Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Fachstudie Nr. 201

Schutz privater Daten auf mobilen Geräten - geht das überhaupt?

Mark Aukschlat, Vanessa Jasny, Sebastian Pirk

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr.-Ing. habil. Bernhard Mitschang
Betreuer/in:	Christoph Stach, Dipl.-Inf.
Beginn am:	02. Juni 2014
Beendet am:	02. Dezember 2014
CR-Nummer:	K.6.5, D.4.6

Kurzfassung

Das Android Betriebssystem ist das einzige System, das dem Enduser zum großen Teil selbst überlässt wie er mit seinen Daten und Apps im Hiblick auf Privacy, Security und Trust umgeht. Somit liegt das Gros der Verabtwortung bei dem User selbst. Doch meist ist sich der User selbst nicht im Klaren wann und in wiefern seine Daten missbraucht werden können und wann die Sicherheit eines Android-Geräts gefährdet wird. Um den Problemen diesen Problemen zu begegnen, stehen dem Benutzer verschiedene Systeme zur Verfügung. In dieser Arbeit werden diese Systeme analysiert und kritisch bewertet. Es wird abschließend versucht eine Empfehlung zu geben, wie der Benutzer diese Systeme verwenden kann, um seine Daten zu schützen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

Verzeichnis der Algorithmen

1 Einleitung

1.1 Ausgangssituation

Smartphones sind heutzutage aus dem alltäglichen Leben kaum noch wegzudenken. Laut einer eMarketer[1] Untersuchung besitzen weltweit über anderthalb Milliarden Menschen ein Smartphone. Im zweiten Quartal 2014 wurden dabei über dreihundert Millionen Smartphones verkauft. Auf etwa fünfundachtzig Prozent dieser Geräte ist nach einer IDC Studie[2] ein Android Betriebssystem installiert.

Software von Drittanbietern, sogenannte Applikationen oder auch kurz *Apps*, sorgen dabei für eine große Vielseitigkeit der Smartphones. Im *Google Play Store*, dem offizielle App-Store von Google, lassen sich laut AppBrain.com über 1,3 Millionen Apps finden. Insgesamt gibt es über 30 App-Stores für Android Apps[3] und auch das Herunterladen von anderen Quellen ist möglich.

Die Vielfalt an Apps bringt jedoch nicht nur unzählige Möglichkeiten sondern auch Probleme mit sich. Oft sind sich Benutzer im Unklaren, über welche Rechte eine App verfügt und auf welche Daten sie Zugriff hat. Dadurch kann es zur ungewollten und unbemerkten Verwendung von privaten Daten kommen. Dieser Missbrauch ist dem Benutzer meist nicht bewusst oder er kann ihn nicht verhindern, will er die App weiter verwenden, [Bac12].

Die Bedrohungen, durch Apps, für den Benutzer lassen sich dabei in zwei grundsätzlich unterschiedliche Gefahrenquellen zusammenfassen. Zum einen müssen das System und seine Daten vor Manipulation oder ungewollten Zugriffen geschützt werden. Sicherheitslücken in Applikationen oder dem mobilen Betriebssystem werden von Angreifern gezielt ausgenutzt, um vom Nutzer unbemerkt Aktionen durchzuführen, die eigentlich nicht legitimiert sind und potenziell Schaden anrichten können. Hierbei spricht man von *Security*. Das andere Problem ist, dass dem Nutzer die Möglichkeit geboten werden muss reglementieren zu können, in welchem Umfang und auf welche Art der Daten eine App legal zugreifen darf. Funktionen dieser Art werden als *Privacy* zusammengefasst.

Das Vertrauen in eine App lässt sich allerdings nicht nur durch erhöhte Privacy- und Security-Maßnahmen steigern (siehe Abbildung 1.1), sondern auch durch andere Vorkehrungen.

[1] http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536
[2] http://www.idc.com/prodserv/smartphone-os-market-share.jsp
[3] Quelle: http://www.onepf.org/appstores/

Abbildung 1.1: Kohärenz zwischen Security, Privacy und Vertrauen.

1.2 Problemstellung

Android ist aktuell das am weit verbreitetsten OS für mobile Geräte wie oben bereits erwähnt. Während Apple alle Apps, die in den App Store[4] gestellt werden, ausführlichen Überprüfungen unterzieht, sind diese Mechanismen beim Play Store noch nicht so weit entwickelt. Außerdem können Anwendungen, die von anderen Quellen als dem Play Store bezogen werden, problemlos installiert werden, [SM13].

Aufgrund der hohen Beliebtheit von Android Geräten und den vorhandenen Lücken in Hinsicht auf Security und Privacy werden in dieser Arbeit Lösungen für das Android Betriebssystem betrachtet.

Ziel dieser Arbeit ist es bestehende Ansätze zur Steigerung von Privacy, Security und Trust für Android-basierte Geräte zu ermitteln und kritisch zu bewerten. In jedem der drei Bereiche wurden bereits zahlreiche Ansätze präsentiert. Manche Ansätze existieren nur als theoretische Ansätze und besitzen maximal einen Forschungs-Prototyp, andere gibt es (oft mit kostenpflichtigen Premiumfunktionen) im App-Store.

Zum einen wird in dieser Arbeit das theoretische Konzept dieser Arbeiten bewertet, was oft der einzige Weg ist, da es keine öffentlich zugänglichen Prototypen gibt. Zum anderen soll die praktisch Umsetzung analysiert werden. Diese ist gerade bei Security-Systemen die einzige Möglichkeit zur Analyse, da die großen Firmen keine Einblicke in ihre Arbeitsweise gewähren. Gerade die Benutzung spielt eine wichtige Rolle, da der Benutzer es verstehen und bedienen können muss.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Anforderungen und Definitionen: In diesem Kapitel werden die Begriffe Security, Privacy und Vertrauen noch einmal feiner definiert, wie sie in dieser Arbeit verstanden werden und die Anforderungen an die jeweiligen Systeme werden aufgezählt.

[4] https://itunes.apple.com/de/genre/ios/id36?mt=8

Kapitel 3 – Security-Systeme: In diesem Kapitel werden Security-Systeme betrachtet und kritisch bezüglich des Anforderungskataloges bewertet. Dabei werden zunächst wissenschaftliche Ansätze erläutert und anschließend Security-Apps aus dem App-Store ausführlich analysiert.

Kapitel 4 – Privacy-Systeme: In diesem Kapitel werden Privacy-Systeme betrachtet und kritisch bezüglich des Anforderungskataloges bewertet. Es findet eine analyse der wissenschaftlichen Ansätze statt, ergänzt durch zur Verfügung stehende Prototypen.

Kapitel 5 – Trust-Systeme: In diesem Kapitel werden Trust-Systeme betrachtet und kritisch bezüglich des Anforderungskataloges bewertet. Dabei standen weder Prototypen noch fertige Apps zum Test bereit, weshalb analysiert wird, ob manche Security und Privacy Systeme den Anforderungen entsprechend einen Teil der Aufgaben übernehmen können.

Kapitel 6 – Fazit: In diesem Kapitel wird ein finales Fazit zu Systemen aus allen drei Bereichen und geben einen Vorschlag, wie den Problemstellungen aus den Bereichen beigekommen werden kann.

Kapitel 7 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2 Anforderungen und Definitionen

Apps bedrohen die Datensicherheit (Security) und greifen auf unsere privaten Daten zu ohne dass wir das wollen oder kontrollieren können (Privacy). Um diesen Problemen entgegen zu kommen wurden Security- und Privacy-Systeme entwickelt. In diesem Kapitel wird zunächst erläutert, was in dieser Arbeit unter Security (Unterkapitel 2.1) und Privacy (Unterkapitel 2.2) verstanden wird und welche Anforderungen an diese Systeme gestellt werden. Außerdem wird in Unterkapitel 2.3 der Begriff des Trust in Hinsicht auf verfügbare Systeme diskutiert und schließlich werden zwei mögliche Definitionen gegeben.

2.1 Security

Security beschreibt den Schutz der Daten und des Systems. Bedroht werden diese zum einen durch Apps, zum anderen aber auch durch fremde Personen, die das Smartphone in die Hände bekommen. Ein Security-System muss Daten und System vor diesen Angriffen ausreichend schützen. Daraus ergibt sich folgende Definition:

Definition 2.1.1 (Security für Android-Systeme)
Bewahrung des Systems vor ungewollten Zugriffen durch fremde Entitäten.

Mittlerweile ist das Smartphone zu einem richtigen Allrounder geworden. Man kann Fotos machen, ins Internet gehen, sich zu einem Punkt navigieren lassen und Telefonieren. Durch dieses breite Spektrum an Möglichkeiten ergibt sich ein immer persönlicheres Gerät, das vor Unbefugten Zugriffen geschützt werden muss. Früher war es nötig eine ganze Palette an Gerätschaften in die Hände zu bekommen, was heute alles in einem einzigen Gerät steckt. Sollte es gelingen auf ein fremdes Gerät zuzugreifen so hat man heute sofort Unmengen an Daten zu verfügung. Von gespeicherten Urlaubsfotos, zu geheimen Geschäftsdaten oder privaten Nachrichten. Man könnte sich ohne weiteres als diese Person in Social Media Netzwerken ausgeben oder kostenpflichtige Nummern anrufen. Da sich immer mehr Menschen ohne Bedenken dieses riesige Spektrum benutzen und sich überhaupt nicht bewusst sind um die Gefahr eines so umfangreichen Systems, ist es auserordentlich wichtig die Daten bestmöglich zu schützen.

2.2 Privacy

Auf unseren Mobiltelefonen befinden sich heutzutage Unmengen an privaten Daten, wie z.B. Kontakte, Fotos, E-Mails etc. Häufig verlangen Apps Berechtigungen für diese Art Daten. Für den Endnutzer ist

dabei meist nicht ersichtlich wofür welche Daten verwendet werden und/oder ob diese überhaupt für die Funktion der App erforderlich sind. Und selbst wenn diese Berechtigungen erforderlich sind, nutzt die installierte App diese Daten nur um die einwandfreie Funktion zu gewährleisten oder werden Daten missbraucht (weitergeschickt, gespeichert, oder zu Werbezwecken an Dritte gesendet).

Es bestehen bereits mehrere Definitionen des Begriffs Privacy, zum Beispiel die Definition von Philip E. Agre und Marc Rotenberg [AR98], welche Privacy wie folgt definieren. „Privacy is the capacity to negotiate social relationships by controlling access to personal information. As laws, policies, and technological design increasingly structure peoples relationships with social institutions, individual privacy faces new threats and new opportunities."

Folgend führen wir eine eigene Definition für Privacy ein.

Es ergibt sich daraus als Definition:

Definition 2.2.1 (Privacy für Android-Systeme)
Daten, die als privat oder sensitiv eingestuft werden, sollen nur für dafür Vorgesehene Entitäten zugänglich sein.

In Folge dessen bilden sich einige Anforderungen an ein Privacy-System. Natürlich bestehen Grund-Anforderungen, wie eine einfache Installation, eine einfache/selbsterklärende Bedienung und natürlich sollte es auch möglich sein Berechtigungen einzuschränken. Die einfache Installation ist grundlegend für den Entschluss zur Installation durch den User. Der Benutzer möchte möglichst einfach an sein Produkt, welches in diesem Fall die Privacy-Anwendung ist, gelangen. Genauso verhält es sich mit der selbsterklärenden Bedienung. Die Bedienung sollte ohne vorheriges einlesen möglich sein. Funktionen und überwachte App beziehungsweise Daten sollten übersichtlich dargestellt werden. Die Möglichkeit Berechtigungen einzuschränken ist hierbei offensichtlich und zwingend erforderlich, da sie Hauptbestandteil einer Privacy-Anwendung ist.

Des weiteren sollte aus dem System erkennbar sein auf welche Daten eine überwachte App jeweils zugreift und wohin oder an wen diese Daten gesendet werden. Dies kann aufzeigen auf welche Daten wirklich zugegriffen wird und welche Daten nicht gebraucht werden. Aufgrund, der daraus erzielten Ergebnisse, können Berechtigungen angepasst werden.

Wenn ein Datenzugriff verboten wurde, sollte ein System dies auch in jedem Fall gewährleisten. Immerhin verlässt sich der User darauf, dass die Beschränkung, der Berechtigungen einzelner Apps, eingehalten wird. Sollte eine Beschränkung auf Berechtigungen einer App notwendig für die Funktion sein, sollte das System zumindest eine Fehlermeldung ausgeben. Anderenfalls kann der Grund einer Fehlfunktion nicht erkannt werden. Die App sollte somit keineswegs falsche Ergebnisse liefern, aufgrund zu hoher und/oder „falscher" Einschränkungen.

Zudem sollte die Laufzeit einer App nicht erheblich beeinträchtigt werden durch die Security-Anwendung. Auch App-Updates sollten weiterhin möglich sein. Dies ist besonders wichtig, da in Updates oft Fehlerbehebungen stattgefunden haben. Bei möglichen Updates einer App sollten Berechtigungen angepasst werden können. Wenn von den Testsystemen selbst Berechtigungen erfordert werden, sollte auch erklärt werden wozu diese benötigt werden.

2.3 Trust

Trust ist ein soziales Phänomen, welches wir auf Applikationen anwenden wollen Gambetta et al. [Gam88] definieren Trust als „eine besondere Ebene der subjektiven Wahrscheinlichkeit mit welcher ein Agent annimmt, dass ein anderer Agent oder eine Gruppe von Agenten eine bestimmte Aktion ausführt [...]." In diesem Fall übernimmt die Rolle des andere Agent eine Applikation, die wir installieren wollen oder bereits installiert haben.

Die Aktionen, welche eine App dabei durchführen kann, können dabei unterschiedlich sein. Yan et al. [YZD12] definieren beispielsweise den Trust in Applikation als „der Glaube des Benutzers an eine Applikation eine Aufgabe wie erwartet zu erfüllen." Sie beziehen sich damit auf die Funktionalität der App und wie hoch die Erwartung ist, dass die App die gewünschten Funktionen bietet.

Vertrauen in eine App endet jedoch nicht bei ihrer Funktionalität. Jøsang [Jøs96] definiert Trust als „der Glaube dass sie [die Entität in die man Vertrauen hat] in ihrem Verhalten keine bösartige Absichten hat." Die Möglichkeiten, auf die eine App Schaden anrichten kann, wurde bereits in Hinsicht auf Security und Privacy dargelegt. Trust in eine App sollte also auch das Gefahrenpotenzial, welches eine App bietet, beinhaltet.

Aus diesem Verständnis für Trust lässt sich folgende eigene Definition geben:

Definition 2.3.1 (Trust in eine Applikation)
Der Glaube des Benutzer an eine mobile Applikation, die versprochenen Funktionalitäten bietet, private Daten nicht gegen den Wunsch des Benutzers weiterzugeben und keinen Schaden anzurichten.

Aus dieser Definition stellen sich für ein Trust-System für Apps einige Anforderungen. Das System sollte einen Wert liefern, der für den Benutzer eine verständliche Aussage trifft, wie weit eine App den Erwartungen bezüglich der drei Kriterien (Funktionalität; Schadenspotenzial; Zugriff auf private Daten) entspricht. Die Entscheidung ob die App genug Trust besitzt, dass ich sie auf meinem Smartphone installieren will, sollte dabei mit Hilfe des Wertes zu treffen sein.

Es wäre bei einem Trust-System des weiteren von Vorteil, wenn dem Benutzer Einsicht geliefert wird, aus welchen Gründen der Trust-Wert für eine App nun gut oder schlecht ist. Eine Gewichtung der einzelnen Kriterien nach dem, wie wichtig sie dem Besitzer sind (manch einer legt keinen Wert auf seine Privatsphäre) sollte möglich sein. Außerdem wäre eine leichte Vergleichsmöglichkeit mit ähnlichen Apps bezüglich des Trust, den man in sie hat von Vorteil, da man sich bei einem oft breiten Angebot an Apps für eine Aufgabe besser für eine der Apps entscheiden könnte.

Nachdem die Begriffe Security, Privacy und Trust ausreichend definiert wurden und Anforderungen an die jeweiligen Systeme gestellt wurden, werden in den folgenden Kapiteln nun die Systeme untersucht und nach den Anforderungen bewertet. Begonnen wird dabei mit Security-Systemen in Kapitel 3.

3 Security-Systeme

Zunächst werden in Unterkapitel 3.1 die Mechanismen betrachtet, welche das Android System bereits bietet um die Security zu gewährleisten und warum diese nicht ausreichend sind. Anschließend werden in Unterkapitel 3.2 Lösungsansätze analysiert, die sich mit Security für Android befassen, und auf ihre Relevanz und Tauglichkeit überprüft. Abschließend wird in Unterkapitel 3.3 ein Fazit zum aktuellen Stand der Forschung und Technik auf diesem Gebiet gezogen. Diese Grundgliederung gilt analog für die folgenden beiden Kapitel.

3.1 Ansätze von Android

Android selbst bietet durch seinen Aufbau schon einige Sicherheitsfeatures. So kann man bei jeder App, die man installieren möchte, zuvor sehen welche Berechtigungen diese haben möchte. Außerdem ist Android so aufgebaut, das es für jede Anwendung eine eindeutige Benutzer-ID (UID) anlegt, es in einem eigenen Prozess laufen lässt und ihr einen eigenen Adressraum zuweist. Allerdings können Apps via des sogenannten IPC kommunizieren um Daten auszutauschen. So können auch harmlose Apps durch bösartige manipuliert werden.

Des weiteren nützen diese Schutzmaßnahmen nichts gegen Physische Angriffe. Zwar gibt es dafür in den bisherigen Android Versionen auch einige Ansätze, zum Beispiel einen Sperrcode um das Handy zu sperren, doch sobald es einmal eingegeben ist, hat man volle Zugriffsrechte. So kann man nicht einzelne Apps sperren oder den Zugriff auf bestimmte Dienste einschränken. Außerdem nützt das bei einem Diebstahl sehr wenig, der neue Benutzer kann das Gerät mit wenigen Klicks zurücksetzen und hat wieder die vollen Zugriffsrechte.

Am 12. November 2014 ist ein neues Android System, Android L, auf den Markt gekommen. Da es auf den meisten Geräten noch nicht erschienen ist, wurde es noch nicht mit in die Einleitung einbezogen. Natürlich geschieht das nun mit der Zeit. Mit diesem neuen System hat Google sehr viele neue Schutzmechanismen herausgebracht, um die Security des Android Gerätes zu erhöhen. Die wichtigsten neuen Features werden im Folgenden vorgestellt:

Bislang war es so das bestimmte Sicherheitslücken nur über ein komplettes Android Update vollzogen wurden. Das beherbergte allerdings einige Probleme, da (abgesehen von den Nexus Geräten) die Hersteller selbst für die Systemupdates verantwortlich waren. Da viele Hersteller einige eigene UIs entwickelt haben und diese immer angepasst werden müssen, dauert es oft Monate bis ein Android Gerät wieder auf dem aktuellsten Stand ist. Außerdem liefern viele Hersteller keinen langen Support, so das man bereits nach einer kurzen Zeitspanne nach dem Kauf eines Gerätes, nicht mehr die Möglichkeit besitzt das neueste Update zu bekommen. Dem will Google nun entgegenwirken indem es die Sicherheitsupdates separat in den Google Play-Store stellt. So könnten auch ältere Handys

immer auf dem neuesten Stand bleiben, wen es um die Sicherheit geht. Außerdem hat Google gesagt das sie diese Patchs spätestens innerhalb von 6 Wochen einspielen wollen, was wesentlich schneller wäre als auf ein Systemupdate des Herstellers zu warten.

Eine weitere neue Funktion von Android L ist die Kill Switch. Bislang war es möglich sein Android Gerät ohne jegliche Passworteingabe einfach per Knopfdruck zurück zu setzen. Das war für Diebe natürlich sehr nützlich, sie klauten einfach ein Handy, setzten es zurück und verkauften es weiter. Sie hatten meist überhaupt kein Interesse an den Daten die davor auf dem Handy waren. Nun soll dies erschwert werden, indem man sich zuerst in das Google Konto des alten Benutzers einloggen muss, bevor man das Gerät wieder auf die Werkseinstellungen zurück bekommt. Bei Apple hat so eine Funktion bereits vor einiger Zeit eingeführt und konnte damit einen drastischen Rückgang der iPhone Diebstähle verzeichnen. So soll die Stadt New York laut dem IT-Portal „The Register“ einen Rückgang von 29 Prozent der Iphone Diebstähle im Gegensatz zum Vorjahr verzeichnet haben können. In San-Francisco wurden 38 Prozent weniger gestohlen iPhones beobachtet, in London fast ein Viertel, [kil].

3.2 Lösungsansätze

Eine Art die Schwachstellen zu schließen sind sogenannte Antivirenprogramme. Sie untersuchen das Gerät nach bekannten Signaturen und entfernen die entsprechende App, sollte eine bekannte bösartige Signatur gefunden werden. So kann bösartige Software schnell und effektiv wieder entfernt werden.

Es gibt mittlerweile eine riesige Auswahl an Antivirenprogrammen, die sich in Ihrem Aufbau und vor allem Ihren Erweiterungen sehr stark Unterscheiden. Viele der Antivirentools kosten etwas, doch es gibt auch eine sehr große Auswahl von Suiten die komplett Kostenlos verfügbar und benutzbar sind.

So zeigt ein Testbericht von Chip.de vom 18.11.2014 dass die meisten Suiten sehr gut ausgestattet sind. Siehe dazu auch Abbildung 3.1.

Insgesamt wurden 25 Android Virenscanner getestet. Es wurde darauf geachtet wie viel der vorgesetzten Malware erkannt wird. Hierbei lag die Höchstpunktzahl für Schutz und Bedienung bei 6,0 Punkten, dazu gab es dann noch einen Extrapunkt für die Features. Man sieht sofort das ein Großteil der Suiten sehr gut abschneiden. Von diesen sind auch einige Gratis Suiten dabei, zum Beispiel Qihoo oder Mobile Clean Master. Außerdem heißen kleine Punktabzüge keinesfalls das es sich hierbei um eine schlechte Suit handelt.Da der Durchschnitt der erkannten Malware bei sehr guten 98 Prozent lag, gab es bereits für kleine Patzer einen Punktabzug. So hat sogar der letzte Platz immerhin eine Erkennungsrate von 91,6 Prozent. Da es immer ein Rennen gibt zwischen dem Erscheinen einer neuen Malware und dem Erkennen dieser, kann es beim nächsten Test wieder völlig anders aussehen.

Viele der getesteten Suiten bieten außerdem auch Schutzmechanismen gegen physische Angriffe. So bietet Avast eine große Auswahl an Lösungen an.

Erweiterungen:

ALLE TESTERGEBNISSE AUF EINEN BLICK

Platz	Virenscanner	Schutz	Bedienung	Features	Gesamt	Download
1	Bitdefender Mobile Security 2.23	6,0	6,0	1,0	13,0	Download
1	BullGuard Mobile Security 14	6,0	6,0	1,0	13,0	Download
1	ESET Mobile Security & Antivirus 3.0	6,0	6,0	1,0	13,0	Download
1	Cheetah Mobile Clean Master 5.8	6,0	6,0	1,0	13,0	Download
1	Cheetah Mobile CM Security 1.8	6,0	6,0	1,0	13,0	Download
1	Qihoo 360 MobileSecurity 1.0	6,0	6,0	1,0	13,0	Download
1	Sophos Mobile Security 3.5	6,0	6,0	1,0	13,0	Download
1	Trend Micro Mobile Security 5.0	6,0	6,0	1,0	13,0	Download
1	Trustlook Antivirus 2.2	6,0	6,0	1,0	13,0	Download
10	Avira Free Android Security 3.5	5,5	6,0	1,0	12,5	Download
10	DU Apps Studio DU Speed Booster 2.0	5,5	6,0	1,0	12,5	Download
10	Kaspersky Internet Security 11.5	5,5	6,0	1,0	12,5	Download
10	McAfee Mobile Security 4.2	5,5	6,0	1,0	12,5	Download
10	TrustGo Mobile Security 1.4	5,5	6,0	1,0	12,5	Download
15	Antiy AVL 2.3	6,0	6,0	0,0	12,0	Download
15	avast! Mobile Security 3.0	5,0	6,0	1,0	12,0	Download
15	Quick Heal Total Security 2.00	5,5	5,5	1,0	12,0	Download
18	Norton Mobile Security 3.8	6,0	4,0	1,0	11,0	Download
19	F-Secure Mobile Security 9.2	3,0	6,0	1,0	10,0	Download
19	Ikarus Mobile Security 1.7	4,0	5,0	1,0	10,0	Download
19	Webroot SecureAnywhere Mobile 3.6	3,0	6,0	1,0	10,0	Download
22	AVG AntiVirus FREE 4.1	5,0	3,0	1,0	9,0	Download
22	Bornaria Mobile Security 1.5	2,5	5,5	1,0	9,0	Download
22	Comodo Mobile Security 2.4	3,0	5,0	1,0	9,0	Download
25	G Data Internet Security 25.6	3,0	3,0	1,0	7,0	Download

Abbildung 3.1: Übersicht über die Ergebnisse des Chip.de Antivirentests vom 18.11.2014.

Abbildung 3.2: Hier sieht man links das Handy im Knox-Modus. Jede App hat ein kleines Schloß Symbol, die Statusleiste besitzt Streifen und es gibt nun den Samsung Knox Store. Rechts ist der herkömmliche Startbildschirm.

SMS- & Call-Filter: Damit ist es möglich spezielle Rufnummern zu sperren, sowohl für Anrufe, als auch für SMS. Außerdem ist es möglich den kompletten Telefonverkehr in vorgegebenen Zeiten lahmzulegen. So kann man verhindern das Fremde unerlaubt die Telefonfunktion benutzen oder SMS versenden. Außerdem gibt es sogenannte Dialer Malware, mit dieser bösartige Apps versuchen kostenpflichtige Nummern zu wählen. So könnte man mit einer gezielten Sperre verhindern das so etwas weiterhin passiert.

Firewall: Mit der in Avast eingebauten Firewall ist es möglich einzelnen Apps den Internet Zugriff zu verweigern.

Diebstahlschutz: In Avast ist ein Schutz integriert, um einen höheren Schutz gegen Diebstahl zu haben. So kann man sein Gerät bei einem Diebstahl orten, das Gerät komplett sperren, eine Sirene ertönen lassen, ihm Nachrichten anzeigen lassen usw.

Apps sperren: Manche Suiten bieten auch einen Schutz gegen das unbefugte Öffnen von Dateien an. So kann man einzelnen Apps einen eigenen Sperrcode zuordnen, ohne den man diese App nicht öffnen kann. Dieser kann auch unabhängig vom Bildschirmsperrcode eingerichtet werden, so das man den einen Code zum Entsperren des Gerätes hat und den anderen zum entsperren einer privaten App.

Während wir bisher Sicherheitslösungen für den normalen Endanwender angeschaut haben, spielen für Firmen ganz andere Sicherheitsaspekte ein Rolle. So hat man in vielen Firmen die Möglichkeit sein eigenes Gerät für die Arbeit einzusetzen. Mit Samsung Knox sollen Kunden mit einem Klick zwischen Privat und Geschäftsgerät wechseln können. Es stellt eine separate Umgebung für Privates und Geschäftliches bereit. So soll bewerkstelligt werden das ein Benutzer nicht versehentlich sensible Daten an dritte Weitergibt und das dass Unternehmen auch keine Zugriffe auf private Daten bekommt. Samsung Knox basiert auf dem von der NSA entwickelten Security Enhanced Android (SE Android) und ermöglicht quasi, zwei Android-Systeme auf einem Gerät laufen zu lassen. Dabei verankert es sich sowohl in Soft- als auch in Hardware Ebene, eine Verschlüsselung auf Dateisystem-Ebene ist

ebenfalls Bestandteil. Systemadministratoren können den Business-Container dabei nach den MDM-Richtlinien des Unternehmens für ein bestimmtes Mobilgerät konfigurieren. Der Mitarbeiter kann sein Business-Container dann aufrufen indem er auf der Startseite des Smartphones die Knox-App startet und sein Passwort eingibt. Nun befindet er sich im Business-Container, von Samsung, auch Knox-Modus genannt. Er erkennt diesen Modus daran das sich am oberen Bildschirmrand streifen befinden und jedes Symbol ein kleines Schloß an der Ecke besitzt. Außerdem gibt es zum Beispiel keinen Play-Store, sondern einen extra Samsung Knox Store, in dem es eine extra Auswahl mit geschützten Apps gibt, [kno14].

Außerdem kann der Systemadministrator den Knox-Modus sperren und der Benutzer kann ihn so lange nicht mehr benutzen bis der Systemadministrator ihn wieder entsperrt. Weiterhin kann der Systemadministrator auch ein Konto löschen, dann werden alle Daten aus dem Knox-Modus gelöscht, die privaten Daten bleiben jedoch erhalten.

Virtual Fort Knox bietet nun ein Konzept um eine sichere Cloudbasierte Kommunikation zu gewährleisten. Bereits bei der Konzipierung eines Systems setzt Virtual Fort Knox einige Standards die eingehalten werden sollen. So wird durch das VFK-Sicherheitskonzept zwischen verschiedenen Sicherheitsbereichen differenziert. Das fängt an beim Schutz vor Störungen aufgrund von Elementarereignissen(Überschwemmung, Erdbeben, Bränden, Stromausfall) und geht bis zur Datensicherheit, bei der es um den Schutz vor nicht berechtigten Zugriff und Manipulation (Zugriffsrechte auf Daten, Kapselung und Verschlüsselung abgespeicherter Daten) geht. Aus Softwaresicht verfolgt Virtual Fort Knox das Gesamtkonzept einer servicebasierten Architektur, die sowohl Services als auch aggregierte Services für das Produktionsumfeld bietet. Ein Service wird hierbei definiert als eine Einheit mit einer konkreten Funktion und eindeutigen Ein- und Ausgangsparametern. Es gibt Anwender, die Anwendungen über die Plattform erwerben können. Außerdem gibt Befähiger(Softwarehersteller) die als Serviceprovider fungieren. Außerdem können sie aggregierte Services erzeugen, indem Services von sowohl denselben als auch unterschiedlichen Softwareherstellern kombiniert werden können, [HWSB13]

3.3 Fazit

Es gibt sehr viele Antiviren Suiten, die auch sehr gute Arbeit leisten. Allerdings können diese natürlich keine 100 % Sicherheit gewährleisten. Bei sogenannter Zero Day Malware, also einer völlig neuen Schadsoftware, ist es für die derzeitigen Antiviren Suiten unmöglich diese aufzuspüren. Ist diese Signatur noch nicht erfasst, finden sie die bösartige Software nicht, da die Antiviren Suiten keine Erkennungsmethoden haben um anhand des Verhaltens einer App zu erkennen ob sie Schaden anrichten will. Als Fazit nehme ich mit, das eine Antiviren Suite sehr nützlich sein kann, vor allem wen man Apps von Drittanbietern installiert, da sich diese überhaupt keiner Kontrolle unterziehen müssen. Außerdem sind vor allem die Erweiterungen sehr interessant, da sie oft einen erweiterten Diebstahlschutz bieten, den es so von Google nicht gibt.

Auch das neue Android Betriebssystem steuert einiges zur Security bei. Lücken im Betriebssystem, die durch Malware missbraucht werden könnten, können schnell geschlossen werden.Außerdem ist Kill Switch eine Neuerung, die man sich nicht entgehen lassen sollte. Wenn die Updates bald angeboten werden, sollte man diese schnellstmöglich installieren.

Samsung bietet einen sehr interessanten Denkansatz um die Benutzung von Android Geräten im Business-Alltag voranzubringen. Dafür muss nämlich ein gewisser Sicherheitsstandard her. Außerdem hat der Nutzer große Vorteile wenn er das Gerät gleichzeitig privat nutzen kann. So ist er nicht gezwungen sich mehrere Geräte anschaffen zu müssen. Allerdings gibt es auch einen Haken. Da Samsung Knox auf das von der NSA entwickelte SE Android basiert gibt es auch viele negative Verschwörungstheorien. Zwar sagt die NSA selbst das sie sich keine Hintertürchen eingebaut hat um Userdaten auszuspähen, doch kann man sich nicht sicher sein ob eins vorhanden ist, [Due13].

4 Privacy-Systeme

Die Gliederung des Kapitels Privacy verhält sich analog zur Gliederung des Security-Kapitels.

4.1 Ansätze von Android

Android 4.3 besitzt ein Sicherheitsfeature „AppOps“, welches Apps Berechtigungen entziehen kann. Hier sind vier Untergruppen von Berechtigungen aufgelistet. Diese lauten Standort, Persönlich, Gerät und SMS/MMS. In diesen vier Berechtigungsgruppen sind dann die Apps gelistet, welche die jeweilige Berechtigung angefordert haben. Unter der Funktion „App Vorgänge“ kann geprüft werden welche zugelassenen Berechtigungen wirklich genutzt wurden und diese auch „an“ oder „ausschalten“. Noch ist App Ops nicht gänzlich fertiggestellt und noch nicht freigeschaltet, um davon profitieren zu können muss ein „Permission Manager“ installiert werden.

Das neue Android 5 Lollipop hat bereits Privacy Features, wie zum Beispiel den „Guest Mode“. Hier kann der Nutzer ein Gastprofil erstellen, welches keinen Zugriff auf Daten des persönlichen Profils hat. In diesem Gastprofil können auch Email Adressen etc. vom Google-Konto, eines Freundes oder Apps vom App-Store heruntergeladen werden. Auch weitere Profile können erstellt werden, wodurch zum Beispiel private und berufliche Daten getrennt werden.

Somit bietet Android bereits gute Ansätze zur Privacy Gewährleistung, doch noch keine ausreichende. Es stellt sich die Frage ob andere Ansätze zur Gewährleistung von Privacy tauglich sind und in wie fern sie halten was sie versprechen.

4.2 Lösungsansätze

Eine Vielzahl von Entwicklern hat sich mit der Sicherung von Privacy beschäftigt und sind dabei zu Ergebnissen wie zum Beispiel AppFence, Dr Android & Mr Hide, Privacy Blocker, MOSES, SRT-AppGuard gekommen.

In dieser Arbeit befassen wir uns Hauptsächlich mit den drei Systemen Dr. Android & Mr. Hide [JMV+12], MOSES [RCCF12] und SRT-AppGuard [BGH+12] [BGH+13].

Alle drei Testsysteme haben den gemeinsamen Schwerpunkt Privacy und Security, welches hier schwer abzugrenzen ist. Die Systeme SRT-AppGuard und Dr. Android & Mr. Hide arbeiten beide mit Hilfe von Quellcode-Modifizierung, was folglich nochmals beschrieben wird.

Dr. Android & Mr. Hide

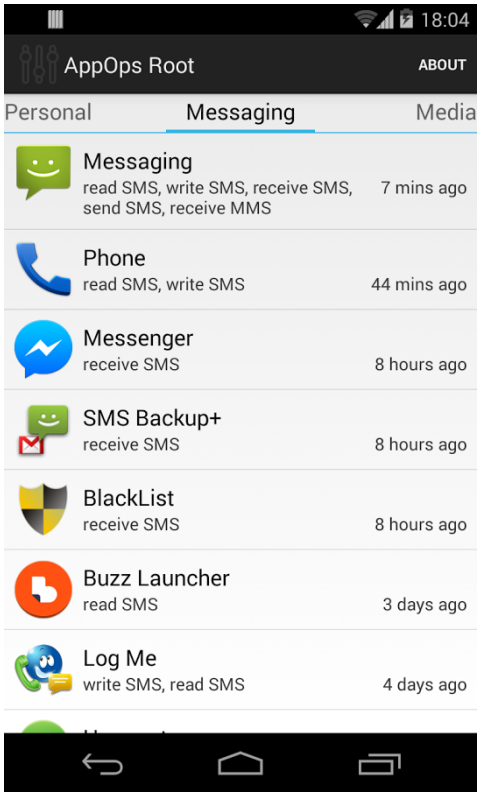

Abbildung 4.1: Sreenshot des Katalogs SMS/MMS von AppOps[a]

[a]Quelle: https://play.google.com/store/apps/details?id=droidmate.appopsinstaller&hl=de

Dr. Android ist angelehnt an die gegenwärtige Berechtigungsanfrage von Android im App-Store. Es werden so genannte Untergruppen von Berechtigungen erstellt. Dabei wurden die fünf größten Berechtigungen ausgewählt und verfeinert. Zu den ausgewählten Berechtigungen gehören Internetzugriff, Kontakte, GPS, Telefondienst und Systemeinstellungen.

Dr. Android steht für Dalwik Rewriter für Android und arbeitet in drei Schritten. Zuerst wird der Dalvik Code der zu überwachenden App modifiziert, um Mr. Hide verwenden zu können und hidelibcode in die App einzubinden. Die Standard Android Berechtigungen werden von Dr. Android entfernt und stattdessen die hidelib Berechtigungen eingefügt. Zuletzt werden noch Quellenverzeichnisse, welche Layout-Schnittstellen definieren, modifiziert.

MOSES

Moses arbeitet mit unterschiedlichen Profilen. Zu Beginn gibt es ein „Security-Profile“, das sich „Default“ nennt. Dieses kann man weder editieren noch löschen, da alle neu installierten Apps zunächst diesem „Default“ Profil zugewiesen werden, wenn nichts anderes definiert wurde.

Weitere Profile können beliebig erstellt werden und auch benannt bzw. umbenannt werden. In jedem Security-Profil werden vom User nun die Berechtigungen eingeschränkt. Apps und Daten können den

verschiedenen zuvor erstellten Profilen zugeordnet werden. Bei eigens erstellten Profilen ist einem selbst überlassen wie man diese editiert und verwaltet. Jedoch gibt es auch die Möglichkeit, Profile von einem Administrator erstellen zu lassen. Bei von einem Administrator erstelltem Profil, kann der Zugriff von diesem auf ihn selbst beschränkt werden.

Somit kann beispielsweise ein Arbeitgeber ein Profil namens „Arbeit“ erstellen und Job-relevante Daten und/oder Apps auf dieses Profil beschränken. Zudem gibt es die Möglichkeit den Zugriff auf bestimmte Zeiten sowie Orte zu beschränken.

SRT-AppGuard

Im Grunde genommen lässt sich die Arbeitsweise in drei Schritte teilen:

1. Scannen und umschreiben
2. Deinstallieren
3. Installieren der modifizierten App

Wobei sich die Technik auf das „Inline reference monitoring(IRM)“ bezieht. SRT-AppGuard macht sich die Tatsache zu Nutze, dass vom App-Store heruntergeladene Apps als .apk-Datei im Dateiordner (der für jeden lesbar ist) gespeichert werden. SRT-AppGuard ist im Stande diese App-Packete zu lesen und sie gegebenenfalls auch umzuschreiben. Bei der Installation einer neuen App vom App-Store scannt SRT-AppGuard zunächst die .apk-Datei. Erforderliche Berechtigungen werden angezeigt, sowie ein zusätzlicherRiskscore. Der Riskscore gibt ein angebliches Risiko an, das möglicherweise von der App ausgeht.

Nun können die Berechtigungen eingeschränkt werden, wobei sich der Riskscore dementsprechend ändert. Ist die Auswahl getroffen, schreibt SRT-AppGuard die besagte Datei um. Wenn der Nutzer nun, die von SRT-AppGuard umgeschriebene App, installieren möchte, muss er zunächst zustimme, die zuvor heruntergeladene Originalversion, deinstallieren zu lassen. Diese Deinstallation wird von SRT-AppGuard durchgeführt. Anschließend wird die modifizierte App installiert.

Die .apk-Datei der „Original-App“ bleibt bestehen und es kommt eine weitere .apk-Datei der modifizierten Version hinzu. Die modifizierte App wird neu gepackt und von SRT-AppGuard mit einem neu generierten Schlüssel versehen.

SRT-AppGuard und Dr. Android & Mr. Hide unterscheiden sich in der Arbeitsweise stark von MOSES, da der Quellcode einer App neu geschrieben wird MOSES so nicht der Fall ist.

Diese Arbeitsweise ist jedoch mit Vorsicht zu genießen. Da die modifizierten Apps neu gepackt und mit einer neuen Signatur versehen werden, wird hier in das Urheberrecht des Entwicklers eingegriffen.

Kommen wir nun zu den verschiedenen Anwendungsgebieten der drei hier betrachteten Systeme. Dr Android & Mr Hide ist für den privaten Gebrauch aber auch für Entwickler geeignet (Da hier erkennbar wird in wie weit Berechtigungen erforderlich sind und/oder wie diese eingeschränkt werden könnten).

Moses kommt überall da wo Daten, Kontakte und Apps in Bereiche geteilt werden zum Einsatz. Vorallem für Berufstätige, welche berufliche und private Daten, Kontakte sowie Apps voneinander

trennen wollen. Sowie für Arbeitgeber, welche Profile für Arbeitnehmer erstellen können, um jobrelevante Daten zu verwalten. Oder auch für Studierende, um studiumrelevante Daten von privaten zu trennen.

SRT-AppGuard ist ausschließlich für den privaten Gebrauch gedacht, was auch in den Nutzungsbedingungen vermerkt ist.

Bei Betrachtung der Anwendungsgebiete der gewählten Systeme ist festzustellen, dass sie größtenteils an verschiedene Personengruppen adressiert sind. Dr Android & Mr. Hide ist anders als die anderen Testsysteme nicht nur ausgelegt darauf, die Daten der User zu schützen, sondern auch auf Entwickler, damit diese feinere Berechtigungen anfordern könnten. Doch es bestehen noch weitere Merkmale, in welchen sich die Systeme unterscheiden.

Moses unterscheidet sich in der Arbeitsweise, dadurch dass es mit Profilen arbeitet und Kontakte, Apps und Berechtigungen in Profile einteilt. Hier wird nicht auf den Quellcode der Apps zugegriffen, wie bei SRT-AppGuard und Dr. Android & Mr. Hide. Zudem besitzt Moses eine Extrafunktion, welche es ermöglicht anhand der Profile den Zugriff von Apps, Kontakten und/oder Daten auf festgelegte Zeiten oder Orte zu beschränken. Dies ist dann von Vorteil, wenn sich zum Beispiel streng vertrauliche Daten auf dem Gerät befinden. Es kann in so einem Fall die Einstellung gewählt werden, dass erst bei Betreten des Büros das Profil „work“ aktiviert wird und somit auch der Datenzugriff dieses Profils. Im Zuge dessen besteht auch die Möglichkeit, dass ein Administrator Profile erstellen kann, welche geschützt sind und nur durch ihn selbst modifiziert werden können.

Weiterblickend fällt auf, dass im Gegensatz zum SRT-AppGuard, bei Dr. Android & Mr. Hide eine feinere Berechtigungseinschränkung möglich ist. So kann beim SRT-AppGuard Zugriff aufs Internet verweigert werden, jedoch nicht wie bei Dr. Android & Mr. Hide auf eine bestimmte Domäne beschränkt werden.

Was als weiteres Feature, nur diesmal bei SRT-Appguard, gesehen werden kann, ist, dass Werbung verborgen wird (was umstritten ist, da sich Apps größtenteils so finanzieren) zudem zeigt das System zusätzlich einen Riskscore an. Dieser Riskscore soll Auskunft über mögliche Bedrohungen, die von der App ausgehen, zeigen.

Hinblickend auf unsere Anforderungen an ein Privacy System, können einige Schlüsse gezogen werden.

Betrachten wir die Anforderung einer einfachen Installation, so können wir feststellen, dass keine der drei Systeme unmittelbar im App-Store zu downloaden ist, was die einfachste Art der Installation darstellen würde. Zwar war SRT-AppGuard bereits im Play-Store vorhanden, wurde aber aufgrund von möglichen Urheberrechtsverletzungen entfernt. Trotzdem liegt die einfachste Installation im Vergleich weiterhin bei SRT-AppGuard. Hier ist ein Benutzerhandbuch vorhanden, das auch für den Laien, in wenigen Schritten, gut erklärt wie die Installation abläuft.

Bei Dr Android & Mr Hide läuft eine Installation über `https://github.com/plum-umd/redexer`. Hier ist Fachwissen erforderlich, sowie die Installation von vorrausgesetzten Tools. Bei Moses ist lediglich ein Quellcode vorhanden. Das Implementieren wird somit dem User überlassen.

Die einfache selbsterklärende Bedienung ist bei SRT-AppGuard gegeben. Die Anwendung lässt sich intuitiv bedienen. Alle installierten Apps werden in einer Liste angezeigt, mit zusätzlichem Riskscore.

Aus der Liste kann man dann beliebige Apps auswählen. Bei der Wahl einer App muss man dann zunächst der Deinstallation zustimmen und anschließend der neuen Installation. Wenn dies getan ist, werden alle Berechtigungen der App angezeigt, aus denen man dann jeweils mit einem „haken“ Berechtigungen entfernen kann. Zudem steht bei der jeweiligen Berechtigung rechts ein kleines Symbol mit einem „i“ oder einem Ausrufezeichen. Klickt man dieses Symbol an, kommen weitere Hinweise, in wie weit die App, mit Hilfe von dieser Berechtigung zum Beispiel schaden anrichten/ auf private Daten zugreifen könnte. Sind diese Berechtigungen geändert, gelten sie ohne explizites speichern. Apps, deren Berechtigungen geändert wurden, erscheinen in einer gesonderten Liste, was mehr Überblick verschafft. Im Nachhinein können Berechtigungen auch einfach wieder geändert werden, indem wieder die App in der Liste angeklickt wird, und Haken neu gesetzt oder doch entfernt werden.

Da es sich bei MOSES und Dr. Android & Mr. Hide um Prototypen handelt, werden hier keine Schlüsse zur Bedienung gezogen.

Die Möglichkeit Berechtigungen einzuschränken ist auf jeden Fall bei allen drei Systemen gegeben. Bei Dr. Android & Mr. Hide können Berechtigungen feiner beschränkt werden, zum Beispiel beim Internet auf bestimmte Domänen oder bei Kontakten auf Email-Adressen. Bei MOSES können Berechtigungen auf bestimmte Profile eingeschränkt werden. Bei SRT-AppGuard können Berechtigungen theoretisch sogar gänzlich eingeschränkt werden, jedoch nicht bei bereits vorinstallierten Apps, da diese nicht als .apk-Datei im Dateiordner vorhanden sind (somit ist kein Zugriff auf den Quellcode möglich).

Bei SRT-AppGuard ist aus dem System erkennbar auf welche Daten die App jeweils zugreift. Nachdem eine App als „überwacht“ markiert wurde (indem die Originalversion deinstalliert und eine modifizierte Version installiert wurde) kann man durch klicken auf die App in der Liste überwachter Apps auf den Punkt „Log“ gelangen. Hier werden tabellarisch die Zugriffe der App aufgezeigt. Mit jeweiligem Datum und der Uhrzeit sind dann gewährte Zugriffe gelistet und auch verweigerte Zugriffe.

In der Tabelle „Log“ finden wir teilweise Informationen dazu wohin welche Daten gesendet werden. Hier ist zwar aufgelistet wann Internetzugriffe auf welche Seiten stattfinden, wodurch man auch auf Informationsübermittlung schließen können, jedoch ist dies nicht explizit angegeben.

Es wird keine Fehlermeldung von SRT-AppGuard ausgegeben, wenn zu hohe Einschränkungen in den Berechtigungen vorgenommen wurden. Berechtigungen können beliebig eingeschränkt werden. Erst bei der Verwendung der jeweiligen App (deren Berechtigungen geändert wurden durch SRT-AppGuard) wird die Fehlfunktion ersichtlich. Dass diese Fehlfunktion auf SRT-AppGuard zurückzuführen ist, kann man nicht erkennen.

Wenn zum Beispiel bei der „VVS“ App alle zu einschränken möglichen Berechtigungen tatsächlich eingeschränkt werden, lässt SRT-AppGuard dies ohne Probleme zu. Wenn wir nun die App ausführen möchten, wird uns eine Fehlermeldung der App selbst angezeigt. Siehe dazu Abbildung 4.2.

Ein weiteres Beispiel: „Avast mobile Security“ wird von SRT-AppGuard als sehr bedrohend eingestuft. Mit einem Riskscore von 9.3, wir eine Überwachung hier stark empfohlen. Wenn man nun prüfen möchte, welche Berechtigungen durch SRT-AppGuard eingeschränkt werden könnten, müssen wir zunächst zustimmen, die Originalversion zu deinstallieren. Anschließend installiert SRT-AppGuard eine modifizierte Version. Erst jetzt können die Berechtigungen eingeschränkt werden. Nur leider funktioniert die App, aufgrund der Installation der modifizierten Version, bereits nicht mehr.

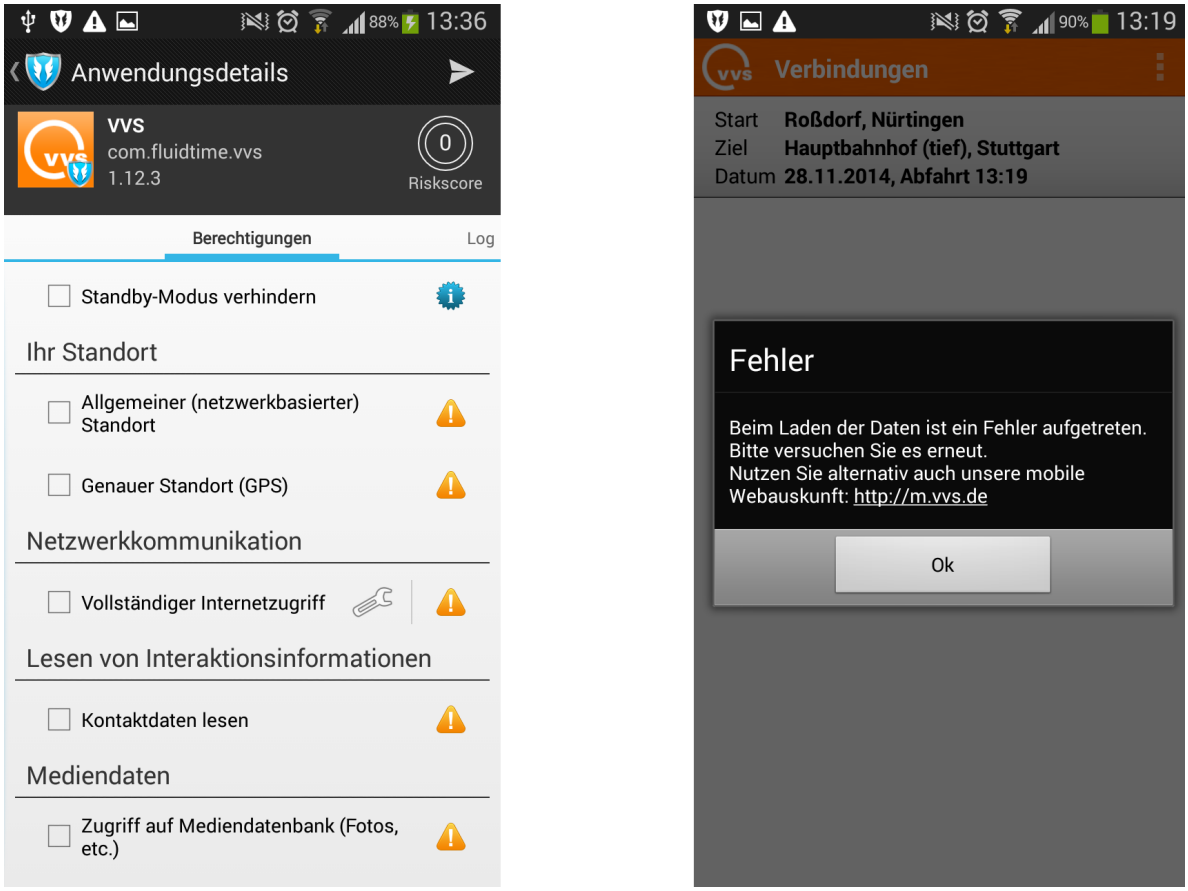

Abbildung 4.2: Beispiel entzogener Permissions an Hand der VVS-App.

App-Updates sind bei allen drei Systemen weiterhin möglich. Wobei beim SRT-AppGuard eine Besonderheit vorliegt. Denn hier muss das Google-Konto, welches mit dem App-Store verknüpft ist, zunächst über den SRT-AppGuard angegeben werden. Nur so können Updates gewährleistet werden, anderenfalls sind Updates nicht möglich.

Da es sich bei MOSES und Dr. Android & Mr. Hide noch um Prototypen handelt, sind aktuell keine benötigten Berechtigungen gelistet.

Benötigte Berechtigungen werden bei SRT-AppGuard erläutert. Das Testsystem benötigt fünf Berechtigungen:

1. Telefonstatus
2. USB-Speicher lesen/schreiben (wird benötigt um die App-Packete lesen zu können und umzuschreiben)
3. Zugriff auf Google-Konten (um Updates zu ermöglichen/optional)
4. Internet (um Updates zu ermöglichen und um den Lizenzschlüssel zu aktivieren)

5. Aktive Apps abrufen/schließen

Was zu dem erwähnt werden muss, ist dass Dr. Android & Mr. Hide nicht mit allen Apps kompatibel ist. Das Gleiche gilt auch für SRT-AppGuard. Vorinstallierte Apps können nicht überwacht werden, da kein Zugriff auf den Quellcode hergestellt werden kann. Aufgrund der Arbeitsweise der Systeme SRT AppGuard und Dr. Android & Mr. Hide kommt es zu einer weiteren Feststellung. Bei SRT-AppGuard und Dr. Android & Mr. Hide werden Apps doppelt gespeichert (damit ist die .apk-Datei gemeint). Einmal die Original-Version und einmal die modifizierte.

Bei SRT-AppGuard ist kein Zugriff auf Google-Maps möglich wenn die App überwacht wird. Es muss in den Einstellungen das Google-Konto angegeben werden, um Updates der Apps zu ermöglichen. Anderenfalls wird der User, bei vorhanden Updates überwachter Apps, nicht informiert.

Zum Schluss ist noch erwähnenswert, dass auch Mitschang und Stach einen Prototypen zur Gewährleistung von Privacy entwickelt haben, das Privacy Managment System (PMP). Das Projekt wird ständig erweitert, so auch bald durch Security Features.

Anders als bei den zuvor genannten Systemen arbeitet PMP in erster Linie mit Hilfe von Ressourcen. Diese Ressourcen ersetzen die gegenwärtigen Berechtigungsanfragen einer App. In Folge dessen ist eine feinere Granularität der Berechtigungen möglich. Berechtigungen können somit auf ein Minimum beschränkt werden und auch Fake-Data kann hier gesendet werden. Hierbei muss jedoch beachtet werden, dass App-Entwickler ihre Apps an diese, von PMP gegebenen, Ressourcen anpassen müssten. Das PMP ist aktuell nicht, auf im App-Store heruntergeladene Apps, anwendbar. Aus diesem Grund wird hier nicht weiter auf PMP eingegangen, [SM$^+$14] [SM13] [SM$^+$14].

4.3 Fazit

Trotz vieler verschiedener Ansätze und Versuche, Privacy für den Nutzer zu gewährleisten, ist noch kein vollständig ausgereiftes System für Android auf dem Markt zugänglich. Es können nicht alle Anforderungen, die an ein Privacy-System gestellt werden, auch wirklich erfüllt werden. Zum Teil sind Ansätze in Hinsicht auf die Rechtslage umstritten. Weshalb, das bereits im App-Store zugänglich gewesene, System SRT-AppGuard entfernt wurde. Andererseits sind Ansätze nicht vollkommen ausgereift und nur als Prototyp vorhanden. Da noch keine zufriedenstellende Alternative zu SRT-AppGuard für den Nutzer zugänglich ist, empfehle ich trotzdem diese für den privaten Gebrauch. Denn die meisten Anforderungen konnten zumindest von SRT-AppGuard erfüllt werden.

5 Trust-Systeme

Die Gliederung folgt analog den beiden vorherigen Kapiteln. Zunächst werden die Ansätze von Android zu Trust betrachtet, dann alternative Ansätze und schließlich folgt ein Fazit.

5.1 Ansätze von Android

Direkte Vorgänge, um den Trust in eine App zu bestimmen, gibt es im Android System noch nicht. Einen bereits existenten Ansatz dafür stellen Reputationssysteme dar, die meist in den App-Stores integriert sind. Benutzer können in diesen Apps bewerten, meist auf einer Skala von 1 bis 5 Sterne. Darüber hinaus hat der Benutzer die Möglichkeit ein Review in textueller Form zu hinterlassen, in welcher er seine Bewertung ausführlich erklären können. Zusätzlich ist anderen Benutzern möglich diese schriftlichen Bewertungen auf ihre Nützlichkeit zu bewerten. Ein Beispiel für ein solches Reputationssystem im Google Play Store findet sich in Abbildung 5.1.

Auch wenn diese Bewertungen als Anhaltspunkt für den Trust in eine App dienen können, sind sie doch keinesfalls ausreichend. Die Bewertung von Apps fällt laut AppBrain.com zum einen durchaus großzügig aus (Anfang Dezember lag die durchschnittliche Wertung von Apps im Google Play Store bei 4,0 von 5 Sternen), zum anderen mangelt es bei vielen Apps an ausreichenden Bewertungen (über 38% der Apps hatten zum gleichen Zeitpunkt weniger als drei Bewertungen).

Ein Problem der Reputationssysteme stellen jedoch Angriffe dar. Um Etwa im Google Play Store eine Bewertung abgeben zu können, muss man nur ein Google-Profil besitzen und die App auf einem Gerät installiert haben. „Bad Mouthing“, das Senken der Wertung einer App durch zahlreiche Abgabe fälschlich schlechter Bewertungen, und „Ballot-Stuffing“, die Erhöhung der Wertung einer App durch bewusst zu gute Bewertungen, stellen dadurch durchaus mögliche Angriffsmethoden dar.

Abbildung 5.1: Beispiel[a] für eine Bewertung im Google Play Store.

[a] Quelle: https://play.google.com/store/apps/details?id=com.facebook.katana

	Funktionalität	Gefahrenpotenzial	Private Daten	Skala
Yan et al.	ja	nein	nein	0% - 100%
Kuehnhausen und Frost	ja	ja	ja	0% - 100%
Dini et al.	ja	ja	ja	3 Stufen
AppGuard RiskScore	nein	ja	nein	0,0 - 10,0
McAfee EMM	nein	ja	ja	4 Stufen
TrustGo	nein	ja	ja	5 Stufen

Tabelle 5.1: Übersicht über vorhandene Ansätze für die Berechnung eines Trust-Wertes.

Darüber hinaus kamen Chia et al. zu der Erkenntnis, dass sich in den Bewertungen einer App primär ihre Funktionalität widerspiegelt. Andere Aspekte wie das Gefahrenpotenzial spielen dabei eine eher untergeordnete oder gar keine Rolle, [CYA12].

Eine befriedigende Lösung bieten Reputationssysteme also nicht, um den Trust in eine App zu bestimmen. Sehr schlechte Wertungen genauso wie Reviews können zwar vor bösartigen oder untauglichen Apps warnen, dabei muss sich der Benutzer jedoch darauf verlassen, dass er den anderen Benutzern und ihren Einschätzungen trauen kann und dies ist nicht garantiert. Außerdem stehen nicht immer genügend Bewertungen zur Verfügung, um eine Erkenntnis aus diesen zu schließen.

Mit der Analyse wie weit man den Bewertungen in Reputationssystemen vertrauen kann und welche anderen Informationen verwendet werden können, um den Trust in eine App zu berechnen, beschäftigen sich einige wissenschaftliche Ansätze. Diese Betrachten und bewerten wir im nächsten Unterkapitel.

5.2 Lösungsansätze

Der Trust in eine App kann bei der heutigen großen Auswahl an Apps ein Argument dafür sein, für welche App der Benutzer sich entscheidet. Eine Studien von Yan et al. [YLNY13] hat gezeigt, dass das Anzeigen von Trust-Informationen das Benutzerverhalten von Apps beeinflussen kann. Im Folgenden werden drei Ansätze zur Berechnung eines Trust-Wertes in Android analysiert. Da diese Ansätze jedoch nicht über den Status eines Prototypen hinaus kamen und selbst diese nicht zur Verfügung stehen, werden ergänzend noch Privacy- und Security-System betrachtet, die Risiko-Werte für Apps berechnen oder diese in Risiko-Kategorien einteilen und damit die Faktoren Gefahrenpotenzial und Missbrauch privater Daten abdecken können. Tabelle 5.1 gibt eine Übersicht der betrachteten Systeme und Ansätze.

Zunächst betrachten wir die Ansätze für reine Trust-Systeme beziehungsweise Metriken. Yan et al. entwickelten *TruBeRepec*, welches einen Nutzung-basierten Vertrauenswert errechnet und damit entsprechend ihrer Definition, die bereits in Unterkapitel 2.3 betrachtet wurde, sich rein auf die Funktionalität der App beziehen. Die anderen Faktoren für Trust in eine App nach der Definition dieser Arbeit ist damit nicht nachgekommen, [YLNY13].

Die Ansätze von Kuehnhausen und Frost [KFM12] sowie *MAETROID* Dini et al. [DMM+13] sind in ihrem Ansatz breiter gefächert und decken alle drei Faktoren aus, die als für den Trust in einer App wichtig, ausgemacht wurden.

Die Ansätze von Yan et al. sowie Kuehnhausen und Frost liefern jeweils einen Vertrauens-Wert zurück, der auf einer Skala von *0%* bis *100%* liegt. Im Vergleich dazu liefert der Ansatz von Dini et al. als Ergebnis, ob eine Applikation entweder *vertrauenswert*, *nicht vertrauenswürdig* oder *trügerisch* ist (orig. *trusted*; *untrusted*; *deceptiv*) ist. Dadurch wird dem Benutzer ein klarer Vorschlag gemacht, ob er eine App installieren sollte oder nicht. Eine Unterscheidung, in welche App man mehr Trust hat, ist zwischen zwei Apps, die in die gleiche Kategorie eingeordnet wurden, nicht möglich. Im Gegenzug ist mit einer prozentuale Skala dem Benutzer eine gute Vergleichsmöglichkeit zwischen Apps, liefert jedoch keine klar erkenntliche Information, ob nun genug Trust in eine App besteht, dass ich sie verwenden will oder nicht. Diese finale Entscheidung wird dem Benutzer überlassen, was keine ideale Lösung ist.

TruBeRepec berechnet den Trust-Wert einer App aus dem Nutzerverhalten dieser App, also wie lange die Nutzer diese App verwenden, wie viel sie die App im Vergleich zu anderen Apps verwenden und ähnliches. Nach Yan et al. besteht ein Zusammenhang zwischen Nutzung einer App und ihrer Funktionalität beziehungsweise Qualität, weshalb sich aus den Nutzerverhalten der Trust von Nutzern in eine App schlussfolgern lässt.

Kuehnhausen und Frost beziehen für ihren Trust-Wert die Bewertungen einer App, Reviews (beide aus dem Google Play Store entnommen) und die Permissions, welche die App verlangt, in ihre Bewertung ein. Für Ratings und Reviews wird dabei ein Glaubwürdigkeitswert berechnet, der den rein aus den Attributen berechneten Wert anpasst, da man sonst bedingungslos den Bewertern vertrauen würde, dass sie keine Absicht haben die Ergebnisse zu manipulieren. Die Permissions werden nach der Anforderung sicherheitskritischen Permissions und der Anforderung von Kombinationen sicherheitskritischer Permissions bewertet. Durch die Einbeziehung von Reviews, Ratings und Permissions deckt diese Metrik in etwa alle von uns verlangten Bereiche ab. Jedoch ist keine wirkliche Differenzierung zwischen den drei Bereichen möglich. Gefahrenpotenzial und Bedrohungen für private Daten können über die Permissions berechnet werden, doch es ist keine wirkliche Unterscheidung möglich. Auch wenn die Werte aus den Ratings, Reviews und Permissions einzeln gewichten lassen, fällt es hier doch schwer dies auf die individuell gesehene Wichtigkeit der einzelnen Bereiche zu übertragen.

In die Berechnung des Trust durch *MAETROID* gehen ein Bedrohungswert (berechnet aus den verlangten Permissions), der Entwickler, die Anzahl der Downloads der Applikation, der App-Store aus welchem die App bezogen wurde und das Rating der App ein. Dieses statisch berechnete Ergebnis kann durch manuelles Feedback zum Fehlverhalten der Applikation (Absturz; erhöhter Batterieverbrauch; (schlechte) Usability; Guthabenverlust; Bugs) dynamisch angepasst werden. Auch dieser Ansatz deckt durch seine breite Fächerung bei der Wahl der einfließenden Werte alle drei von uns verlangten Bereiche ab. Gerade durch das gegebene Feedback kann ein besseres Ergebnis erzielt werden, da wir präzisere Erkenntnisse haben, als wenn wir nur Ratings oder Permissions betrachten. Jedoch wird dem Benutzer keine Einsicht gewährt, welche Probleme mit der App tatsächlich gemeldet werden, was ihn unnötig im Unwissen lässt.

Der Benutzer kommt während der Berechnung des Trust-Wertes in Unterschiedlicher Form mit den einzelnen Systemen in Kontakt. Für die Messung des Nutzerverhaltens durch *TruBeRepec* muss

Software auf den Smartphones der Benutzer installiert werden. Diese sendet dann automatisch anonymisierte und gesicherte Daten an die Server. Der Benutzer ist damit - außer natürlich in dem er die nötige Software installiert und Apps nutzt - nicht direkt in den Berechnungsprozess eingebunden. Bei der Methode von Kuehnhausen und Frost kommt der Benutzer während dem Berechnungsvorgang dabei nicht mit der Metrik in Verbindung. Einzig seine Bewertung und seine Reviews werden aus dem Store entnommen und das Ergebnis wird im dann im Store oder auf einer Website präsentiert werden. Das Feedback, die große Stärke des Ansatzes von Dini et al. ist auch eine Schwäche. Das Feedback baut darauf, dass Benutzer die Ereignisse, über welche sie Rückmeldung geben bewusst wahrnehmen. Gerade bei erhöhtem Batterieverbrauch und Guthabensverlust stellt sich die Frage, wie viele dies wirklich feststellen. Dies dann auch einer App zuordnen zu können ist keine Selbstverständlichkeit. Auch wenn das Feedback gegen Unfair Rating Attacks in Form bewusst falsche Meldungen abgesichert ist, stellt sich doch die Frage, wie viel Feedback zu erwarten ist und wie korrekt dieses ist. Hier lastet viel Verantwortung auf dem Benutzer.

Alle drei Systeme haben ihre Stärken und Schwächen. *TruBeRepec* ist unbefriedigend, da es nur den Bereich der Funktionalität abdecken, modelliert diese aber am besten. Kuehnhausen und Frost sind davon abhängig, dass ein Reputationssystem zur Verfügung steht, dessen Bewertungen und Reviews analysiert werden können, da sonst lediglich die verwendeten Permissions in den Wert einfließen können, wodurch nur noch eine Risikoabschätzung übrig bleibt, wie viele Security- und Privacy-Syteme sie als Feature besitzen. *MAETROID* ist der vielversprechendste Ansatz, jedoch nur wenn eine solide Benutzerbasis aufgebaut werden kann, um ein brauchbares Feedback zu garantieren. Außerdem sollten die Ergebnisse des Feedback dem Benutzer direkt kommuniziert werden und nicht nur verwendet werden, um eine App möglicherweise in eine alternative Vertrauenskategorie einordnen zu können.

Keines der Systeme kann separate Werte für den Trust in einen der drei definierten Bereiche liefern und so ist eine Gewichtung dieser natürlich unmöglich. Außer wenn das Feedback in *MAETROID* dem Benutzer Verfügbar gemacht wird, bietet keines der Systeme eine befriedigende Möglichkeit dem Benutzer das Zustandekommen des Trust-Wertes zu erläutern, zumeist weil dies auch nicht die Intention der Entwickler war.

Das größte Manko bleibt wie bereits erwähnt, dass keines der Systeme über den Status des Prototypen hinaus kam und selbst die Prototypen nicht zur Verfügung steht. Aktuell verfügbare Methoden um den Trust in eine App berechnen zu können lassen sich als Features für Security- und Privacy-Systeme finden. Die Funktionalität der App wird dabei nicht betrachtet, da sie für diese Systeme eine Untergeordnete Rolle spielt.

AppGuard, welches bereits in Kapitel 4 analysiert wurde, berechnet für alle Apps eine Riskscore, welche sich aus den verwendeten Permissions einer App zusammensetzt (siehe Abbildung (5.2). Entzieht der Benutzer der App eine oder mehrere Permissions passt sich diese Riskscore an. Das Problem bei diesem Ansatz ist, dass Apps mit breitem Funktionsspektrum entsprechend viele und kritische Permissions verlangen, wodurch ihre Riskscore sehr hoch ist. Chia et al. [CYA12] stellten fest, dass gerade populäre Apps dies ihren vielfältigen angebotenen Funktionen verdanken und der Benutzer somit an viele benötigte Permissions und damit an viele Apps mit einer hohen Riskscore gewöhnt ist. Haben die meisten Apps eine hohe Riskscore ist es kaum möglich mit Hilfe dieser den Trust in eine App zu bestimmen.

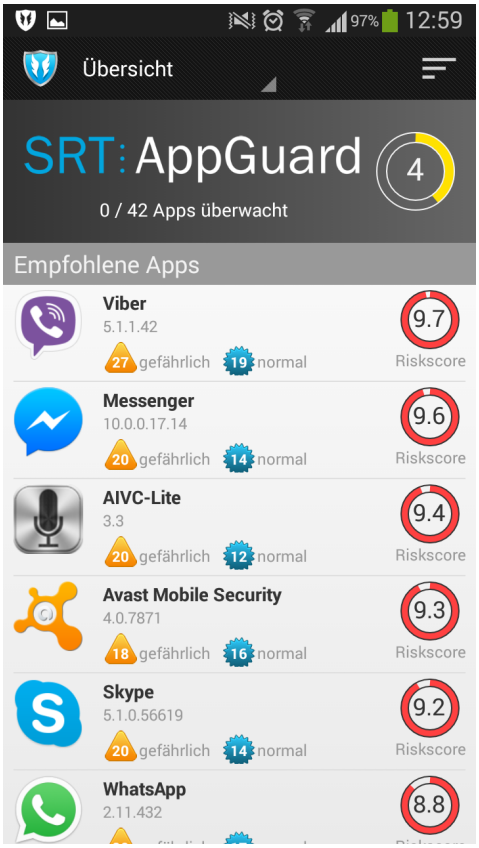

Abbildung 5.2: Riskscore von AppGuard.

McAfee Ordnet Apps Trust-Kategorien zu, abhängig vom Trust, den man in die App bezüglich der Security und der Data Privacy hat. Diese Trust-Werte entspringen einer automatischen Analyse, in die keine Einsicht gewährt wird, weswegen keine Analyse und Bewertung möglich ist. Die Trust-Kategorien reichen von *Safe* über *Low-risk* hin zu *Suspicious* und schließlich *Malicious*. Anzumerken ist, dass diese Feature nur für die Enterprise Version McAfee EMM und damit dem privaten Anwender nicht zur Verfügung steht, [Shu13].

TrustGo führt ebenfalls Analysen durch, um Apps in Trust-Kategorien einzuteilen. Auch wenn *TrustGo* selber die Kategorien „Security Level“ nennt[1], trifft Trust eher zu. Apps, die Zugriff auf persönliche Daten haben oder den Benutzer Geld kosten können werden beispielsweise als High Risk eingeordnet, wobei diese App in diesem Moment nur das Schadpotenzial hat, aber ein wahrhaftiger Missbrauch nicht zwangsweise beobachtet wurde..

Eine Bewertung über die Qualität der beiden Ansätze von *McAfee* und *TrustGo* fällt weg, da wie bereits erwähnt kein Einblick in die Vorgänge der jeweiligen Analysen möglich ist. Sie stellen aber interessante Ansätze dar.

[1]Quelle: http://www.trustgo.com/security-levels

5.3 Fazit

Das Android-System selber besitzt keine effektiven Mittel, um den Trust in eine App zu bestimmen. Die Reputationssysteme in Stores können zur groben Orientierung dienen, aber mehr auch nicht. Um den Trust in eine App zu bestimmen bedarf es anderer Mittel.

Die Forschung hat Ansätze für die Berechnung eines Trust-Wertes einer App geliefert, aber Momentan sind diese nicht über den Status des Prototypen hinaus und bieten so keine Lösung.

Security- und Privacy-Systeme besitzen oft eine Riskscore oder eine Einteilung in Trust-Kategorien als Feature. Die Riskscore erbt dabei das Problem der Permissions aus denen sie berechnet wird und zwar, dass es ihr an Aussagekraft fehlt. Die Einteilung in die Trust-Kategorien stellt noch den besten Ansatz, auch wenn hier die Informationen fehlen, um diese Einteilung bewerten zu können.

Ein Trust-System, welches unseren Anforderungen entspricht existiert zur Zeit nicht auf dem Markt. Es bestehen jedoch gute Ansätze und es bleibt abzuwarten, wie sich diese Entwickeln.

6 Fazit

In den Kapiteln 3 bis 5 haben wir den aktuellen Stand der Forschung und Technik bezüglich Trust, Security und Privacy in Android Apps analysiert. Dabei haben wir zunächst die Ansätze betrachtet, die Android selber liefert. Diese bieten nicht immer für alle Problemstellungen eine Lösung. Deshalb bedarf es alternativer Systeme, die sich damit befassen.

In Bezug auf Security-Probleme gibt es eine Vielzahl an Antiviren-Apps, die diesen Bedrohungen beikommt. Dabei ist die Arbeitsweise und auch der Erfolgsgrad der Systeme sehr ähnlich. Die wenigen Unterschiede ergeben sich meistens erst durch bezahlten Varianten, dieser ansonsten kostenlosen Software. Der grundlegende Schutz ist jedoch bei den Gratisversionen bereits vorhanden. Eine Security-App auf dem Android-Gerät zu installieren ist durchaus ratsam, dabei spielt die Wahl welche App nun genau genommen wird keine erhebliche Rolle.

Bei Privacy Systemen gibt es eine überschaubare Menge an Ansätzen. Die meisten davon befinden sich darüber hinaus erst im Zustand des Prototypen. Auf Grund der aktuellen Rechtslage befinden sich Systeme wie SRT-AppGuard und Dr. Android & Mr. Hide in einer rechtlichen Grauzone, da sie den Quellcode von Apps umschreiben. Abgesehen davon ist SRT-AppGuard der vielversprechenste Ansatz und kann für den privaten Gebrauch empfohlen werden. SRT-AppGuard erfüllt die meisten der von uns gestellten Anforderungen. Auch wenn die Bedienung von AppGuard recht einfach ist, wird doch ein gewisses Vorwissen verlangt, damit es effektiv verwendet werden kann.

Trust Systeme könnte darüber hinaus eine sinnvolle Erweiterung darstellen. Gerade in Hinsicht darauf, von welchen Apps wir mit AppGuard Zugriffe verbieten wollen, wäre eine effektive Trust-Metrik für Apps von Vorteil. Leider gibt es hierzu aktuell keine befriedigenden eigenständigen Lösungen die dem Status des Prototypen entwachsen sind. Einige Privacy- und Security-Systeme verwenden Risk-Scores welche sie Apps zuweisen und Risk-Kategorien in welche sie Apps einteilen. Wie weit auf diese Ansätze Verlass ist, bleibt dabei dahin gestellt, da kein Einblick in das Zustandekommen möglich war.

Um der Großzahl der Bedrohungen für private Daten und das System auf Android-Geräten entgegen zu kommen, ist es empfehlenswert eine Kombination aus Antiviren-App und Privacy-System auf dem Gerät zu installieren.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden die Begriffe Trust, Security und Privacy für Android definiert und Anforderungskataloge wurden aufgestellt, mit den Erwartungen an ein System, das sich auf einen dieser Bereiche konzentriert.

Danach wurde jeweils analysiert, welche Lösungen Android selber liefert, um den Problemen beizukommen. Diese sind jedoch in den meisten Fällen nicht ausreichend. Deswegen wurden alternative Ansätze untersucht, auf ihre Tauglichkeit überprüft und mit dem aufgestellten Anforderungskatalog abgeglichen.

Da es eine Vielzahl an Security-Anwendungen auf dem Android-Markt gibt, werden hier mehrere Antivirus Suites oberflächlich behandelt. Außerdem wurde auf einige Security-Systeme eingegangen, die spezifische Aufgaben erfüllen, die über jener einer Antivirus-App hinaus gehen.

Beim Befassen mit Privacy-Systemen wurde der Schwerpunkt auf drei Systeme gelegt. Zu den näher betrachteten Systemen gehören Prototypen von MOSES und Dr. Android & Mr.Hide sowie auch eine mehr oder weniger ausgereifte Version von SRT-Appguard. Nachdem kurz die Arbeitsweise der genannten Systeme beschrieben wurde, wird auf Gemeinsamkeiten und Unterschiede eingegangen. Anschließend wird beschrieben in wie weit, vor allem bei SRT-Appguard, Anforderungen erfüllt werden. Zuletzt wird ein Fazit zur Privacy-Sicherung gezogen.

Auch wenn es vielversprechende Ansätze zu Trust-Systemen gibt, stehen aktuell keine eigenständigen zur Verfügung. Deswegen wurden Security- und Privacy-Systeme betrachtet, die eine Risk-Score oder Risik-Kategorien für Apps bieten. Eine befriedigende Lösung für unsere Anforderungen wurde jedoch nicht gefunden, da diese Systeme nicht auf die Funktionalität der Apps eingehen.

Abschließend wurde eine Empfehlung gegeben, mit welchen Systemen ein Android-Nuter sein Gerät gegen Security- und Privacy-Bedrohungen absichern kann.

Ausblick

Das PMP stellt einen vielversprechenden Ansatz dar. Es wird ständig um neue Komponenten erweitert. Durch einige dieser anstehenden Erweiterungen bezieht sich PMP danach auf Privacy, Security und Trust. Diese wären damit erstmalig in einem System vereinigt und bildet eine Alternative zu den in dieser Arbeit vorgestellten Insellösungen für die einzelnen Bereiche.

Der Google-Store erhält auch Neuerungen, die eine bessere Identifikation von Malware gewährleistet. Das neue Android 5.0 L bietet, wie bereist erwähnt, neue Ansätze bezüglich Privacy- und Security-Problemen. In diese Richtung könnten weitere Nachforschungen angestellt werden.

Literaturverzeichnis

[AR98] P. E. Agre, M. Rotenberg. *Technology and privacy: The new landscape*. Mit Press, 1998. (Zitiert auf Seite 10)

[Bac12] M. Backes. New Approach Uncovers Data Abuse on Mobile End Devices. Technischer Bericht, University Saarland, 2012. (Zitiert auf Seite 5)

[BGH+12] M. Backes, S. Gerling, C. Hammer, M. Maffei, P. von Styp-Rekowsky. AppGuard - real-time policy enforcement for third-party applications. Technischer Bericht, SaarlÃ¤ndische UniversitÃ¤ts- und Landesbibliothek, Postfach 151141, 66041 Saarbruecken, 2012. URL http://scidok.sulb.uni-saarland.de/volltexte/2012/4902. (Zitiert auf Seite 19)

[BGH+13] M. Backes, S. Gerling, C. Hammer, M. Maffei, P. von Styp-Rekowsky. AppGuard–Enforcing User Requirements on Android Apps. In *Tools and Algorithms for the Construction and Analysis of Systems*, S. 543–548. Springer, 2013. (Zitiert auf Seite 19)

[CYA12] P. H. Chia, Y. Yamamoto, N. Asokan. Is This App Safe?: A Large Scale Study on Application Permissions and Risk Signals. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, S. 311–320. ACM, New York, NY, USA, 2012. doi:10.1145/2187836.2187879. URL http://doi.acm.org/10.1145/2187836.2187879. (Zitiert auf den Seiten 28 und 30)

[DMM+13] G. Dini, F. Martinelli, I. Matteucci, M. Petrocchi, A. Saracino, D. Sgandurra. Evaluating the Trust of Android Applications through an Adaptive and Distributed Multi-criteria Approach. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, S. 1541–1546. 2013. doi:10.1109/TrustCom.2013.189. (Zitiert auf Seite 29)

[Due13] K. Duell. Samsung Knox - Der naechste Meileinstein mit Technik von NSA und Centrify?, 2013. URL http://goo.gl/xZYfNt. (Zitiert auf Seite 18)

[Gam88] D. Gambetta. Trust: Making and breaking cooperative relations. 1988. (Zitiert auf Seite 11)

[HWSB13] P. Holtewert, R. Wutzke, J. Seidelmann, T. Bauernhansl. Virtual Fort Knox Federative, Secure and Cloud-based Platform for Manufacturing. *Procedia {CIRP}*, 7(0):527 – 532, 2013. doi:http://dx.doi.org/10.1016/j.procir.2013.06.027. URL http://www.sciencedirect.com/science/article/pii/S2212827113002965. Forty Sixth {CIRP} Conference on Manufacturing Systems 2013. (Zitiert auf Seite 17)

[JMV+12] J. Jeon, K. K. Micinski, J. A. Vaughan, A. Fogel, N. Reddy, J. S. Foster, T. Millstein. Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, S. 3–14. ACM, New York, NY, USA, 2012. doi:10.1145/2381934.2381938. URL http://doi.acm.org/10.1145/2381934.2381938. (Zitiert auf Seite 19)

[Jøs96] A. Jøsang. The Right Type of Trust for Distributed Systems. In *Proceedings of the 1996 Workshop on New Security Paradigms*, NSPW '96, S. 119–131. ACM, New York, NY, USA, 1996. doi:10.1145/304851.304877. URL http://doi.acm.org/10.1145/304851.304877. (Zitiert auf Seite 11)

[KFM12] M. Kuehnhausen, V. Frost, G. Minden. Framework for assessing the trustworthiness of cloud resources. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on*, S. 142–145. 2012. doi:10.1109/CogSIMA.2012.6188367. (Zitiert auf Seite 29)

[kil] Kill Switch macht Handy- Diebstahl unattraktiv. URL http://www.krone.at/Digital/Kill_Switch_macht_Handy-Diebstahl_unattraktiv-Zahlen_belegen_-Story-408897. (Zitiert auf Seite 14)

[kno14] Meet evolving enterprise mobility challenges with Samsung KNOX, 2014. URL http://goo.gl/jLqFON. (Zitiert auf Seite 17)

[RCCF12] G. Russello, M. Conti, B. Crispo, E. Fernandes. MOSES: Supporting Operation Modes on Smartphones. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, SACMAT '12, S. 3–12. ACM, New York, NY, USA, 2012. doi:10.1145/2295136.2295140. URL http://doi.acm.org/10.1145/2295136.2295140. (Zitiert auf Seite 19)

[Shu13] A. Shukla. Introducing App Reputation for Android Apps, 2013. URL http://blogs.mcafee.com/business/security-connected/introducing-app-reputation-for-android-apps. (Zitiert auf Seite 31)

[SM13] C. Stach, B. Mitschang. Privacy Management for Mobile Platforms – A Review of Concepts and Approaches. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, Band 1, S. 305–313. 2013. doi:10.1109/MDM.2013.45. (Zitiert auf den Seiten 6 und 25)

[SM+14] C. Stach, B. Mitschang, et al. Design and Implementation of the Privacy Management Platform. In -. Auenwald: IEEE Computer Society Conference Publishing Services, 2014. (Zitiert auf Seite 25)

[YLNY13] Z. Yan, C. Liu, V. Niemi, G. Yu. Exploring the Impact of Trust Information Visualization on Mobile Application Usage. *Personal Ubiquitous Comput.*, 17(6):1295–1313, 2013. doi:10.1007/s00779-013-0636-4. URL http://dx.doi.org/10.1007/s00779-013-0636-4. (Zitiert auf Seite 28)

[YZD12] Z. Yan, P. Zhang, R. Deng. TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applicationsnull. *Personal and Ubiquitous Computing*, 16(5):485–506, 2012. doi:10.1007/s00779-011-0420-2. URL http://dx.doi.org/10.1007/s00779-011-0420-2. (Zitiert auf Seite 11)

Alle URLs wurden zuletzt am 02. 12. 2014 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift