Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit Anwendungen 1

Peter Salchow

Sicherheit in Android

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Peter Salchow

Sicherheit in Android

Seminarausarbeitung im Rahmen der Veranstaltung Anwendungen 1
im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft

Abgegeben am 24. Juli 2008

Inhaltsverzeichnis

Abbildungsverzeichnis **4**

1 Einführung **5**
1.1 Motivation 5
1.2 Zielsetzung 6

2 Aufbau von Android **7**
2.1 Was ist Android? 7
2.2 Architektur von Android 8
2.2.1 Application-Schicht 9
2.2.2 Framework-Schicht 9
2.2.3 Library-Schicht 11
2.2.4 Kernel-Schicht 12

3 Sicherheit in Android **14**
3.1 Sicherheitsfeatures 14
3.1.1 Anwendungsisolation 14
3.1.2 Datenzugriff und UserID 15
3.1.3 Rechtevergabe 15
3.1.4 Datenverschlüsselung 16
3.2 Schwachstellen 16
3.2.1 Datenablage 16
3.2.2 Apache-Lizenz 17
3.2.3 Zertifizierung von Anwendungen 17
3.2.4 Zugriff über JNI 17
3.3 Bewertung der Sicherheit 17

4 Fazit und Ausblick **19**
4.1 Fazit 19
4.2 Ausblick 19

Literaturverzeichnis **21**

Abbildungsverzeichnis

2.1 Architektur von Android [vgl. Google] 8
2.2 Application-Schicht in der Android Architektur [vgl. Google] 9
2.3 Framework-Schicht in der Android Architektur [vgl. Google] 9
2.4 Lifecycle-Kontrolle des Activity-Managers [vgl. Google] 10
2.5 Bibliotheken in der Library-Schicht [vgl. Google] 11
2.6 Android-Runtime in der Library-Schicht[vgl. Google] 12
2.7 Kernel-Schicht in der Android Architektur [vgl. Google] 13

1 Einführung

1.1 Motivation

Mobile Geräte sind in unserer Gesellschaft sehr stark verbreitet. Dazu gehören unter anderem Digitalkameras, Navigationsgeräte und Mobiltelefone. Besonders weit verbreitet sind Mobiltelefone. Ihr Absatz steigt momentan stark an, da der Bedarf nach mobiler Nutzung des Internets in der Gesellschaft zunimmt. Dieser Bedarf ist einerseits in der Kommunikation und andererseits in der Informationsbeschaffung begründet. Mobiltelefone bilden somit eine sehr solide Grundlage, um mobile Anwendungen zu verbreiten. Innovative Anwendungen, die zukünftig entwickelt werden, können sehr komfortable Dienste für die Anwender zur Verfügung stellen. Durch die Vielzahl an Anwendern entstehen neue Bedürfnisse, die die Entwicklung weiterer Anwendungen fördert.

Der aktuelle Stand am Markt ist es, dass viele Mobiltelefone von unterschiedlichen Herstellern existieren. Jeder Hersteller stellt meist eine eigene, speziell angepasste, Plattform für die Anwendungen zur Verfügung. Die Anwendungsentwickler müssen ihre Anwendungen daher in einer sehr heterogenen Landschaft entwickeln. Damit die Anwendungen von einer breiten Masse akzeptiert werden, ist es notwendig, dass die Anwendungen auf nahezu allen Geräten ausgeführt werden können. Eine Anwendung muss daher für verschiedene Plattformen entwickelt und angepasst werden. Dabei kommt erschwerend hinzu, dass die Anwendungsentwickler oftmals nicht die volle Kontrolle über das Gerät haben und somit nicht alle Funktionen ausnutzen können. Auf den einzelnen Geräten wird zwischen nativen Anwendungen und Anwendungen von Drittanbietern unterschieden.

Um diese Hürden zu umgehen hat die „Open Handset Alliance“ das Betriebssystem Android entwickelt, welches mit sehr vielen Mobiltelefonen ausgeliefert werden soll. Mit Hilfe dieses Betriebssystems soll der Markt der Mobiltelefone für neue Anwendungen geöffnet werden. Android ist ein Betriebssystem, dass auf nahezu allen Mobiltelefonen eingesetzt werden kann indem es von der darunter liegenden Hardware abstrahiert und somit den Anwendungsentwicklern eine einheitliche Plattform zur Verfügung stellt. Ein solches Betriebssystem würde die Anwendungsentwicklung für mobile Geräte stark vereinfachen und dadurch die Entwicklungskosten für mobile Anwendungen senken. Durch den breiten Einsatz eines Betriebssystems würden aber auch die Kosten für Mobiltelefone sinken und Anwendungen könnten sehr flächendeckend ausgeliefert werden.

1.2 Zielsetzung

Ziel dieser Ausarbeitung ist es, das Thema Sicherheit in dem Betriebssystem Android näher zu untersuchen. Durch die verstärkte Nutzung von mobilen Geräten und der darauf laufenden Anwendungen, steigt auch die Relevanz der auf dem Gerät gespeicherten Daten. Um diese Daten vor dem Missbrauch durch Dritte zu schützen, müssen sie zum Beispiel verschlüsselt übertragen und gespeichert werden. Zusätzlich muss sichergestellt werden können, dass nur Personen mit bestimmten Rechten Zugriff auf das System bekommen. Um all diesen Anforderungen gerecht werden zu können, sollte das Betriebssystem einheitliche Mechanismen im Bereich Security implementieren und den Anwendungen zur Verfügung stellen.

In welchem Umfang der Bereich Security in Android realisiert wurde, soll in dieser Arbeit analysiert werden. Darüber hinaus soll festgestellt werden, welche Risiken durch die Architektur von Android entstehen können und welche potenziellen Angriffsmöglichkeiten sich daraus ergeben können.

2 Aufbau von Android

In diesem Kapitel wird zunächst der Aufbau von Android näher beschrieben. Dazu wird zuerst ein allgemeiner Überblick über Android vermittelt. Anschließend soll die Architektur von Android aus Softwaretechnischer Sicht genauer untersucht werden. Die weitere Arbeit, speziell die Betrachtungen der Sicherheit (s. Kapitel 3), bauen hierauf auf.

2.1 Was ist Android?

Im allgemeinen kann gesagt werden, dass Android ein Software-Stack für mobile Geräte ist. Android besteht aus folgenden Komponenten:

- Betriebssystem
- Middleware
- Anwendungen

Diese Komponenten sind in Schichten angeordnet [vgl. OHA a]. Der genaue Aufbau und die Funktionsweise der einzelnen Schichten werden in Kapitel 2.2 beschrieben.

Mit Android wird auf mobilen Geräten eine Umgebung zum Ausführen von Anwendungen zur Verfügung gestellt. Diese Umgebung abstrahiert von der darunter liegenden Hardware und stellt den Anwendungen Schnittstellen für den Zugriff auf das Gerät zur Verfügung. Alle Anwendungen werden in Java geschrieben [vgl. Google].

Momentan wurde noch kein Gerät mit Android ausgeliefert. Der komplette Stack kann vorerst nur im Emulator ausgeführt werden. Somit müssen auch neue Anwendung im Emulator ausgeführt und getestet werden. Das Vorhaben, erste Geräte mit Android im dritten Quartal 2008 auf den Markt zu bringen, kann scheinbar nicht mehr verwirklicht werden. Die ersten Auslieferungen sind für das vierte Quartal 2008 angekündigt [vgl. heise 2008].

Android wird von der „Open Handset Alliance“ entwickelt. Dies ist ein Zusammenschluss von momentan 34 Mitgliedern. Dazu gehören unter anderem T-Mobile, Samsung und ebay. Der Initiator und somit auch Mitglied dieses Zusammenschlusses ist Google. Die Ziele, die die „Open Handset Alliance“ mit der Entwicklung von Android verfolgt, sind die Bereitstellung

eines kostenlosen Betriebssystems für mobile Geräte, eine geräteunabhängige Plattform zu entwickeln und eine schnelle und einfache Anwendungsentwicklung zu fördern. Die „Open Handset Alliance“ stellt Android unter der Apache-Lizenz v2 zur Verfügung [vgl. OHA b].

2.2 Architektur von Android

Wie bereits in Kapitel 2.1 erwähnt, besteht Android aus verschiedenen Komponenten, die in Schichten angeordnet sind. In der Gesamtarchitektur ist zu erkennen, dass Android aus vier Schichten besteht. Das sind die Application-Schicht, die Framework-Schicht, die Library-Schicht mit der Android-Runtime und die Kernel-Schicht. Abbildung 2.1 zeigt die Gesamtarchitektur von Android mit der Anordnung der zuvor genannten Schichten. Die blau gefärbten Komponenten sind in Java geschrieben.

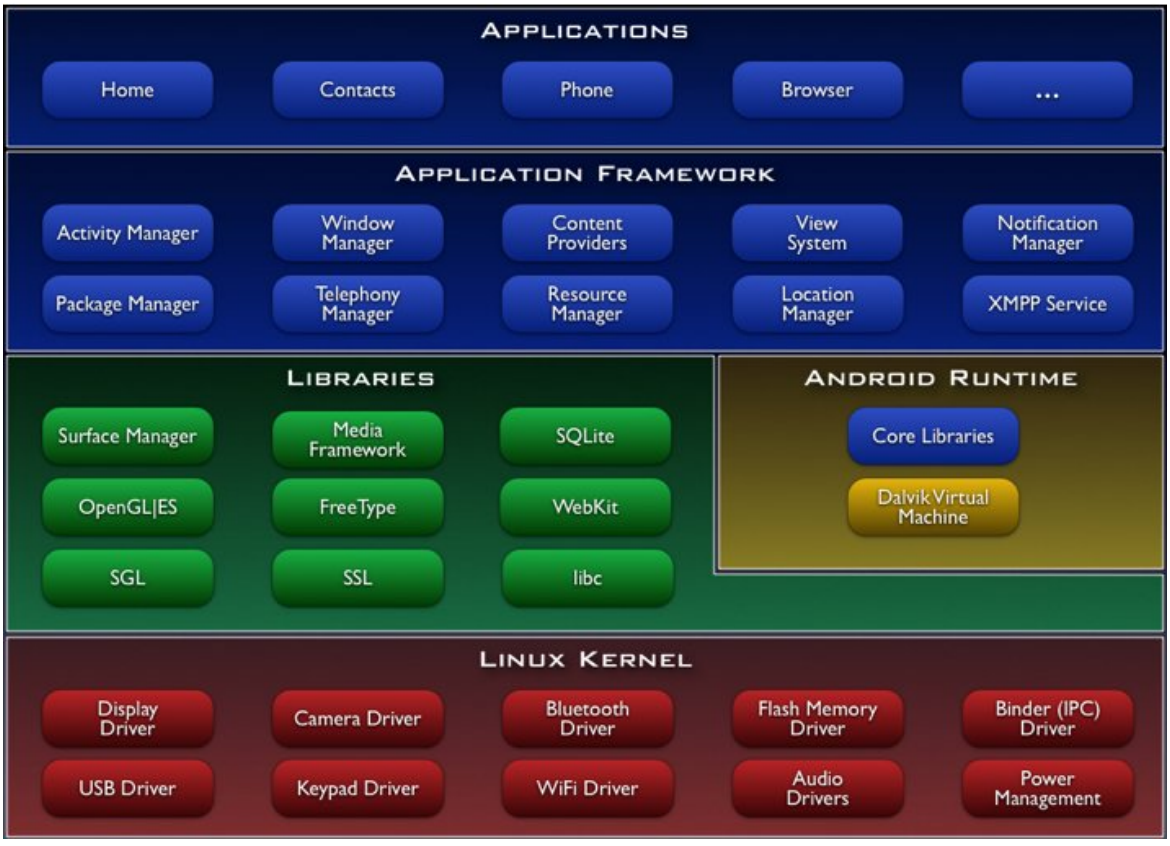

Abbildung 2.1: Architektur von Android [vgl. Google]

2.2.1 Application-Schicht

Die oberste Schicht im Android Software-Stack ist die Application-Schicht. In dieser Schicht sind alle Anwendungen angesiedelt, die in Android ausgeführt werden. Dazu gehören sowohl die Core-Anwendungen, die bereits mit Android ausgeliefert werden, als auch die neu entwickelten Anwendungen. Zu den Core-Anwendungen gehören zum Beispiel E-Mail-Client, Kalender und Telefon. Abbildung 2.2 zeigt den Schematischen Aufbau der Application-Schicht und den in ihr enthaltenen Core-Anwendungen.

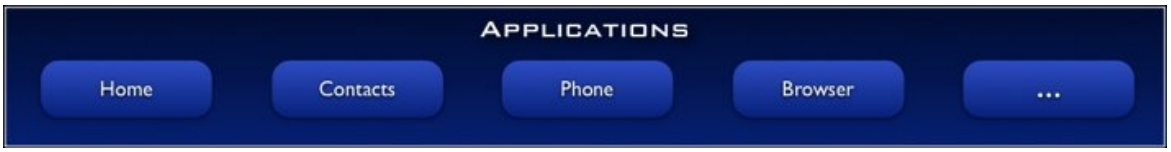

Abbildung 2.2: Application-Schicht in der Android Architektur [vgl. Google]

In Android existiert keine Trennung mehr zwischen nativen Anwendungen und neuen Anwendungen. Wie die Architektur verdeutlicht befinden sich beide in der selben Schicht. Somit werden sie vom System auch gleich behandelt und erhalten auch Zugriff auf die selben Schnittstellen.

2.2.2 Framework-Schicht

Unter der Application-Schicht befindet sich die Framework-Schicht. Abbildung 2.3 zeigt den Aufbau der Schicht und die in ihr enthaltenen Komponenten. Android ist so konzipiert, dass alle Anwendungen auf eine gemeinsame Schnittstelle zugreifen bzw. in der gleichen Umgebung ausgeführt werden. Diese Funktion wird von der Framework-Schicht übernommen. In Abbildung 2.3 ist zu erkennen, dass auch diese Schicht und ihre Komponenten blau eingefärbt und somit in Java geschrieben sind.

Abbildung 2.3: Framework-Schicht in der Android Architektur [vgl. Google]

Eine Hauptfunktionalität dieser Schicht ist es, über den Window-Manager, den Anwendungen Views zur Verfügung zu stellen. Darüber hinaus übernimmt das Framework die Lifecycle-Kontrolle. Der Activity-Manager startet die Anwendungen und verwaltet ihren Zustand. Beim Auftreten bestimmter Ereignisse (z. B. Anwendung in den Hintergrund legen oder beenden) informiert dieser die entsprechenden Anwendung indem Methoden aufgerufen werden. Abbildung 2.4 verdeutlicht den Lifecycle einer Anwendung (Activity). Es wird gezeigt, in welchen Zuständen sich die Anwendung befindet (Ovale) und wann welche Callback-Methoden aufgerufen werden (Rechtecke).

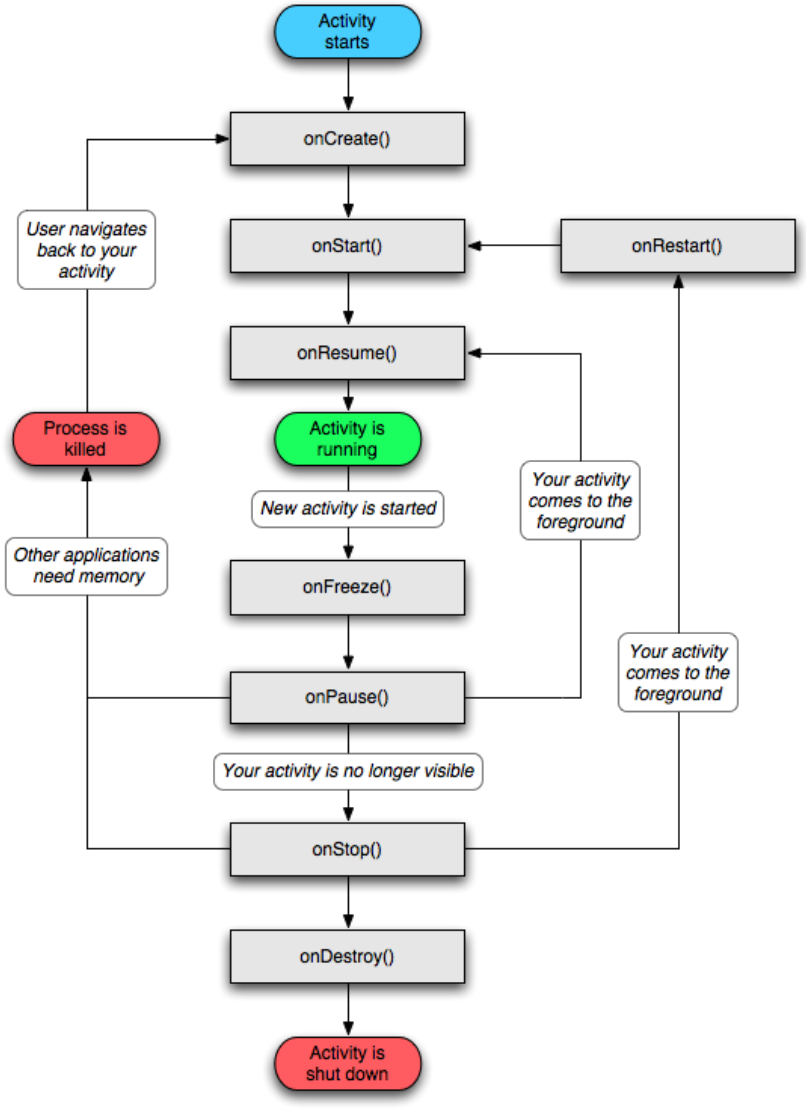

Abbildung 2.4: Lifecycle-Kontrolle des Activity-Managers [vgl. Google]

Des Weiteren regelt das Framework die Kommunikation zwischen den einzelnen Anwendungen. Diese Anwendungen können ihre Ressourcen für andere Anwendungen zur Verfügung stellen. Das Framework übernimmt dann die Zugriffskontrolle, indem die Rechte bestimmter Anwendungen für die zu benutzenden Funktionen überprüft werden. Dadurch wird die Wiederverwendung von Komponenten gewährleistet. Auf diesem Wege können andere Anwendungen beispielsweise auf ein gemeinsames Adressbuch zugreifen, oder die Funktionen des Kalenders mit benutzen [vgl. Google].

2.2.3 Library-Schicht

Die Library-Schicht in Android besteht aus zwei Teilen. Das ist zum Einen die Android-Runtime-Umgebung und zum Anderen eine Ansammlung von Bibliotheken.

Libraries

Unterhalb des Frameworks befinden sich die Bibliotheken, die in Android bestimmte Kernfunktionalitäten zur Verfügung stellen. Diese Bibliotheken wurden in C und C++ geschrieben und werden von vielen anderen Komponenten genutzt. Entwickler von Anwendungen erhalten nur über das Framework Zugriff auf die Bibliotheken. Dadurch werden alle Funktionen gekapselt und der Zugriff wird überwacht. Abbildung 2.5 zeigt eine Auswahl an Bibliotheken in der Library-Schicht.

Abbildung 2.5: Bibliotheken in der Library-Schicht [vgl. Google]

So ist zum Beispiel SQLite als relationales DBMS in den Bibliotheken enthalten. Anwendungen können dadurch, mit Hilfe des Frameworks, ihre Daten in einer Datenbank ablegen. Der Zugriff auf die Daten wird vom Framework kontrolliert. Weiterhin sind die Standard C-Library und der Surface-Manager, zur Darstellung von 2D- und 3D-Grafiken, in dieser Schicht enthalten.

Android-Runtime

Die Android-Runtime-Umgebung führt die einzelnen Anwendungen in Android aus. Sie besteht aus der Dalvik-Virtual-Mashine und den entsprechenden Java-Klassenbibliotheken (Core-Libraries). Abbildung 2.6 zeigt den Aufbau der Android-Runtime-Umgebung.

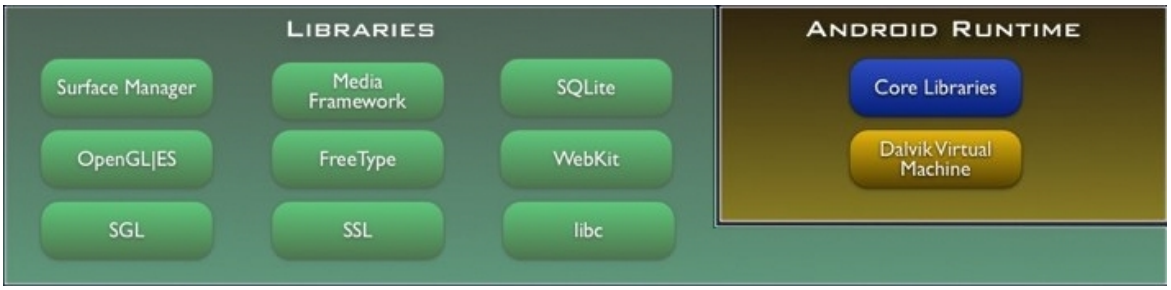

Abbildung 2.6: Android-Runtime in der Library-Schicht[vgl. Google]

Die Dalvik-VM is eine spezielle VM die für mobile Geräte optimiert wurde [vgl. Google]. Dabei wurde besonders die begrenzte Speicher- und Rechenkapazität berücksichtigt, sowie auf geringen Stromverbrauch geachtet. Im Gegensatz zur gewöhnlichen Java-VM basiert die Dalvik-VM auf einem Registerautomaten und nicht auf einem Kellerautomaten. Die Dalvik-VM führt aber auch Bytecode aus. Die vom Java-Compiler erzeugten *class*-Dateien müssen für die Dalvik-VM, mithilfe eines Cross-Compilers in einen speziellen Bytecode (Dalvik-Executables) überführt werden.

Eine Besonderheit in Android ist, dass jede Anwendung in einer eigenen Instanz der Dalvik-VM ausgeführt wird [vgl. Google]. Dazu wird von jeder Instanz sowohl das Framework, als auch die entsprechende Anwendung geladen. Die VM ist aber so optimiert, dass dabei vom Framework nur die benötigten Komponenten geladen werden. Das Threading und das Low-Level Memory-Management wird dabei der darunter liegenden Schicht, dem Linux-Kernel, überlassen (s. Kapitel 2.2.4).

2.2.4 Kernel-Schicht

Die unterste Schicht im Android-Stack ist die Kernel-Schicht. In der aktuellen Version von Android (m5-rc15) wird der Linux-Kernel in der Version 2.6 als Grundlage eingesetzt. Abbildung 2.7 veranschaulicht den schematischen Aufbau der Schicht mit einigen ihrer Komponenten.

Abbildung 2.7: Kernel-Schicht in der Android Architektur [vgl. Google]

Die Aufgabe des Kernels ist es, von der Hardware des Gerätes zu abstrahieren und dem restlichen Software-Stack eine einheitliche Umgebung zur Verfügung zu stellen. Der Linux-Kernel enthält die Gerätetreiber für die im Gerät befindliche Peripherie. Außerdem übernimmt der Kernel die Prozessverwaltung und das Scheduling, sowie das Speichermanagement. Auch der Network-Stack ist im Kernel implementiert.

3 Sicherheit in Android

Im folgenden Kapitel soll die Sicherheit in Android genauer untersucht werden. Dabei werden besonders die Eigenschaften der Architektur des Software-Stacks berücksichtigt. Es werden sowohl die Sicherheitsfeatures, als auch einige Schwachstellen vorgestellt und erläutert. Am Ende des Kapitels wird die Sicherheit noch einmal abschließend bewertet.

3.1 Sicherheitsfeatures

Bei der Konzeption von Android wurden viele Sicherheitsaspekte berücksichtigt. Einige dieser Aspekte, die mit in die Architektur eingeflossen sind, sollen an dieser Stelle vorgestellt werden.

3.1.1 Anwendungsisolation

Ein sehr wichtiger Aspekt ist die gegenseitige Isolation der einzelnen Anwendungen während der Ausführung. Dies ist wichtig, damit jede Anwendung ihren eigenen Speicherbereich exklusiv nutzen kann und auch während der Ausführung nicht von anderen Anwendungen unerwartet beeinflusst wird. Dazu gehört beispielsweise das Beenden von Anwendungen außerhalb der Lifecycle-Kontrolle des Frameworks (s. Kapitel 2.2.2).

Um den Aspekt der Anwendungsisolation umsetzen zu können, wird jede Anwendung, wie bereits in Kapitel 2.2.3 vorgestellt, in einer eigenen VM ausgeführt. Somit arbeitet jede Anwendung mit einer eigenen Instanz des Frameworks und der Android-Runtime. Dies hat zwar zum Nachteil, dass mehr Speicherplatz benötigt wird. Aber aus dem Blickwinkel der Sicherheit ist dieses Vorgehen als positiv zu werten, da fehlerhafte Anwendungen nicht das gesamte System blockieren können. Die Kontrolle über jede einzelne Anwendung bleibt beim System.

3.1.2 Datenzugriff und UserID

Ein weiteres Feature in Android ist die Kontrolle des Datenzugriffs. Alle Daten, die von einer Anwendung geschrieben werden, sind für andere Anwendungen nicht sichtbar. Das System überwacht den Zugriff und gibt nur die Daten für die entsprechende Anwendung frei.

Dieses Konzept wird über eine UserID realisiert. Jeder Anwendung wird bei der Installation eine eigene UserID zugewiesen. Die UserID bleibt konstant, solange die Anwendung auf dem Gerät installiert ist [vgl. Google]. Die Prozesse, in denen die Anwendungen ausgeführt werden, laufen mit dieser UserID. Beim Schreiben von Dateien, wird jeder Datei die UserID der entsprechenden Anwendung zugewiesen. Über dieses Attribut regelt der Linux-Kernel den Zugriff der einzelnen Prozesse auf die zugehörigen Daten. Für die Umsetzung dieses Konzepts wurde einfach der bereits im Linux-Kernel integrierte Dateisystemmechanismus verwendet.

Bei Bedarf ist es möglich, dass mehreren Anwendungen die gleiche UserID zugewiesen wird. Dazu muss der *sharedUserId*-Tag im Deployment-Descriptor der Anwendung gesetzt werden. Die Anwendungen werden dann vom System mit gleichen Rechten ausgestattet und gleich behandelt. Damit dieses Konzept nicht missbraucht werden kann, müssen die entsprechenden Anwendungen die gleiche Signatur besitzen, um die gleiche UserID zu bekommen [vgl. Google]. Dateien können aber auch über bestimmte Flags für andere Anwendungen freigegeben werden.

3.1.3 Rechtevergabe

Anwendungen in Android können ihre Funktionalität für andere Anwendungen als Dienst zur Verfügung stellen. Beispielsweise können andere Anwendungen den SMS-Dienst der SMS-Anwendung nutzen. Damit diese Dienste genutzt werden können, müssen die Anwendungen die entsprechenden Rechte beantragen. Dies geschieht im Deployment-Descriptor. In Android wird bei der Vergabe der Rechte der pessimistische Ansatz verfolgt. Eine Anwendung kann solange keine Dienste nutzen, wie diese nicht beantragt wurden. Während der Installation einer Anwendung kann der Benutzer den einzelnen Dienstnutzungen zustimmen, oder die Rechte dafür entziehen. Während der Ausführung wird der Benutzer dann nicht wieder gefragt. Die Verantwortung über die Dienstnutzung wird somit an den Benutzer übertragen. Bietet eine Anwendung einen neuen Dienst an, so können für diesen Dienst eigene Rechte definiert werden.

3.1.4 Datenverschlüsselung

Android unterstützt die Verschlüsselung von Daten. Es besteht beispielsweise die Möglichkeit zur verschlüsselten Datenübertragung über SSL. Um diese Funktionalität zur Verfügung zu stellen, sind die Pakete *javax.net.ssl* und *javax.security* standardmäßig in den Java-Klassenbibliotheken von Android integriert. Sie ermöglichen es auch, beliebige Daten auf Anwendungsebene zu verschlüsseln und zu signieren.

3.2 Schwachstellen

Neben den zuvor genannten Sicherheitsfeatures, die Android anbietet, existieren auch noch einige Schwachstellen. Diese sollten beim Umgang mit Android unbedingt berücksichtigt werden. Einige der Schwachstellen sind aufgrund mangelnder oder fehlender Implementierungen entstanden, andere ergeben sich durch das grundlegende Konzept von Android.

3.2.1 Datenablage

Anwendungen in Android können Daten entweder im Dateisystem oder über die Datenbank SQLite abspeichern. Sollen die Daten als Datei im Dateisystem abgelegt werden, bietet Android keine performante Lösung zur verschlüsselten Datenablage. Bei Verlust oder Diebstahl des Gerätes können die Daten entweder durch andere Applikationen oder aber direkt aus dem Speicher ausgelesen werden. Falls man Root-Rechte für das System erlangt, kann auf die Dateien aus jeder beliebigen Anwendung heraus zugegriffen werden.

Ähnlich Verhält es sich momentan noch mit der Datenablage in SQLite. Bislang wurde noch kein Krypto-Modul für SQLite in Android eingebunden. Der Grund hierfür ist, dass im Moment keine frei zugänglichen Krypto-Module für SQLite existieren und von einer Eigenentwicklung vorerst abgesehen wird, da die Android-Entwickler die Verschlüsselung für unwichtig erachten [vgl. Google Issue 191].

Um dieses Problem zu umgehen, könnte man die Daten direkt in der Anwendung verschlüsseln (s. Kapitel 3.1.4). Dieser Ansatz ist allerdings sehr rechenintensiv und die Verschlüsselung muss in jeder Anwendung neu implementiert werden.

3.2.2 Apache-Lizenz

Eine weitere Schwachstelle ergibt sich durch die Veröffentlichung von Android unter der Apache-Lizenz. Das eigentliche Vorhaben ist es, die Sicherheit durch Open-Source zu verbessern. Dies wird erreicht, da der Source-Code von vielen unabhängigen Entwicklern eingesehen und überprüft wird. Drittanbieter können durch Open Source Programmteile modifizieren oder ganze Komponenten durch eigene Implementierungen ersetzen. Durch die Apache-Lizenz wird es jedoch ermöglicht, dass Veränderungen und neue Komponenten nicht freigegeben werden müssen. Besonders bei sicherheitsrelevanten Komponenten ist das Verbergen der Implementierungen kritisch zu bewerten und kann sich unter Umständen negativ auf die Sicherheit auswirken.

3.2.3 Zertifizierung von Anwendungen

Für Anwendungen ist in Android ein Zertifizierungsprozess vorgesehen, der die Qualität der Anwendungen und deren Herkunft sicherstellen soll. Aus sicherheitstechnischer Sicht ist eine Zertifizierung von Anwendungen sehr wichtig, da sichergestellt wird, dass die Funktionen der Anwendungen ausreichend getestet wurden. Bislang hat Google jedoch noch kein Konzept für diesen Zertifizierungsprozess vorgestellt. Eine abschließende Bewertung der Qualität der Zertifizierung ist zu diesem Zeitpunkt noch nicht möglich. Anwendungen können vorerst nicht zertifiziert werden.

3.2.4 Zugriff über JNI

Android erlaubt die Verwendung von JNI bei der Anwendungsentwicklung. Darüber könnte es möglich sein, die in Kapitel 2.2.3 beschriebenen Bibliotheken direkt zu laden und Funktionen daraus auszuführen. Auf diesem Weg wäre es dann möglich, die im Framework implementierten Sicherheitsmechanismen zu umgehen und somit das gesamte Sicherheitskonzept in Android außer Kraft zu setzen. Ob die JNI-Funktionalität in Android auf dem zuvor beschriebenen Weg genutzt werden kann, konnte im Rahmen dieser Ausarbeitung noch nicht festgestellt werden. Für die Entwicklung sicherheitskritischer Anwendungen sollte dieser Punkt noch einmal genauer untersucht werden.

3.3 Bewertung der Sicherheit

Zum Thema Sicherheit in Android kann abschließend gesagt werden, dass in Android gute Konzepte für die Rechtevergabe und den Dateizugriff existieren. Auch die Isolation der An-

wendungen und die zentral kontrollierte Kommunikation über Dienste wurde gut umgesetzt. Allerdings bestehen im Bereich Datenverschlüsselung große Lücken, da kein zentrales Konzept hierfür in Android existiert.

Darüber hinaus bleibt anzumerken, dass die Vergabe von Zugriffsrechten auf Dienste (s. Kapitel 3.1.3) auch nicht als sehr sicher eingestuft werden kann. Durch die Zustimmung des Benutzers kann eine Anwendung Zugriff auf alle Dienste des Gerätes erlangen. Beispielsweise können, nach der Freigabe durch den Benutzer, Daten über das Internet gesendet und empfangen werden. Wohin die Daten während der Ausführung der Anwendung gesendet werden, kann nicht weiter eingesehen und eingeschränkt werden. Die Vergabe von Rechten ist in diesem Fall zu grobgranular.

Des Weiteren können sich Anwendungen auf bestimmte Ereignisse (z. B. Tastendruck, Anruf) registrieren und die erhaltenen Daten weiterverarbeiten. Wie diese Daten verarbeitet werden, bleibt vor dem Benutzer verborgen.

Diese Bewertung der Sicherheit zeigt, dass viele Konzepte zur Steigerung der Sicherheit in Android umgesetzt wurden. Allerdings wurden nicht alle diese Konzepte ausreichend analysiert. Somit birgt die Architektur von Android, besonders für unerfahrene Benutzer, noch viele Risiken.

4 Fazit und Ausblick

4.1 Fazit

Ziel dieser Arbeit war es, Android als Betriebssystem für mobile Geräte vorzustellen und dabei besonders die Aspekte der Sicherheit näher zu untersuchen. Da Android sich noch in einer sehr frühen Entwicklungsphase befindet und noch keine Geräte mit Android ausgeliefert wurden, lassen sich nur wenige fundierte Quellen zu diesem Thema finden. Um detaillierte Erkenntnisse zum Thema Sicherheit zu erlangen, ist es notwendig, mögliche Angriffspunkte aufzudecken und diese anhand geeigneter Szenarien im Emulator nachzustellen. Darüber hinaus kann es dabei hilfreich sein, den Android Source-Code zu inspizieren.

In dieser Arbeit wurden verschiedene Sicherheitsfeatures von Android vorgestellt und einige Sicherheitslücken beschrieben. Die vorgestellten Punkte wurden bewertet und für einige sicherheitsrelevante Probleme konnten Lösungsansätze vorgestellt werden. Nach Abwägung der Vor- und Nachteile kann gesagt werden, dass Android ein ernst zu nehmender Kandidat auf dem Markt der Betriebssysteme mobiler Geräte sein wird. Durch den starken Rückhalt renommierter Unternehmen kann eine große Verbreitung erzielt und Android etabliert werden.

Trotz bestehender Sicherheitslücken sollte ein Einsatz von Android in Erwägung gezogen werden. Die schwerwiegenden Sicherheitslücken werden in zukünftigen Releases geschlossen werden. Um eine noch größere Sicherheit gewährleisten zu können, werden zukünftig auch auf Mobiltelefonen Virenscanner und Firewalls zum Einsatz kommen. Durch die Verbreitung als Open-Source ist ein kostengünstiger Einsatz möglich. Die große Community, die hinter Android steht, wird dessen Qualität sicherstellen.

4.2 Ausblick

Für den realen Einsatz von Android müssen noch einige Aspekte tiefgründiger untersucht werden. In dem Projekt „Pervasive Gaming“ von Prof. Dr. Olaf Zukunft, an dem ich teilnehmen werde, wird mit mobiler Software gearbeitet. Als Plattform wird dort möglicherweise Android zum Einsatz kommen. Dabei können die bereits gewonnenen Erkenntnisse mit einge-

bracht und neues Wissen erarbeitet werden. Dabei müssen ausgewählte Sicherheitsaspekte auf konkrete Anwendungsbereiche projiziert werden. Beispielsweise kann die Sicherheit von Spielen auf mobilen Geräten untersucht und getestet werden. Da Spiele einen großen Absatzmarkt haben und sich schnell verbreiten, sollte untersucht werden, wie Spiele konzipiert werden müssen, damit sie sicher sind.

In diesem Zusammenhang gibt es auch Berührungspunkte mit anderen Arbeiten aus dem Seminar. Zum Beispiel eignen sich die Erkenntnisse, die Thomas Preisler in seiner Ausarbeitung „Vorgehensmodelle für mobile Anwendungen“ erarbeitet hat, um die entwickelten Sicherheitskonzepte der Anwendungen testen zu können [vgl. Preisler 2008].

Die Ausarbeitung von Sicherheitsaspekten in mobilen Anwendungen erscheint mir auch im Hinblick auf meine Master-Thesis sehr interessant. Ich könnte mir vorstellen, dieses Thema in den kommenden Studienveranstaltungen weiter zu verfolgen und es möglicherweise auf mobile Agenten auszuweiten. Ein Beschränkung des Themas allein auf Android wäre in diesem Fall zu einseitig und nicht sehr interessant.

Literaturverzeichnis

Brodkin 0603

BRODKIN, Jon: Google: Android won't suffer from incompatibility. In: *Network World* (06.032008)

Google

GOOGLE: *Android - An Open Handset Alliance Project*. http://code.google.com/android/, Abruf: 19. Juli 2008. – Einstiegsseite der Google Android Homepage

Google Issue 191

GOOGLE (Hrsg.): *Issue 191: Support for encrypted SQLite databases*. http://code.google.com/p/android/issues/detail?id=191, Abruf: 20. Juli 2008

heise 2008

HEISE (Hrsg.): *Bericht: Verzögerungen bei der Entwicklung von Android-Handys*. Version: Juni 2008. http://www.heise.de/mobil/Bericht-Verzoegerungen-bei-der-Entwicklung-von-Android-Handys--/newsticker/meldung/109827, Abruf: 19. Juli 2008

Miner 2007

Vortrag von Rich Miner beim Computer Systems Colloquium der Stanford University. http://deimos3.apple.com/WebObjects/Core.woa/Browse/itunes.stanford.edu.1294764018.01488377969.1489155906?i=1822422778. Version: Herbst 2007

OHA a

OPEN HANDSET ALLIANCE (Hrsg.): *Android*. http://www.openhandsetalliance.com/android_overview.html, Abruf: 19. Juli 2008

OHA b

OPEN HANDSET ALLIANCE (Hrsg.): *FAQ*. http://www.openhandsetalliance.com/android_faq.html, Abruf: 19. Juli 2008

Preisler 2008

PREISLER, Thomas: *Vorgehensmodelle für mobile Anwendungen*. 2008