wird, um diese zu überwachen. Dies kann zum Einen durch Strafverfolgungsbehörden und Geheimdienste geschehen, oder auch durch Arbeitgeber, Geschäfts- oder Lebenspartner und sonstigen Personen, die Zugriff auf das Gerät haben und an den persönlichen Daten interessiert sind. Dies wird zum Beispiel durch Dienste wie FlexiSpy, Mobile Spy oder MobiStealth ermöglicht, welche Anrufe, Nachrichten und GPS-Daten aufzeichnen, oder gar Audiovisuelle-Mitschnitte tätigen. Die Daten werden anschließend an den Server des Dienstleisters geschickt, wo sie dann gegen hohe Gebühren von den Tätern abgefragt werden können. Ausspähen ist ein Eingriff in die Privatsphäre und in vielen Ländern strafbar, somit ist der Gebrauch solcher Dienste mit Vorsicht zu genießen.

Wird keine Lücke des Systems ausgenutzt und wird der Trojaner auch nicht von Dritten installiert, so muss der Nutzer dazu gebracht werden, den Trojaner selbst zu installieren. Für die meisten mobilen Plattformen bedeutet dies, dass der Trojaner in den Market der jeweiligen Plattform gebracht werden muss. Je nach Market werden Apps unterschiedlich geprüft, so dass die Hürde hierbei unterschiedlich hoch ist, siehe auch Abschnitt 4.4, Sicherheit von Apps. Im Google Market findet beispielsweise keine Vorabüberprüfung von Apps statt, so dass es laut [11] und [60] dort bereits mehrfach dazu kam, dass Trojaner es in den Market geschafft haben.

Trojaner können, unter anderem, die folgenden Schadroutinen mitliefern:

- Überwachung des Datenverkehrs
- Überwachung von Benutzeraktionen
- Anrufen von teuren Telefonnummern oder versenden von Premium-SMS
- Anzeige von Werbung
- Umleitung von Webseiten

Als Nutzer sollten ungewöhnliche Verhaltensmuster des Smartphones untersucht werden, um Schäden durch unentdeckte Spionage-Apps vorzubeugen. Ebenso empfiehlt sich bei Verdacht die Nutzung einer Antivirensoftware, welche bekannte Trojaner anhand der Signatur und einer Blacklist erkennt und unschädlich macht. Siehe hierzu Abschnitt 4.10, Sicherheitssoftware für Smartphones.

5.4 IPv6 und wie daraus Rückschlüsse auf Benutzer gezogen werden können

Das Internet Protokoll in der Version 6 ist seit Dezember 1996 mit der Publikation von RFC 2460 auf dem Standards Track offiziell zum Nachfolger von IPv4 gekürt worden. Grund war die Adressenknappheit von IPv4, da der Arbeitsraum von IPv4 nur ungefähr 4,3 Milliarden Adressen bereit hält. Das rasante Wachstum im Internet erforderte neue Lösungen und so einigte man sich auf das Internet Protokoll Version 6, das einen Arbeitsraum in Höhe von ungefähr 340 Sextillionen Adressen zur Verfügung stellt, was einen Vergrößerungsfaktor von 2^{96} entspricht.

Heutige Smartphones von Apple in der Version 4, sowie Android Geräte ab der Version 2.1, unterstützen das Internet Protokoll v6. Die Telekom, sowie andere Mobilfunkanbieter, kündigten an, im Laufe des Jahres 2011 IPv6, zusätzlich zum alten IPv4, freizuschalten. Leider übertragen die Smartphones eine eindeutige ID, die Rückschlüsse auf den Benutzer zulässt. In der Regel bestimmen Geräte die Hälfte ihrer IPv6-Adresse selbst (den sogenannten Interface Identifier). Doch Smartphones geben sich im WLAN-Netz nur wenig Mühe bei der Erstellung. Sie ergänzen lediglich die weltweit eindeutige MAC-Adresse um zwei immer gleiche Bytes und verwenden sie als Teil der Adresse. Damit übertragen sie bei jedem Kontakt zu einem IPv6-tauglichen Server eine eindeutige Hardware-ID. Da Smartphones in der Regel nur von einer Person genutzt werden, können die Service-Provider und andere Netzbeobachter Rückschlüsse auf die Person und ihr Surfverhalten beziehen. Normalerweise verwenden Desktop PCs Methoden wie den Interface Identifier, um Adressen zufällig und wechselnd zu erzeugen. Diese Sicherheitseinstellung nennt sich „Privacy Extension“ [62] und ist bei Windows ab Werk aktiv. Sie lässt sich auch bei anderen Betriebssystemen nachträglich aktivieren. Leider hat man

bei Smartphones weder die Möglichkeit diese Sicherheitseinstellung zu aktivieren, noch IPv6 abzuschalten. Es fehlt ihnen aber lediglich eine Möglichkeit in der Benutzeroberfläche zur Aktivierung, denn die Privacy Extension ist im Kernel jedes Smartphones enthalten. Mit Hilfe eines Root-Zugangs zum Gerät lässt sich zum Beispiel, wie beim iPhone ab iOS 4 mit dem Befehl

```
sysctl -w net.inet6.ip6.use_tempaddr=1
```

, die Privacy Extension freischalten. Apple hat diese Sicherheitslücke mittlerweile erkannt und in iOS Version 4.3 die Privacy Extension standardmäßig aktiviert, welche statt der Hardware-ID nun regelmäßig wechselnde Zufallszahlen benutzt [25].

5.5 Sicherheit von Ortungsdiensten

Das Global Positioning System (GPS) dient der Ortung der eigenen Position über das Signal vierer Satelliten. Es werden 4 Satelliten benötigt, um Längengrad und Breitengrad, sowie Höhe über Null zu ermitteln. Das vierte Signal ist notwendig, um die Signallaufzeiten bestimmen zu können, da die Uhrzeit der Satelliten ebenfalls unbekannt ist. Ist gerade kein GPS verfügbar, so können zusätzlich die Positionsdaten von WLAN-Hotspots und Funkmasten genutzt werden, um eine gröbere Ortung zu ermöglichen. Da heutzutage jedes Smartphone aus dem Mittelsegement über einen GPS-Empfänger verfügt, können diese Daten genutzt werden um den eigenen Standort zu ermitteln oder ein Navigation-Programme zu nutzen. Allerdings können die Daten auch missbraucht werden um Informationen über den Aufenthaltsort des Benutzers zu enthüllen, oder gar ein vollständiges Bewegungsprofil anzufertigen.

Um dies zu verhindern müssen Apps den GPS-Zugriff bei Googles Android Handy klar kennzeichnen. Die Nutzung der GPS-Daten wird vor Installation der App als benötigtes Recht angezeigt, so dass der Nutzer die Installation abbrechen kann. Apples iOS geht einen ähnlichen Weg: Die App fragt den Benutzer interaktiv mit Hilfe eines Pop-Ups, ob die GPS-Daten verwendet werden dürfen. Somit hat der Nutzer die Möglichkeit, die Nutzung innerhalb der App zu unterbinden, und muss nicht auf die kompletten Funktionen der App verzichten. Desweiteren legt iOS unter *Einstellungen/Ortungsdienste* eine Übersicht aller Apps an, die auf die Positionsdaten des Geräts zugreifen dürfen. Somit kann auch nachträglich kontrolliert und entschieden werden, welche App den Zugriff erhält und welche nicht. [4]

Im April 2011 sorgte eine Meldung über eine Datenschutzpanne in iOS 4 für Aufregung. Laut einem IT-Experten legt iOS seit Version 4 eine Datenbank namens *consolidated.db* auf dem iPhone an. Diese enthalte angeblich sensible Positionsinformationen inklusive Zeitstempel, die das Gerät seit Veröffentlichung der Version 4 angelegt hat. Umso schlimmer war die Nachricht, dass die Informationen unverschlüsselt vorlagen und somit für jeden mit Zugriff auf das Gerät einsehbar waren. Eine von Pete Warden entwickelte Mac Anwendung namens *iPhone Tracker* visualisierte die Daten dieser Datenbank kurze Zeit darauf, um auf das Problem aufmerksam zu machen. Die Daten wurden jedoch nur unscharf abgebildet, um groben Missbrauch zu verhindern. [26]

Aufgrund dieser offenbar erheblichen Datenschutzverletzungen wuchs der Druck auf Apple von verschiedenen Seiten, darunter prominente Datenschützer und die amerikanische Regierung. In Südkorea wurde sogar eine Klage gegen Apple mit einer nicht unerheblichen Schadensersatzforderung im Millionenbereich eingereicht. Diese kam aus dem Zusammenschluss einiger Koreaner zu Stande, die sich in ihren Rechten verletzt fühlten. Vermutungen zur Folge speichere Apple die Daten, um eine Art Geodaten-Datenbank aufzubauen. Diese sollte dazu dienen, Apples Ortungsdienste zu verbessern. Das Problem wurde als „Locationgate-Affäre“ bekannt. [27]

Apple kommentierte die Vorwürfe kurz darauf auf seiner Homepage [2] und stellte den Sachverhalt wie folgt dar: Das iPhone übermittle zwar Positionsdaten, doch handle es sich hierbei um eine anonyme Sammlung

der Positionsdaten von WLAN-Hotspots und Funkmasten, welche Apple dazu dienen würden, die Berechnung der eigenen Position bei ausgeschaltetem GPS drastisch zu beschleunigen. Hierzu werde ein Teil der so gesammelten riesigen Positionsdatenbank auf dem Gerät zwischengespeichert, welche dann als consolidated.db im Backup zu finden ist. Es handelt sich also nicht um die Aufzeichnung der eigenen Koordinaten. Die Dateigröße, sowie die ausbleibende Löschung dieser Datenbank bei Ausschalten der Ortungsdienste, führt Apple auf einen Bug zurück. Apple wolle zur Behebung des Datenschutzproblems

- die Größe der zwischengespeicherten Positionsdaten auf 7 Tage beschränken,
- die Positionsdaten nicht mehr mit einem iTunes Backup sichern,
- die Positionsdaten beim Ausschalten der Ortungsdienste komplett löschen,
- sowie die Positionsdaten auf dem iPhone verschlüsseln, um sie vor Missbrauch zu schützen.

5.6 Vorratsdatenspeicherung

Die Vorratsdatenspeicherung, die durch die EU Richtlinien 2006/24/EG über die Vorratsspeicherung von Daten gefordert wird, sieht das Speichern von Verbindungsdaten aller Mobilfunkgespräche, SMS und E-Mails zur Terrorbekämpfung vor. Ebenso werden Verbindungsdaten von Internet Service Providern gespeichert, die unter anderem die IP-Adresse und die aktuelle Funkzelle des Nutzers enthalten. Hierüber kann ein eindeutiger Rückschluss auf den Benutzer und dessen Aufenthaltsort erfolgen. Die Daten der Vorratsdatenspeicherung dürfen jedoch nicht ohne triftigen Grund eingesehen werden. Dieser ist definiert als erhebliche Gefahr der öffentlichen Sicherheit. Bürger haben die Befürchtung, dass diese Hürde jedoch umgangen wird und Daten auch bei geringeren Straftaten, zum Beispiel File Sharing, in die Hände der Polizei fallen würden. Ebenso muss sichergestellt sein, dass diese Daten nicht durch Zugriff unbefugter Dritter, zum Beispiel Hacker, in falsche Hände fallen, woraus neue Gefahrenquellen eröffnet werden. Somit stellt die Vorratsdatenspeicherung eine sensible Einrichtung in Hinblick auf den Datenschutz der Bürger dar. Die Vorratsdatenspeicherung ist in der jetzigen Form verfassungswidrig und wurde vom Bundesverfassungsgericht am 2. März 2010 außer Kraft gesetzt. Die Regierung arbeitet bereits an einem überarbeiteten Entwurf, der die Richtlinien des Datenschutzes berücksichtigt.

5.7 Diebstahlschutz

Smartphones stellen mittlerweile ein lukratives Ziel für Langfinger dar: Sie speichern Zugangsdaten zu sozialen Netzwerken, persönliche Informationen sowie Passwörter für private und Firmennetzwerke. Zudem sind Smartphones aufgrund ihrer Größe schnell entwendet. Somit muss ein Smartphone bei Diebstahl ausreichend geschützt sein um Angreifern davon abzuhalten an die sensiblen Daten heranzukommen.

Früher reichte hierzu die Eingabe eines vier-stelligen PINs völlig aus. Heute kommt die Verwendung einer Passphrase oder dem Zeichnen eines bestimmten Entsperrungsmusters (nur Android) zum Einsatz. Neuere Android Versionen ab 4.0 lassen sich zudem via Gesichtserkennung sperren.

Leider bieten diese Ansätze nur begrenzten Schutz. So kann bei Androidsystemen über eine USB-Verbindung und der USB-Debugging-Funktion auf einen Teil des Dateisystems zugegriffen und sogar Apps installiert werden. Dies reicht unter Umständen schon aus, um den Sicherheitsmechanismus zu umgehen oder das System mit Schadsoftware zu infizieren. Die Funktion ist standardmäßig jedoch deaktiviert.

Ist ein Gerät einmal entwendet, so erlaubt Apple eine Formatierung beziehungsweise Löschung sämtlicher Daten aus der Ferne. Hierzu kontaktiert das iPhone beim Start einen bestimmten Apple Server. Steht das Gerät auf einer Blacklist geht es sofort in den Formatierungsmodus über und lässt sich nicht mehr starten. Es erfordert somit mehr Geschick um an die Daten zu kommen und den Mechanismus auszuhebeln. Auf Seiten von Google gibt es diese Funktion seit Android 2.2 mit dem Android Device Administration nur in Form

einer API(s. Unterabschnitt 4.7.1, Android). Diese wird von einigen Sicherheits-Apps verwendet und rüstet die Funktion somit auch für Android-Nutzer nach. Als Beispiel sucht die App „WaveSecure“ in eingehenden SMS nach einer bestimmten Parole, welche die Fernlöschung startet. Dies funktioniert jedoch nur solange die SIM-Karte im Gerät verbleibt. Eine Alternative stellt „Lost Phone“ dar, welche beim Wechsel der SIM-Karte automatisch eine SMS an Freunde verschickt. Weitere Informationen zu Sicherheitssoftware finden sich in Abschnitt 4.10, Sicherheitssoftware für Smartphones.

Zum Schutz der Mobilfunknummer kann die SIM-Karte wie bei herkömmlichen Handys über den Mobilfunkprovider gesperrt werden. Hierzu reicht die Mobilfunknummer sowie Identität des Teilnehmers aus. Die SIM-Karte wird dann über die International Mobile Subscriber Identity (IMSI) gesperrt. Zusätzlich kann über die International Mobile Station Equipment Identity (IMEI) das Mobilfunkgerät für die Nutzung mit anderen SIM-Karten unbrauchbar gemacht werden. Die 15-stellige ID findet sich meist in Geräteinformationsseiten unter den Smartphone Einstellungen und sollte vorab notiert werden. Im Folgenden findet sich eine Liste von Informationsseiten der Mobilfunkanbieter zum Verlust der SIM-Karte und den nötigen Service-Adressen.

- T-Mobile `http://www.t-mobile.de/karte-verloren/0,20546,23853-_,00.html`
- Vodafone `https://shop.vodafone.de/service/main.jsp?t=solutionTab&solutionId=1005`
- O2 `http://www.o2online.de/nw/support/mobilfunk/sim/sperren/sim-sperren.html`
- Eplus `http://www.eplus.de/Kontakt-und-Hilfe/Mein-E-Plus/Mein-E-Plus.asp`

Quelle: [19]

5.8 Implementierung von Datenschutzmechanismen

Das folgende Kapitel befasst sich mit aktiven Sicherheitsmaßnahmen der Betriebssystemhersteller zum Schutz persönlicher Daten. Darunter fallen unter anderem die Verschlüsselung des Datenspeichers mit geeigneten Algorithmen, sowie Maßnahmen zum Diebstahlschutz.

5.8.1 Google Android

Android Apps laufen in einer eigenen Sandbox. Dies bedeutet, dass pro App ein Linux-User existiert. Dadurch kann eine App nicht auf die Daten anderer Apps zugreifen. Kommunikation zwischen Apps sind nur über (von Apps festgelegten) Schnittstellen möglich. Die Daten von Apps liegen im Normalfall auf dem internen Speicher des Gerätes. Ob diese Daten verschlüsselt sind, ist von der App abhängig und lässt sich vom Entwickler einstellen. Zusätzlich können Apps Daten auf der SD-Karte ablegen. Jede App, die die Rechte hat auf die SD-Karte zuzugreifen, kann alle Daten auf der SD-Karte lesen. Sollen Daten auf der SD-Karte verschlüsselt werden, so muss dies vom Entwickler selbst implementiert werden. Hier bietet Android nur geringe Unterstützung.

Seit Android 4.0 unterstützt auch Google die vollständige Verschlüsselung des Speichers. Diese muss jedoch manuell vom Nutzer aktiviert werden und lässt sich nicht umkehren. Für die Verschlüsselung des Speichers wird das Passwort der Bildschirmsperre verwendet. Bei Verlust des Schlüssels ist eine komplette Systemrücksetzung erforderlich. Ein Backup ist somit zwingend notwendig. Um bei früheren Android Versionen eine Verschlüsselung der SD-Karte nachzurüsten kann beispielsweise die App Droid Crypt verwendet werden.

Mit der Android Version 4.0 kam zusätzlich eine zentrale Keychain hinzu, die einen zentralen Ort für sensible Daten bieten und diese automatisch verschlüsselt. Somit wird es für Entwickler einfacher sensible Daten, wie Passwörter und Zertifikate sicher zu speichern. die Keychain setzt, wie schon die Verschlüsselung des Systems, auf das Passwort der Bildschirmsperre auf.[33]

Verfügbarkeit	**Dateisystem** *NSFileProtection...*	**Keychain** *kSecAtrtrAccessible...*
Gerät nicht gesperrt	Complete	WhenUnlocked
Nach erstem Öffnen der Datei (im nicht gesperrten Zustand)	UnlessOpen (seit iOS 5)	
Nach erster Eingabe der Codesperre	UntilFirstUserAuthentication (seit iOS 5)	AfterFirstUnlock
Immer	None	Always
Verfügbarkeit auf Gerät beschränkt		WhenUnlockedThisDeviceOnly, FirstUnlockedThisDeviceOnly, AlwaysThisDeviceOnly

Tabelle 5.1: Schutzklassen

Zum Schutz vertraulicher Daten gehört auch die Rechteimplementierung für Zugriff auf Systemfunktionen. So wird bei Installation der App angezeigt, auf welche Systemfunktionen die App Zugriff haben wird. Der nutzen kann dem somit zustimmen, oder es verweigern. Hierunter fallen zum Beispiel Ortungsdaten des GPS-Sensors, das Telefonbuch oder Nachrichtenfunktionen.

5.8.2 Apple iOS

Wie in Unterabschnitt 4.4.2, iOS bereits beschrieben verwendet Apple für die Ausführung von Apps eine Sandbox. Diese verhindert, dass Applikationen ungewollt auf persönliche und sensible Daten zugreifen können. Sie haben nur Zugriff auf den ihnen bereit gestellten Speicher, sowie den zur Verfügung stehenden APIs.

Seit dem iPhone 3GS verschlüsselt Apple zudem die Daten des Flashspeichers mit einem Kennwort und einem 256 Bit Algorithmus basierend auf AES. Bei Verlust des Gerätes können die Daten so durch Löschen des Schlüssels schnell unbrauchbar gemacht werden. Dies ist jedoch weniger sicher als das langsamere Überschreiben mit Nullen durch eine Formatierung, denn Unbefugte können via Jailbreak über die USB-Schnittstelle im DFU-Mode (Device Firmware Update) Flashinhalte im Klartext auslesen.

Seit Version 4 bietet iOS die Apple Data Protection zum Schutz von vertraulichen Informationen wie Emails, Passwörter und Zertifikate durch Verschlüsselung. Voraussetzung hierfür sind Geräte mit integriertem AES-Verschlüsselungschip, zu denen iPads, iPhones ab dem 3GS, und iPods der dritten Generation zählen. Das Benutzerpasswort fließt in die Verschlüsselung mit ein. Das Passwort muss somit ausreichend stark sein, um Schutz zu gewähren.

Die Schutzklassen des Apple Data Protection werden in Tabelle 5.1 aufgelistet.

Für das Dateisystem wird NSFileProtectionComplete empfohlen. Dateien liegen hiermit bei aktiver Codesperre verschlüsselt im Dateisystem vor. Achtung: Eine fehlende Codesperre seitens des Anwenders macht den Schutz zunichte, die Daten liegen dann im Klartext vor. Der Schutz lässt sich auch auf ganze Verzeichnisse innerhalb der Sandbox der Applikation anwenden, was die Anwendung vereinfacht. Probleme kann es geben, wenn Multitasking-Apps Daten im Hintergrund laden und in Dateien schreiben wollen. Es ist also ein Kompromiss zwischen Gebrauchstauglichkeit und Sicherheit zu finden. Zur zusätzlichen Verschlüsselung bietet Apple die CommonCrypto API an, welche unter anderem AES 256 beherrscht.

Die Keychain ist Apples Speicherort für sensible Daten und Passwörter. Hier werden unter anderem Systemrelevante Benutzerdaten hinterlegt mit Passwörtern für E-Mail, Groupware, VPN, WLAN und Webseiten. Zudem können auch Zertifikate von 3rd Party Apps hinterlegt werden. Somit stellt die Keychain ein lukratives Ziel für Einbrecher dar, und ein ebenso schützenswertes Gut für Benutzer. Die Keychain stellt eine auf Basis der Apple Data Protection verschlüsselte relationale Datenbank dar. Die Keychain ist somit ebenso

verschlüsselt wie der restliche Datenspeicher. Die Schutzklassen geben hierbei den grad der Verschlüsselung an. Der Zusatz *ThisDeviceOnly* sorgt dafür, dass Daten nur auf dem Gerät, dass sie verschlüsselt hat, brauchbar sind. Bei einem Geräte-Backup gehen die Informationen verloren. Die höchstmögliche Sicherheit bietet somit *WhenUnlockedThisDeviceOnly*. Es ist zu beachten, dass die Verschlüsselung nicht im Simulator verfügbar ist und auf einem echtem Gerät getestet werden muss. Erlangt ein Einbrecher Zugriff auf die Keychain, so sind Daten mit der Schutzklasse *Always* gefährdet. Nähere Informationen zur Sicherheit der Keychain findet sich in Unterabschnitt 5.2.3, Sicherheit der iOS Keychain.

Sollten Geräte in die falschen Hände fallen, so bietet Apple eine Fernlöschung der Daten durch eine Formatierung des Gerätespeichers an. Des Weiteren lässt sich das Gerät orten. Eine PIN-Eingabesperre verhindert zudem, dass unendlich viele Versuche unternommen werden können. Der Speicher wird nach einer festgelegten Anzahl ebenso gelöscht.

Zur Optimierung der Autokorrektur der integrierten Bildschirmtastatur speichert iOS eingegebene Wörter in einem Keyboard Cache. Dieser kann ein Datenleck darstellen, und kann deshalb über die Option *autocorrectionTyp* mit *UI-TextAutocorrectionTypeNo* verhindert werden. Im Simulator kann dies unter */Library/ApplicationSupport/iPhoneSimulator/<SDKVersion>/Library/Keyboard/*dynamic-text.dat* überprüft werden.

iOS erstellt zum Animieren bei Beendigung einer App einen temporären Screenshot des Fensterinhalts, sobald der Nutzer die Home-Taste betätigt. Dies kann verhindert werden, indem der Fensterinhalt zuvor ausgeblendet wird, oder die Option „Application Does not run in background“ in der Datei info.plist gesetzt wird. Hierbei werden keine Screenshots erstellt.

Seit iOS 3 können Inhalte in die Zwischenablage kopiert werden, um sie an anderer Stelle wieder einzufügen. Um diese Daten zu löschen wenn die App in den Hintergrund wechselt kann die Funktion *pasteBoard.items = nil* verwendet werden.

Seit iOS 5 ist die Synchronisation von App-Daten über iCloud möglich. Apple stellt hierzu eine API bereit, die bei jedem Speichervorgang die Daten an die Cloud sendet. Daten werden hierbei per Secure Socket Layer (SSL) verschlüsselt übertragen. Allerdings können diese Daten auch auf weiteren Geräten landen, sofern diese die selbe Apple ID verwenden. Sensible Daten sollten somit nicht über die Cloud synchronisiert werden.

5.8.3 Windows Phone 7

Windows Phone 7 Apps laufen, ähnlich zu Android und iOS, in ihrer eigenen Sandbox. Microsoft bezeichnet diese Sandbox als `Chamber`, zu deutsch Kammer. Microsoft hat hierzu die Sicherheitsarchitektur des eigenen .net-Frameworks, welches ursprünglich vom Desktop-PC stammt, angepasst. Während das .net-Framework auf Desktop-PCs eine einzige Sandbox für alle Anwendungen gemeinsam bereitstellt, gibt es bei der Sicherheitsarchitektur des .net-Frameworks für Windows Phone 7 für jede App eine eigene Chamber. Jede App verwendet ihren eigenen Prozess, ihren eigenen Speicherbereich und einen eigenen Speicherort für anfallende Daten (isolated storage), und kann somit nicht auf fremde Daten zugreifen. Eine App darf, laut Microsoft, erst nach der konkreten Zustimmung durch den Benutzer auf das Telefon oder die Kurznachrichtenfunktion zugreifen, um SMS zu verschicken. Des Weiteren unterstützt Phone 7 zwar Micro-SD Karten, unterbindet jedoch den Austausch und somit die Erweiterbarkeit. Microsoft vermarktet dies als Sicherheits-Feature. Man möchte damit verhindern, dass man durch das Entnehmen der SD-Karte an die Benutzerdaten gelangt und im Gegenzug Malware einschleppt. Windows Phone 7 überträgt Daten verschlüsselt per Secure Socket Layer (SSL), je nach Serververbindung mit 128 oder 256 Bit. Dadurch kann sich Windows Phone 7 sowohl mit vor Ort befindlichen, als auch Cloud-basierten Diensten wie Exchange Server und SharePoint Server, sicher verbinden.

5.8.4 HP webOS

Da webOS Apps in JavaScript entwickelt und mit Googles Javascript Engine V8 ausgeführt werden, wird kein Maschinencode direkt auf dem System ausgeführt. V8 erzeugt eine Sandbox, die Programme daran hindert, den Speicher oder die Hardware direkt zu manipulieren. Zusätzlich werden einzelne Apps voneinander abgeschottet, indem jede in ihrer eigenen Sandbox läuft. Ein Zugriff auf das System oder zu anderen Apps ist nur über definierte Schnittstellen möglich. [49] Leider hat HP es bislang versäumt, eine Verschlüsselung des Speichers zu implementieren. Für Datensicherheit muss somit auf eine Lösung eines Drittherstellers zurückgegriffen werden. Allerdings bietet HP für webOS-Geräte Mechanismen für verloren gegangene Geräte an, wie zum Beispiel die Fernlöschung des Speichers.

5.8.5 BlackBerry OS

BlackBerry-Geräte sind speziell auf den Betrieb in Unternehmen ausgelegt und bieten einen hohen Sicherheitsstandard. Vor allem bei Verbindungen sowie dem Senden und Empfangen von Daten bemüht sich Hersteller RIM, verschlüsselte Verbindungen zu nutzen. Dies trifft vor allem beim Blackberry Enterprise Server zu, bei dem Verbindungen stets verschlüsselt erfolgen. Richtlinien sorgen dafür, dass Berechtigungen für den Datenzugriff festgelegt werden können. Zudem bietet RIM eine Fernlöschung des Gerätespeichers an, falls Geräte einmal verloren gehen und in falsche Hände fallen sollten. Auf der Entwicklungsseite bietet RIM jedoch weniger Möglichkeiten zur Datensicherheit an als die Konkurrenz. So gibt es kein Sandboxing oder andere zeitgemäße Sicherheitsmaßnahmen zum Schutz des Systems, wie es bei der Konkurrenz Google und Apple bereits der Fall ist. Hier muss RIM somit auf eine gute Prüfung der eigens angebotenen Apps setzen, um Malware von den Geräten fern zu halten.

6 Bewertung

Im folgenden werden die Plattformen Android und iOS als weit verbreiteste Systeme gegenübergestellt. Einzelne Sicherheitsaspekte aus den vorigen Kapiteln werden nochmals für beide Plattformen beleuchtet und bewertet.

6.1 Handhabung

Zur Sicherheit der Systeme zählt letztlich auch die Handhabung der Geräte. Hierbei stellen sich iOS und Windows Phone 7 durch ihre einfache Bedienung heraus, was dazu beiträgt, dass die Geräte von Durchschnittsnutzern auch einfacher abzusichern sind. Im Gegensatz hierzu machen es einem Android und webOS mit ihren offenen Betriebssystemen leichter, zusätzliche Sicherheitsfunktionen zu ergänzen. Somit ergibt sich die Bewertung wie in Tabelle 6.1. Informationen zu den Systemen findet sich in Kapitel 3, Vorstellung der Plattformen.

Kategorie	**Android**	**iOS**	**Windows Phone 7**	**webOS**	**Blackberry OS**
Einfachheit		x	x		
Erweiterbarkeit	x			x	
Gesamt	1	1	1	1	0

Tabelle 6.1: Vergleich Handhabung

6.2 Datenschutz

Diese Kategorie bewertet den Umgang mit, sowie den Schutz persönlicher Daten. Android und iOS bieten eine vollständige Verschlüsselung der Daten an, so dass diese bei Verlust des Gerätes sicher sind. Zusätzliche Diebstahlschutzmechanismen finden sich bei iOS, Windows Phone 7, webOS und Blackberry OS. Alle Plattformen bieten die Möglichkeit einen zusätzlichen Diebstahlschutz zu aktivieren, da dieser bei Android standardmäßig nicht mitgeliefert wird, sondern erst durch Software von Drittherstellern installiert werden muss, erhalten hier alle Plattformen außer Android einen Punkt. Android, sowie Apple bieten mit der Keychain einen zentralen, verschlüsselten Speicher für sensible Daten an. Auf den Konkurrenzplattformen müssen sensible Daten manuell gesichert werden. Die Wertung findet sich in Tabelle 6.2. Informationen zu Datenschutzmaßnahmen der einzelnen Hersteller finden sich in Abschnitt 5.8, Implementierung von Datenschutzmechanismen.

6.3 Softwaresicherheit

Android, iOS, Windows Phone 7 und WebOS setzen Sandboxing und versuchen so Eindringlinge vom System auszuschließen. Laut [7] ist der Einbruch aufgrund von eXecute Never und Code-Signatur in iOS schwieriger. Gelingt dies jedoch, so steht dem Eindringling in iOS mehr Spielraum für Manipulationen zur Verfügung. Zudem sind iOS Apps in Objective-C geschrieben und können hardwarenahen C-Code enthalten. Das Auslesen

Kategorie	Android	iOS	Windows Phone 7	webOS	Blackberry OS
Verschlüsselung	x	x			x
Diebstahlschutz		x	x	x	x
Zentraler Passwort-speicher	x	x			
Gesamt	2	3	1	1	2

Tabelle 6.2: Vergleich Datenschutz

und Manipulieren von Daten fällt bei Android, Windows Phone 7 und webOS hingegen deutlich schwieriger aus, da diese Plattformen keine direkten Prozessor-Instruktionen zulassen, da sie auf Java(Android), C#(Windows Phone 7) und JavaScript(webOS) aufsetzten, die eine zusätzliche Abstraktion vom Betriebsystem bieten. Des Weiteren erschwert Android durch eine Kernelseitige Trennung der Apps den Zugriff auf äußere Prozesse. Weitere Sicherheit bietet die Vorabprüfung von Apps vor der Veröffentlichung im jeweiligen App-Markt, sowie dass der App-Markt die einzige Bezugsquelle für Apps darstellt, so dass diese nicht aus unsicheren Quellen installiert werden können. Tabelle 6.3, zeigt die Bewertung dieser Kategorie. Informationen finden sich in Abschnitt 4.4, Sicherheit von Apps.

Kategorie	Android	iOS	Windows Phone 7	webOS	Blackberry OS
Sandboxing	x	x	x	x	
Prüfung von Apps		x	x		x
Einzige Bezugsquelle		x	x		
Gesamt	1	3	3	2	1

Tabelle 6.3: Vergleich Softwaresicherheit

6.4 Netzsicherheit

Aufgrund der Standardisierung der Übertragungstechnologien gibt es keine Unterschiede unterhalb der verschiedenen Plattformen. Die Sicherheit hängt von der Wahl des Netzes ab. So ist UMTS sicherer als die Verwendung von GSM, WLAN mit WPA2 sicherer als mit WEP. Dieser Aspekt fließt somit neutral in die Bewertung mit ein. Die Plattformen schlagen sich hier also unentschieden. Siehe Abschnitt 4.6, Sicherheit drahtloser Verbindungen.

6.5 Systemsicherheit

Die Sicherheit eines Systems hängt maßgeblich von der Schließung offener Sicherheitslücken ab. In dieser Kategorie werden deshalb die Updatehäufigkeit der Systeme betrachtet, die angibt, wie oft ein Systemhersteller seine Software aktualisiert. Des weiteren beschreibt der Update-Support die Dauer der Unterstützung alter Geräte sowie die zeitnahe Implementierung auf den Endsystemen. Google muss hier auf Grund der Third-Party Gerätehersteller Einbußen hinnehmen, während Apple mit der Unterstützung seiner schmalen Produktpalette punkten kann. Microsoft versucht hingegen, die Gerätehersteller zu regelmäßigen Updates zu bewegen. Unter anderem wurden Anwender über die Seite „Where is my phone update?“ darüber informiert, wann das neuste Update für ihr Gerät verfügbar ist. Nach neuesten Entwicklungen nahm Microsoft die Seite jedoch vom Netz und bekommt somit hier keinen Punkt [38]. Da webOS offiziell von HP Palm eingestellt wurde, finden sich für diese Plattform leider keine Updates mehr. RIM unterstützt ihr System

nur für die aktuellsten Geräte und muss somit ebenfalls einen Punkt einbüßen. Die Wertung findet sich in Tabelle 6.4. Informationen zur Bereitstellung von Updates findet sich in Abschnitt 4.9, Systemupdates.

Kategorie	Android	iOS	Windows Phone 7	webOS	Blackberry OS
Update-Häufigkeit	x	x	x		x
Update-Support		x			
Gesamt	1	2	1	0	1

Tabelle 6.4: Vergleich Systemsicherheit

6.6 Unternehmenstauglichkeit

Alle Systeme sind potentiell für den Einsatz in Unternehmen gedacht und verfügen über Sicherheitsstandards wie eine Fernlöschung oder Geräterichtlinien. Sowohl RIM als auch Microsoft können hier jedoch punkten. Das Blackberry OS ist von jeher auf den Einsatz in Unternehmen ausgelegt und verschlüsselt seine Übertragung zum BlackBerry Enterprise Server vollständig. Microsofts Windows Phone 7 verschlüsselt seine Übertragung je nach Serververbindung mit 128 bzw. 265 Bit Es ergibt sich die Bewertung wie in Tabelle 6.5. Informationen zu Tauglichkeit in Unternehmen findet sich in Abschnitt 4.7, Sicherheit in Unternehmen.

Kategorie	Android	iOS	Windows Phone 7	webOS	Blackberry OS
Fernwartung & Richtlinien	x	x	x	x	x
Vollständig verschlüsselte Übertragung			x		x
Gesamt	1	1	2	1	2

Tabelle 6.5: Vergleich Unternehmenstauglichkeit

6.7 Gesamtbewertung

Aus den obigen Kategorien ergibt sich eine Gesamtbewertung wie in Tabelle 6.6 aufgezeigt. Somit stellt sich Apples iOS als Sieger in Sachen Systemsicherheit heraus, dicht gefolgt von Microsofts Windows Phone 7, Googles Android und Blackbery OS. Apple hat auf Grund der längeren Erfahrung durch die frühe Markteinführung einen Vorsprung auf dem Gebiet Sicherheit. HP Palm kann auf Grund der Einstellung von webOS nicht mithalten. RIM hat zwar langjährige Erfahrung auf dem Gebiet der Unternehmenstauglichkeit, hat es aber verpasst, im Consumerbereich mitzuhalten. Ebenso fehlen Blackberry OS kritische Sicherheitsmaßnahmen wie etwa Sandboxing, mit denen die Konkurrenz aufwartet.

Kategorie	Android	iOS	Windows Phone 7	webOS	Blackberry OS
Handhabung	+	+	+	+	0
Softwaresicherheit	+	+++	+++	++	+
Unternehmens-tauglichkeit	+	+	++	+	++
Systemsicherheit	+	++	+		+
Datenschutz	++	+++	+	+	++
Gesamt	6	10	8	5	6

Tabelle 6.6: Gesamtwertung

7 Fazit und Ausblick

Jede der dargestellten Plattformen hat Schwächen in bestimmten Bereichen, bietet aber auch individuelle Stärken. Auch wenn iOS derzeit einen Vorsprung hat in Sachen Sicherheit - bei der Wahl der geeigneten, beziehungsweise sichersten mobilen Plattform gibt es keine hundertprozentig Lösung. Viele der angesprochenen Sicherheitsaspekte beziehen sich auf alle Plattformen und sind somit nicht plattformabhängig. Laut [59] wächst der Marktanteil der Smartphones mit Windows Phone 7noch dieses Jahr auf 9 Prozent. Android soll seinen Vorsprung auf knapp 54 Prozent weiter ausbauen, iOS bleibt voraussichtlich bei 18 Prozent. 2015 soll Microsoft mit rund 16 Prozent sogar vor iOS liegen und damit auf dem zweiten Platz. Die Marktsituation wird sich somit noch stark ändern - auch neue Angriffsformen werden früher oder später auf allen Plattformen Einzug halten, was die Wahl der Plattform relativiert. Somit liegt die Verantwortung für Schutzmaßnahmen zum größten Teil beim Benutzer, der dies durch sein Nutzungsverhalten beeinflussen muss.
Sind mobile Plattformen somit von Haus aus unsicher? Nein, denn Hersteller wie Apple versuchen durch ihre starken Restriktionen Schadsoftware vom Nutzer abzuwenden. Google übergibt diese Verantwortung zu einem großen Teil an den Benutzer ab, weshalb dieser sich seiner Situation bewusst sein sollte.
Mit ein paar Hintergrundkenntnissen ist es jedoch möglich, die eigene Sicherheit zu erhöhen. Der wichtigste Aspekt ist wohl ein gesundes Maß an Wachsamkeit und Misstrauen, beispielsweise bei den bekannten Phishing-Versuchen durch Schadsoftware. Diese sind sowohl auf Smartphones, als auch auf Desktop-PCs gleichermaßen vertreten. Höhere Risiken bietet beim Smartphone vor allem die Möglichkeit der leichteren Entwendung und der hohe Anteil an gespeicherten persönlichen Daten. Auch hier kann der Nutzer entgegenwirken, indem er nicht jede vertrauliche Information auf seinem Mobilgerät hinterlegt. Bei Angst vor Verlust des Gerätes helfen Backups und gegebenenfalls die Installation einer Sicherheitssoftware zur Verschlüsselung der Daten oder Fernlöschung des Gerätes.
Ist Sicherheitssoftware somit notwendig um mobile Systeme sicher zu machen? Diese Frage lässt sich nicht mit einem klaren „Ja“ oder „Nein“ beantworten. Heutige Smartphones verfügen zwar schon über die nötige Systemleistung um Überwachungssysteme wie Antivirensoftware laufen zu lassen, doch haben diese immer noch einen negativen Einfluss auf die Akkulaufzeit und Leistung des Geräts. Schließlich sollen mobile Geräte auch mobil bleiben und nicht an der Ladestation halt machen. Sicherheitsmechanismen, wie eine Fernlöschung oder Lokalisierung des Gerätes, sind durchaus nützlich, falls diese nicht schon von Haus aus angeboten werden.
Letztendlich gelangt die meiste Software jedoch kontrolliert durch den eigenen Download auf das Handy, und potentielle Spyware lässt sich leider nicht immer eindeutig erkennen. Als Nutzer eines Smartphones sollte man somit Datenschutz ernst nehmen und sich zumindest bei der Installation darüber informieren, welche Rechte eine Software an den eigenen Daten erhält.
Laut Statistik wächst die Anzahl an Angriffen auf mobile Systeme weiter, ebenso die Anzahl der Smartphone-Besitzer, den potentiellen Opfern. Um bei diesem Wettrennen auf der sicheren Seite zu stehen hilft also nur eins: Auf dem neusten Stand zu sein und sich über aktuelle Gefahren zu informieren - denn Wissen ist auf dem Gebiet der Security & Privacy tatsächlich Macht, um ungewollte Angreifer fern zu halten. Als Ergänzung zur Sicherheitsübersicht und -bewertung dieser Studie, folgen in Kapitel 8, Sicherheitstipps deshalb noch einige praktische Maßnahmen, um sich vor Angriffen auf den unterschiedlichen Plattformen zu schützen.

8 Sicherheitstipps

Generell gilt: Smartphones fungieren mit all ihren gespeicherten Daten als moderner Generalschlüssel für das Leben des Nutzers im Web 2.0. Das Schützen des Smartphones vor unbefugten Zugriffen Dritter steht somit an oberster Stelle und sollte nicht zuletzt in Realität durch Achtsamkeit wahrgenommen werden. Smartphones sollten niemals unbeaufsichtigt an öffentlichen Orten liegen gelassen werden, da Langfinger so nur allzu schnell an die begehrte Wahre sowie den persönlichen Daten gelangen können. Doch auch zum eigenen Schutz lässt sich die Bedienbarkeit durch einige Handgriffe sicherer gestalten. Im Folgenden werden einige Punkte aufgelistet, die man als Nutzer des jeweiligen Systems kennen sollte um die Sicherheit zu erhöhen. Diese werden auch in einem Artikel des heise-Verlags beschrieben. [7]

8.1 Allgemeine Tipps

Die folgenden Tipps können auf alle Varianten von Smartphones angewandt werden.

Verbindungen wie WLAN oder Bluetooth sollten nur verschlüsselt erfolgen. UMTS sollte dem eher angestaubten GSM vorgezogen werden, was jedoch in der Regel automatisch geschieht, sofern das Handy dies unterstützt. Bei WLAN empfiehlt sich der Einsatz der Verschlüsselungstechnik WPA2. Unter Bluetooth sollten Authentifizierung und Verschlüsselung möglichst hoch gewählt, sowie die Pairing-Schlüssel geändert werden. Generell sollten ungenutzte Verbindungen bei Nichtgebrauch ausgeschaltet werden, um potentielle Sicherheitslöcher zu schließen. Dies verlängert zudem die Akkulaufzeit. Beim surfen im Internet sollte auf jeden Fall das *HTTPS*-Protokoll verwendet werden, da hier der komplette Datenverkehr verschlüsselt übertragen wird.

Sensible Daten sollten nicht auf Smartphones gespeichert werden, es sei denn es ist unumgänglich. Spätestens dann sollten diese jedoch verschlüsselt abgelegt werden. Dies kann mit einer entsprechenden App leicht nachgerüstet werden, siehe Abschnitt 4.10, Sicherheitssoftware für Smartphones. Zu sensiblen Daten zählen unter anderem Zugangsdaten, Passwörter und Kontodaten sowie Firmendaten.

Neuste Updates sollten bei Erscheinen zeitnah installiert werden. Meist enthalten sie Patches für zuvor bekannt gewordene Sicherheitslücken.

Die IMEI ist die International Mobile Station Equipment Identity, die jedes Mobiltelefon eindeutig identifiziert. Diese sollte für den Notfall, d.h. dem Abhandenkommen des Smartphones durch Diebstahl oder anderweitig, an einem sicheren Ort notiert werden. Beim iPhone findet sich diese unter *Einstellungen / Allgemein*, bei Android unter *Einstellungen / Telefoninfo / Status*. Somit kann beim Verlust des Gerätes nicht nur die SIM-Karte, sondern auch das Mobilgerät über den Provider gesperrt werden (siehe Abschnitt 5.7, Diebstahlschutz).

Online-Banking sollte prinzipiell nicht über das Handy betrieben werden. Banking-Apps könnten Kontodaten speichern und so den Missbrauch fördern. Wird dennoch Online-Banking benötigt, so sollte das Speichern der Kontodaten unterbunden werden sowie ein sicheres Verfahren zur TAN-Übermittlung auf einem zweiten Weg gewählt werden. Die Benutzung von Mobile-TAN auf dem selben Endgerät macht den Schutz durch die TAN unbrauchbar.

8.2 Tipps unter Android

Backups anlegen Anlegen regelmäßiger manueller Backups gegen Verlust der Daten. Hierzu kann die Software *Titanium Backup* verwendet werden.

Synchronisation Regelmäßige oder automatische Synchronisation mit dem Google-Konto. Somit werden Kontakte, E-Mails und Kalendereinträge im Web gesichert. Die automatische Synchronisation kann unter *Einstellungen / Konten & Synchronisation* eingeschaltet werden.

Komplexen Code setzen Einstellen eines Entsperrungsmusters, oder besser, einer Passphrase um das Gerät vor unbefugtem Zugriff zu schützen. Dies kann unter *Einstellungen / Standort & Sicherheit / Display-Sperre einrichten* vorgenommen werden.

USB-Debugging abschalten Abschalten des USB-Debugging Modus unter *Einstellungen / Anwendungen / Entwicklung*, der als Hintertür fungieren kann, sofern gerade nicht auf dem Gerät entwickelt wird.

Diebstahlschutz nachrüsten Fernlöschung/-sperrung durch zusätzliche Apps ermöglichen: WaveSecure, Smrt-Guard, WatchDroid Pro, oder Avast! Mobile Security.

App-Rechte prüfen Rechte von Apps bei Installation prüfen oder nachträglich unter *Einstellungen / Anwendungen / Anwendungen verwalten.*

Unbekannte Quellen meiden Abschalten der Funktion „Unbekannte Quellen“ unter *Einstellungen / Anwendungen* und Apps ausschließlich über den Android Market oder Amazon Market beziehen.

Daten verschlüsseln Da Android alle Daten unverschlüsselt auf der Micro-SD-Karte speichert, kann diese problemlos entfernt und ausgelesen werden. Um dies zu verhindern bietet Android ab Version 4.0 ICS eine komplette Systemverschlüsselung an. Wenn kein Update auf Version 4.0 verfügbar ist kann die SD-Karte mit dem Tool Droid Crypt nachträglich verschlüsselt werden.

8.3 Tipps unter iOS

Die folgenden Tipps wurden mit Hilfe der Quelle [19] und eigenem Wissen aus dem Praxisbetrieb zusammengetragen.

Automatische Sperre aktivieren Aktivierung der automatischen Sperre nach einer bestimmten Zeitspanne unter *Einstellungen / Allgemein / Automatische Sperre.* Diese schützt das Gerät, sollte es einmal eine längere Zeit unbeaufsichtigt herum liegen.

Komplexen Code setzen Einstellen des 4-stelligen PINs, oder besser ab iOS 4, einer Passphrase mit mindestens 6 Stellen. Hierzu muss die Option *Einfacher Code* deaktiviert werden. Dies sichert, unter anderem, die von iOS verwendete Keychain in der sensible Daten hinterlegt sind. Das Passwort kann unter *Einstellungen / Allgemein / Code-Sperre* gesetzt werden.

Daten löschen Setzen der Einstellung, dass alle Daten nach 10 erfolglosen Eingabeversuchen gelöscht werden. Hierbei sollte stets ein funktionierendes Backup vorhanden sein, falls die Funktion versehentlich ausgelöst wird. Zu finden unter *Einstellungen / Allgemein / Code-Sperre / Daten löschen.*

Siri sperren Die Sprachsteuerung des iPhone 4S über die Texterkennung *Siri* ignoriert im Grundzustand die Code-Sperre des iPhones. Zur Sprach(fremd)steuerung genügt ein Druck auf die Home-Taste. Dies kann unter *Einstellungen / Allgemein / Code-Sperre / Bei Codesperre Zugriff auf Siri zulassen* geändert werden.

In-App-Käufe schützen Um sich vor ungewollten In-App-Käufen durch Dritte zu schützen, kann unter *Einstellungen / Allgemein / Einschränkungen* ein weiteres Kennwort zum Schutz vergeben werden.

Backups verschlüsseln Regelmäßige Backups können via Synchronisation mit iTunes an einem PC oder MAC erstellt werden. Um diese vor unbefugtem Zugriff zu schützen, kann iTunes die Backups verschlüsseln. Die Option findet sich in iTunes. Sobald in der linken Spalte das gewünschte Gerät ausgewählt wurde, kann unter *Übersicht* nun *iPhone-Back-up verschlüsseln* gewählt werden.

Ortungsdienste steuern Ortungsdienste sollten nur vertrauenswürdigen Apps zugänglich gemacht werden. Der Zugriff lässt sich unter *Einstellungen / Allgemein / Ortungsdienste* steuern.

iCloud abschotten Durch die Synchronisation mit iCloud werden Daten auf allen Geräten mit gemeinsam genutzter Apple-ID verteilt. Um dies zu verhindern kann unter *Einstellungen / iCloud* ein separater Account pro Gerät eingestellt und verwendet werden.

Ortung aktivieren Zur Verwendung der Diebstahlschutzfunktionen, wie Ortung, Sperre und Fernlöschung, kann Apples Webseite `https://www.icloud.com/` verwendet werden. Für die Ortung muss jedoch zunächst die Option *iPhone suchen* unter *Einstellungen / iCloud* gesetzt werden.

Kein Jailbreak Von einem Jailbreak absehen und nur Apps aus dem offiziellen AppStore installieren. Falls ein Jailbreak durchgeführt wird, so müssen anschließend die Passwörter der Benutzer "root" und "mobile" geändert und gegebenenfalls der SSH-Dienst deaktiviert werden. Siehe hierzu Kapitel Abschnitt 4.8, Sicherheit von Geräten.

8.4 Tipps unter Windows Phone 7

Die folgenden Tipps wurden mit Hilfe der Quelle [44] zusammengetragen.

Windows Live ID erstellen Anwender sollten sich eine kostenlose Windows Live ID zulegen. Dadurch erweitert sich das Live Konto um Mobile-Funktionen. Gestohlene oder verloren gegangene Geräte können dann so lokalisiert und gegebenenfalls gesperrt oder gelöscht werden.

4 stellige Geräte-PIN Diese ist dazu da, Unbefugten den Zugriff auf das Gerät zu verbieten. Dazu wechselt man von der Startseite aus nach rechts zur *Anwendungsliste* und tippt dann auf *Einstellung / Sperre & Hintergrund*.

Updates durchführen Mit der Zusatzsoftware „Zune“ (für Windows) und „Windows Phone 7 Connector“ (für Mac) sollten regelmäßige Updates durchgeführt werden. Diese schließen die meisten aktuellen Sicherheitslücken

Literaturverzeichnis

[1] Android.com. Licenses. Android.com, Januar 2012. http://source.android.com/source/licenses.html.

[2] Apple. Apple q&a on location data. apple.com, April 2011. https://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html.

[3] Apple. icloud. icloud.com, Dezember 2011. http://www.icloud.com.

[4] Apple. ios 4: Grundlagen zu den ortungsdiensten. support.apple.com, November 2011. https://support.apple.com/kb/HT1975?viewlocale=de_DE&locale=de_DE.

[5] Apple. Support - einsatz in unternehmen. apple.com, Oktober 2011. http://www.apple.com/de/support/iphone/enterprise/.

[6] BSI. M 2.384 auswahl geeigneter kryptoverfahren für wlan. BSI, 2009 Dezember. https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02384.html.

[7] Collin Mulliner Daniel Bachfeld. Mobile bedrohungen – spionageangriffe und abzocke auf android und iphone. heise.de, Oktober 2010. http://www.heise.de/mobil/artikel/Spionageangriffe-und-Abzocke-auf-Android-und-iPhone-1103471.html.

[8] Chris Davie. Apple app store has over 500,000 apps. Slash Gear, Dezember 2011. http://www.slashgear.com/apple-app-store-has-over-500000-apps-12201591/.

[9] Michael DeGusta. Android orphans: Visualizing a sad history of support. theunderstatement, Oktober 2011. http://theunderstatement.com/post/11982112928/android-orphans-visualizing-a-sad-history-of-support.

[10] Adam Dickter. Android market marks 10 billion served with dime downloads. www.mobiledevicenow.com, Dezember 2011. http://www.mobiledevicenow.com/article/Android_Market_Marks_10_Billion_Served_with_Dime_Downloads/81267/index.html.

[11] DiePresse.com. Android market leidet wieder unter trojaner-befall. diepresse.com, Juli 2011. http://diepresse.com/home/techscience/mobil/android/677203/Android-Market-leidet-wieder-unter-TrojanerBefall.

[12] Hans-Christian Dirscherl. Sicherheit bei windows phone 7. pcwelt.de, 11 2010. http://www.pcwelt.de/ratgeber/Sicherheit-bei-Windows-Phone-7-1078577.html.

[13] Diverse. Mobilfunknetz. wikipedia.org, Januar 2011. https://de.wikipedia.org/wiki/Mobilfunknetz.

[14] dpa. Google löscht betrügerische android-apps. connect.de, Dezember 2011. http://www.magnus.de/news/google-loescht-betruegerische-android-apps-1223027,293.html.

[15] Elektronik-Kompendium. Ieee 802.11i - wpa/wpa2 - wifi protected access. elektronik-kompendium.de, Januar 2012. http://www.elektronik-kompendium.de/sites/net/0907111.htm.

[16] Skipper Eye. How to secure your jailbroken iphone from ssh hack. www.redmondpie.com, November 2009. http://www.redmondpie.com/how-to-secure-your-jailbroken-iphone-from-ssh-hack-9140084/.

[17] Bundesamt für Sicherheit in der Informationstechnik. Bluetooth - gefährdungen und sicherheitsmaßnahmen. swr3.de, Januar 2003. www.3sat.de/neues/downloads/Bluetooth-BSI.pdf.

[18] Dan Frommer. Apple sues amazon over "app store" name right before amazon's is supposed to launch. Business Insider, März 2011. http://articles.businessinsider.com/2011-03-21/tech/30091398_1_app-store-android-phones-ios-apps.

[19] Jörg Geiger. Die 10 wichtigsten sicherheitstipps für das iphone. www.zehn.de, November 2011. http://www.zehn.de/die-10-wichtigsten-sicherheitstipps-fuer-das-iphone-7510610-0.

[20] Google. Adt plugin for eclipse. Android developers. http://developer.android.com/sdk/eclipse-adt.html.

[21] Hamst0r. Rooten – der jailbreak für android. blog.atomhamster.com, April 2011. http://blog.atomhamster.com/technik/rooten-der-jailbreak-fur-android/.

[22] heise.de. Ungesicherte einsichten – sicherheit von apps für android und iphone, Oktober 2010. http://www.heise.de/mobil/artikel/Sicherheit-von-Apps-fuer-Android-und-iPhone-1103681.html.

[23] heise.de. Cookie-klau durch lücken im android-browser, August 2011. http://www.heise.de/security/meldung/Cookie-Klau-durch-Luecken-im-Android-Browser-1318495.html.

[24] heise.de. Ibm-sicherheitsreport: Mehr komplexe, mehr mobile und mehr zielgerichtete angriffe. heise.de, Oktober 2011. http://www.heise.de/newsticker/meldung/NFC-Standard-erweitert-Datenaustausch-1353164.html.

[25] heise.de. ios 4.3: Apple bessert beim datenschutz nach, März 2011. http://www.heise.de/netze/meldung/iOS-4-3-Apple-bessert-beim-Datenschutz-nach-1206770.html.

[26] heise.de. Mac-software liest gespeicherte iphone-aufenthaltsorte aus. heise.de, April 2011. http://www.heise.de/mac-and-i/meldung/Mac-Software-liest-gespeicherte-iPhone-Aufenthaltsorte-aus-1231120.html.

[27] heise.de. Sammelklage wegen apples "locationgate". heise.de, April 2011. http://www.heise.de/mac-and-i/meldung/Sammelklage-wegen-Apples-Locationgate-1232430.html.

[28] heise.de. ios führt vor blackberry und android in firmennetzwerken. heise.de, Januar 2012. http://www.heise.de/newsticker/meldung/iOS-fuehrt-vor-Blackberry-und-Android-in-Firmennetzwerken-1417450.html.

[29] Steffen Schindler Hendrik Pilz. Are free android virus scanners any good? AV-Test, The Independent IT-Security Institute, November 2011. http://www.av-test.org/fileadmin/pdf/avtest_2011-11_free_android_virus_scanner_english.pdf.

[30] Fraunhofer Institut. ios keychain weakness faq. sit.sit.fraunhofer.de, November 2011. http://sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords-faq.pdf.

[31] Fraunhofer Institut. Lost iphone? lost passwords! sit.fraunhofer.de, Februar 2011. sit.sit.fraunhofer.de/studies/en/sc-iphone-passwords.pdf.

[32] IX. Quellen zum artikel „App-Hilfe". heise.de, Januar 2012. http://www.heise.de/ix/artikel/2012/02/links/044.shtml.

[33] IX. Quellen zum artikel „App-Sicherung". heise.de, Januar 2012. http://www.heise.de/ix/artikel/2012/02/links/052.shtml.

[34] Thomas Joos. Ratgeber: Windows phone 7 für admins. tecchannel.de, August 2011. http://www.tecchannel.de/kommunikation/handy_pda/2036148/ratgeber_microsoft_windows_phone_7_fuer_admins_vor_und_nachteile/.

[35] kaspersky.com. Mobile security 9. heise.de, Oktober 2011. http://www.kaspersky.com/de/kaspersky-mobile-security.

[36] Stefan Keppler. Neue studie liefert zahlen zur verbreitung der smartphones in deutschland. billiger.de, Januar 2011. http://news.billiger.de/

16879-neue-studie-liefert-zahlen-zur-verbreitung-der-smartphones-in-deutschland.html.

[37] Stefan Krempl. 28c3: Neue angriffe auf gsm-handys und schutzmechanismen. heise.de, Dezember 2011. http://www.heise.de/newsticker/meldung/28C3-Neue-Angriffe-auf-GSM-Handys-und-Schutzmechanismen-1401633.html.

[38] Andrea Lira. Keine update details mehr für windows phone 7. pocketpc.ch, Januar 2012. http://www.pocketpc.ch/c/1663-keine-update-details-mehr-fuer-windows-phone-7.html.

[39] Android Market. App-inhaltsbewertungen. Android Market. http://support.google.com/androidmarket/bin/answer.py?hl=de&answer=1075738.

[40] Register of CopyrIghts Marybeth Peters. Recommendation of the register of copyrights. wired.com, Juni 2010. http://www.wired.com/images_blogs/threatlevel/2010/07/dmcaexemps.pdf.

[41] McAfee. Mcafee all access 2012 datenblatt. www.mcafee.com, Oktober 2011. http://home.mcafee.com/store/supportDocs.aspx?prodcode=MAA_IND2011.

[42] Microsoft. Application certification requirements for windows phone. msdn.microsoft.com, Dezember 2011. http://msdn.microsoft.com/en-us/library/hh184843(v=VS.92).aspx.

[43] Microsoft. Clevere entscheidung für unternehmen. www.microsoft.com, Juni 2011. http://www.microsoft.com/windowsphone/de-de/features/business/default.aspx?pf=true.

[44] Microsoft. Tipps zum sicherstellen der handysicherheit. Microsoft, 2011. http://www.microsoft.com/windowsphone/de-de/howto/wp7/basics/tips-to-help-keep-my-phone-secure.aspx.

[45] MobiHand. Mobihand app store. http://onlyblackberry.mobihand.com/appstore.asp, 01 2012. http://onlyblackberry.mobihand.com/appstore.asp.

[46] Palm. Überblick zur sicherheit von palm webos für unternehmen. www.hpwebos.com, Januar 2010. http://www.hpwebos.com/de/de/assets/pdfs/DE_Security.pdf.

[47] Microsoft Phone. Mein windows phone. https://www.windowsphone.com/de-de/my.

[48] Check Point. Check point software technologies ltd. Check Point, 2012. http://www.checkpoint.com/.

[49] Programming4.us. Webos security - introduction to the platform. Programming4.us, Februar 2011. http://programming4.us/mobile/3158.aspx.

[50] QNX. Operating systems, development tools, and professional services for connected embedded systems. QNX, 01 2012. http://www.qnx.com/products/neutrino-rtos/index.html.

[51] Roland Quandt. Windows phone 7: Marktanteil soll nur wenig steigen. winfuture.de, September 2010. http://winfuture.de/news,58009.html.

[52] Roland Quandt. Deutsche hacker knacken gprs-verschlüsselung. winfuture.de, August 2011. http://winfuture.de/news,64849.html.

[53] Parlamentarischen Rat. Grundgesetz für die bundesrepublik deutschland, Mai 1949.

[54] Rednoize. Reverse engineer md5 hashes. rednoize.com, Dezember 2011. http://md5.rednoize.com/.

[55] R.H. iphone gehackt – dailer, trojaner und backdoors kommen. rhde.wordpress.com, November 2009. https://rhde.wordpress.com/2009/11/04/iphone-gehackt-dailer-trojaner-und-backdoors-kommen/.

[56] Volker Riebartsch. Sichere daten am iphone mit code-sperre. www.macwelt.de/, 04 2011. http://www.macwelt.de/artikel/_Ratgeber/376438/sichere_daten_am_iphone_mit_code_sperre.

[57] BalckBerry RIM. Balckberry app world. http://appworld.blackberry.com/webstore/?lang=de, 2011. http://appworld.blackberry.com/webstore/?lang=de.

[58] Mark Zimmermann Ronny Sckmann. App-hilfe. IX, 2:8, Januar 2012.

[59] Arnulf Schäfer. Prognose: Windows phone liegt 2015 vor apples ios. connect.de, Januar 2012. http://www.connect.de/news/prognose-windows-phone-liegt-2015-vor-apples-ios-1234659,912.html.

[60] Lookout Mobile Security. Update: Rufraud: European premium sms toll fraud on the rise. blog.mylookout.com, Dezember 2011. http://blog.mylookout.com/blog/2011/12/11/european-premium-sms-fraud/.

[61] Symantec. Norton mobile security lite. www.symantec.com, Oktober 2011. http://us.norton.com/mobile-security/.

[62] S. Krishnan T. Narten, R. Draves. Privacy extensions for stateless address autoconfiguration in ipv6, September 2007.

[63] T-Online. Keine chance für bluetooth-hacker. handy.t-online.de, Mai 2010. http://handy.t-online.de/bluetooth-handy-vor-hacker-angriffen-sichern-/id_41666228/index.

[64] W3C. Html5 a vocabulary and associated apis for html and xhtml. W3C, 01 2012. http://dev.w3.org/html5/spec/Overview.html.

[65] webgfrast. Authentisierung und verschlüsselung. umtslink.at, Oktober 2011. http://www.umtslink.at/content/authentisierung-75.html.

[66] wikimedia.org. Android (betriebssystem). wikimedia.org, Dezember 2011. https://secure.wikimedia.org/wikipedia/de/w/index.php?title=Android_(Betriebssystem)&oldid=97012219.

[67] wikimedia.org. Blackberry os. wikimedia.org, 01 2012. http://en.wikipedia.org/wiki/BlackBerry_OS.

[68] wikipedia.org. Phishing. wikipedia.org, November 2011. https://secure.wikimedia.org/wikipedia/de/wiki/Phishing.