Technische Hochschule Ingolstadt

Fakultät Elektrotechnik und Informatik

Studiengang Informatik (M.Sc.) - Schwerpunkt Safety & Security

Masterarbeit

Thema: Entwicklung eines Modells zur Beurteilung des (aktuellen) Sicherheitslevels eines Smartphones einschließlich der Bewertung von mobilen Anwendungen hinsichtlich deren Sicherheit

Vor- und Zuname: Sebastian Stefan Schuster

ausgegeben am: 11. Oktober 2017

abgegeben am: 17. November 2017

Erstprüfer:	Prof. Dr.-Ing. Hans-Joachim Hof
Zweitprüfer:	Prof. Dr.-Ing. Ernst-Heinrich Göldner
Betreuer optimaBIT GmbH:	Dr. Bruce J. Sams

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebene Quellen oder Hilfsmittel benutzt, sowie wörtliche oder inhaltlich übernommene Stellen als solche gekennzeichnet habe.

München, den 17. November 2017

Kurzfassung

Die Verbreitung von Smartphones in Deutschland und den westlichen Industrieländern hat im Jahr 2017 einen „De-Facto-Standard“ erreicht. Während zu Beginn der Ära von Smartphones noch die Nutzung der ursprünglichen Funktionen eines Mobiltelefons für Telefonie- und Nachrichtenzwecke im Fokus stand, veränderte sich durch die Einführung des mobilen Internets die Verwendung von Smartphones dramatisch. Beginnend mit dem Abrufen von E-Mails und der Verbreitung von Instant Messaging, nahmen die Verwendungsmöglichkeiten stetig zu. Im Jahr 2017 stellt das mobile Bezahlen, sowie das Abwickeln von Bankgeschäften keine Seltenheit mehr dar. Durch die zunehmende Verarbeitung von sensitiven und kritischen Daten auf Smartphones im privaten und geschäftlichen Umfeld, erhöhen sich die Auswirkungen im Falle eines erfolgreichen Angriffs. Dies lässt sich durch eine Zunahme an mobiler Malware und an einer steigenden Anzahl an Meldungen über hohe bis kritische Schwachstellen bei Smartphones begründen. Ein etabliertes Vorgehen zur Überprüfung der Sicherheit im mobilen Bereich ist der Penetrationstest von mobilen Anwendungen. Der Fokus von mobilen Penetrationstests liegt in der Regel auf Anwendungsebene und kann nach der Überprüfung keine Aussage über die Gesamtsicherheit der Daten auf einem Gerät treffen. Um der zunehmenden Bedrohung von Smartphones durch Schwachstellen und dem Fehlen eines Modells zur Beurteilung der Gesamtsicherheit eines Smartphones entgegenzutreten, wird in dieser Arbeit ein Sicherheitsmodell zur Bestimmung des (aktuellen) Sicherheitslevels eines Smartphones, einschließlich der Bewertung von mobilen Anwendungen hinsichtlich deren Sicherheit, entwickelt. Nach Anwendung des Modells auf drei unterschiedliche Nutzergruppen mit zwei verschiedenen Smartphones zeigte sich, dass das ausgewählte iOS-Gerät bei allen Nutzergruppen über ein höheres Sicherheitslevel als das Android-Gerät verfügt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einleitung

Die Nutzung von Smartphones für private oder geschäftliche Zwecke stellt im Jahr 2017 mehr die Regel als die Ausnahme dar. Durch die permanente Internetanbindung erfreuen sich mobile Messenger, wie beispielsweise WhatsApp, vor allem bei Privatnutzern einer hohen Popularität. In den letzten Jahren nahm zusätzlich die Verwendung des Smartphones für mobile Online-Bestellungen zu. Einer Umfrage zufolge, verwendeten 2011 nur 23 Prozent der Nutzer ihr Smartphone für Onlineeinkäufe, bis zum Jahr 2016 stieg die Zahl auf 69 Prozent an[Bon]. Eine ähnliche Entwicklung ist beim mobilen Bezahlen festzustellen. Während das Volumen für mobile Bezahlungen bei Paypal 2011 noch bei 4 Milliarden US Dollar lag, stieg es bis 2016 auf 102 Milliarden US-Dollar an[Pay]. Die Entwicklung führte ebenfalls dazu, dass de facto alle deutschen Banken mobile Applikationen anbieten, welche zum Teil eine komplette Verwaltung des Bankkontos ermöglichen. Manche junge FinTech-Startups der Bankenbranche bieten den vollen Funktionsumfang eines Bankkontos ausschließlich über eine mobile Applikation auf einem Smartphone an[Zie17].
Im geschäftlichen Umfeld ist der Abruf von dienstlichen E-Mails via Smartphone seit vielen Jahren etabliert. Microsoft veröffentlichte 2015 seine Office-Anwendungen für die Plattformen Android und iOS, um ein mobiles Erstellen und Bearbeiten von Dokumenten zu ermöglichen. Unternehmensintern sind in den letzten Jahren zusätzlich Anwendungen entwickelt worden, welche die effiziente Durchführung individueller Unternehmensprozesse ermöglichen.

Mit Zunahme der Nutzungsmöglichkeiten von Smartphones stieg auch die Anzahl an kritischen Schwachstellen, welche die Sicherheit der Daten und Kommunikation auf Smartphones gefährdeten. Eine der populärsten Sicherheitslücken, welche 95 Prozent der damaligen Android-Geräte betraf, erschien im Juli 2015 und wurde *„Stagefright“* genannt. Sie ermöglichte einem Angreifer über eine präparierte Mediendatei die Ausführung von Schadcode mit privilegierten Rechten.[Eik]
Weitere populäre und kritische Schwachstellen erschienen unter den Namen *„Dirty Cow“*, *„Rowhammer“*, *„Cloak and Dagger“*, *„BlueBorne“* und *„BroadPwn“*. Zusätzlich existier(t)en eine Vielzahl an kritischen Schwachstellen, welche aus Übersichtsgründen nicht weiter aufgeführt werden.

Gleichzeitig ist ein Smartphone einer Vielzahl an neuen Bedrohungen ausgesetzt, welche bei Verwendung eines klassischen Computers nicht vorhanden waren. Während der Verlust eines Computers in der Regel nur durch einen Diebstahl möglich ist, erhöht sich das Risiko des physikalischen Verlusts bei Smartphones durch die permanente Mitführung und des deutlich kleineren Formats. Die Einführung von Schutzmechanismen, welche sich im Umfeld von Computern etabliert haben, ist ohne Adaption nicht möglich. So erhöht die Verwendung von Hardware-Firewalls auf Netzwerkebene die Sicherheit von Smartphones nur bedingt, da diese aufgrund der permanenten Mitführung eine Vielzahl an Netzwerkzugängen nutzen und die Hoheit (z.b. öffentlicher HotSpot) über diese Netze nicht durchgehend vorhanden ist. Die Installation von Antiviren-Lösungen erhöht die Sicherheit von Smartphones ebenfalls nur bedingt, da aufgrund der Architektur mobiler Betriebssysteme eine Analyse installierter Anwendungen teilweise nicht durchgeführt werden kann.

Parallel nimmt die Verarbeitung von sensiblen Daten auf mobilen Endgeräten, sowohl im privaten als auch im geschäftlichen Umfeld, weiter zu und erhöht damit die persönlichen und technischen Auswirkungen, sollte ein Gerät kompromittiert werden. Dieser Umstand sorgt dafür, dass ein Smartphone ein lohnendes Ziel für einen Angreifer darstellt. Das Ausnutzen einer Schwachstelle ermöglicht einem Angreifer seinen Schadcode zusätzlich auf einer Vielzahl an Geräten auszuführen. So betraf beispielsweise eine 2015 in Samsungs Tastaturanwendung aufgedeckte Schwachstelle insgesamt 600 Millionen Endgeräte[Wel]. Der von McAfee veröffentlichte *Mobile Threat Report 2016* stellt eine Zunahme von Malware für mobile Endgeräte von rund 300.000 Malwareanwendungen (Quartal 4 2014) zu über 1.4 Millionen (Quartal 4 2015) derartiger Anwendungen innerhalb eines Jahres fest[Sne16].

Unternehmen, welche ihre entwickelten Anwendungen in einem Penetrationstest untersuchen lassen, testen nur die Anwendung auf Schwachstellen. Eine individuelle Evaluation der Laufzeitumgebung findet in der Regel nicht statt. Ein derartiger Penetrationstest identifiziert zuverlässig mögliche Angriffsvektoren und Bedrohungen, welche von der getesteten Anwendung ausgehen, kann jedoch keine Aussage über die Gesamtsicherheit des Gerätes treffen. Dies schließt eine Beurteilung der Sicherheit aller gespeicherten Daten, einschließlich der Daten der getesteten Anwendung, ein.

Die Zunahme der Verarbeitung sensibler Daten auf Smartphones in Verbindung mit einem Anstieg an infizierten Geräten durch Schwachstellen bei mobilen Endgeräten sowie die fehlende Möglichkeit durch einen Penetrationstest die Gesamtsicherheit eines Smartphones zu beurteilen, erfordern die Implementierung eines Sicherheitsmodells für Smartphones. Dieses Sicherheitsmodell soll die Möglichkeit bieten, dass (aktuelle) Sicherheitsniveau eines ausgewählten Gerätes zu ermitteln. In Verbindung mit einem Penetrationstest der Anwendung(en), welche auf dem Gerät ausgeführt wird, lässt sich so die Gesamtsicherheit des Gerätes beurteilen.

Im Rahmen dieser Arbeit werden im Grundlagenteil (Kapitel 2), aufbauend auf den Kenntnissen eines Informatik Masterstudiums, alle für das Verständnis der Arbeit notwendigen Grundlagen vermittelt. Das Kapitel 3 Aktuelle Forschung liefert eine Einschätzung hinsichtlich einer Verwendung bereits existierender Sicherheitsmodelle für die Beurteilung des Sicherheitslevels eines Smartphones. Im Hauptteil (Kapitel 4) wird der Prozess zur

Erstellung des Sicherheitsmodells erläutert, das entwickelte Sicherheitsmodell vorgestellt und eine Anwendung des Modells auf drei unterschiedliche Nutzergruppen, mit jeweils zwei aktuellen Smartphones, durchgeführt. Der Fokus des Modells liegt in der Sicherheit des Gerätes, eine Beurteilung des Datenschutzes aus Sicht der Privatsphäre wird aufgrund der vielen unterschiedlichen nationalen Gesetzgebungen der einzelnen Länder nicht vorgenommen. Im 5. Kapitel wird ein systematisches Vorgehen für die Schwachstellensuche von mobilen Anwendungen im Rahmen eines Penetrationstests entwickelt. Die Masterarbeit schließt mit einer Zusammenfassung und erörtert neue Forschungsthemen und -fragen auf Basis dieser Arbeit.

2. Grundlagen

Dieses Kapitel legt eine Basis aller für diese Arbeit relevanten Grundlagen. Aufbauend auf den im Masterstudium Informatik erworbenen Kenntnissen, die als Voraussetzung zum Verständnis dieser Arbeit betrachtet werden, werden alle erforderlichen Begriffe und Erklärungen erläutert.

2.1 Begriffsdefinitionen

Dieser Abschnitt definiert wichtige Begriffe, welche für das Verständnis der Arbeit erforderlich sind.

Wert

> *„Werte sind alles, was wichtig für eine Institution ist (Vermögen, Wissen, Gegenstände, Gesundheit).“[Sicb]*

Vertraulichkeit

> *„Vertrauliche Informationen müssen vor unbefugter Preisgabe geschützt werden.“ [Sicf, S. 14]*

Verfügbarkeit

> *„Dem Benutzer stehen Dienstleistungen, Funktionen eines IT-Systems oder auch Informationen zum geforderten Zeitpunkt zur Verfügung.“[Sicf, S. 14]*

Integrität

> *„Die Daten sind vollständig und unverändert. Der Begriff „Information“ wird in der Informationstechnik für „Daten“ verwendet, denen je nach Zusammenhang bestimmte Attribute wie z. B. Autor oder Zeitpunkt der Erstellung zugeordnet werden können. Der Verlust der Integrität von Informationen kann daher bedeuten, dass diese unerlaubt verändert wurden oder Angaben zum Autor verfälscht wurden oder der Zeitpunkt der Erstellung manipuliert wurde.“[Sicf, S. 14]*

Verbindlichkeit

„Unter Verbindlichkeit werden die Sicherheitsziele Authentizität und Nichtabstreitbarkeit zusammengefasst. Bei der Übertragung von Informationen bedeutet dies, dass die Informationsquelle ihre Identität bewiesen hat und der Empfang der Nachricht nicht in Abrede gestellt werden kann.“[Sicb]

Zertifizierung

„Maßnahme durch einen unparteiischen Dritten, die aufzeigt, dass ein angemessenes Vertrauen besteht, dass ein ordnungsgemäß bezeichnetes Erzeugnis, Verfahren oder eine ordnungsgemäß bezeichnete Dienstleistung in Übereinstimmung mit einer bestimmten Norm oder einem bestimmten anderen normativen Dokument ist.“[05]

Validierung

„Bestätigung durch objektiven Nachweis, dass die Anforderungen für eine bestimmte Anwendung oder einen bestimmten Gebrauch erfüllt sind.“[15]

Risiko

„Auswirkung von Ungewissheit auf ein erwartetes Ergebnis.“[15]

Bedrohung

„Eine Bedrohung ist ganz allgemein ein Umstand oder Ereignis, durch den oder das ein Schaden entstehen kann. Der Schaden bezieht sich dabei auf einen konkreten Wert wie Vermögen, Wissen, Gegenstände oder Gesundheit. Übertragen in die Welt der Informationstechnik ist eine Bedrohung ein Umstand oder Ereignis, der oder das die Verfügbarkeit, Integrität oder Vertraulichkeit von Informationen beeinträchtigen kann, wodurch dem Besitzer bzw. Benutzer der Informationen ein Schaden entstehen kann. Beispiele für Bedrohungen sind höhere Gewalt, menschliche Fehlhandlungen, technisches Versagen oder vorsätzliche Handlungen. Trifft eine Bedrohung auf eine Schwachstelle (insbesondere technische oder organisatorische Mängel), so entsteht eine Gefährdung.“[Sicb]

Schwachstelle

„Eine Schwachstelle ist ein sicherheitsrelevanter Fehler eines IT-Systems oder einer Institution. Ursachen können in der Konzeption, den verwendeten Algorithmen, der Implementation, der Konfiguration, dem Betrieb sowie der Organisation liegen. Eine Schwachstelle kann dazu führen, dass eine Bedrohung wirksam wird und eine Institution oder ein System geschädigt wird. Durch eine Schwachstelle wird ein Objekt (eine Institution oder ein System) anfällig für Bedrohungen.“[Sicb]

Penetrationstest

„Ein Penetrationstest ist ein gezielter, in der Regel simulierter, Angriffsversuch auf ein IT-System. Er wird als Wirksamkeitsprüfung vorhandener Sicherheitsmaßnahmen eingesetzt.“[Sicb]

2.2 Modell

Dieses Kapitel erklärt die allgemeinen Grundlagen eines Modells und deren Erzeugung.

Allgemeine Modelltheorie

Die Bedeutung eines Modells - einschließlich der Modellierung - wird in der Wissenschaft sehr unterschiedlich betrachtet. Einen allgemeinen Ansatz zur Definition lieferte Stachowiak 1973 in seinem Buch „Allgemeine Modelltheorie“ ([Sta73]). Nach Stachowiak ([Sta73, S. 131-133]) besteht ein Modell aus drei Merkmalen:

- *Abbildungsmerkmal:* Ein Modell ist Abbild oder Vorbild
- *Verkürzungsmerkmal:* Ein Modell erfasst nur die relevanten Attribute
- *Pragmatisches Merkmal:* Ein Modell wird für seinen Verwendungszweck geschaffen

Der Prozess der Modellbildung wird in Abbildung 2.1 verdeutlicht. Das Original zeigt dabei einen Ausschnitt bzw. Abbild der Realität. Das Modell, welches für einen bestimmten Zweck erstellt wurde, gibt eine Perspektive aus Sicht des Subjekts auf das Original wieder. [Tho02]

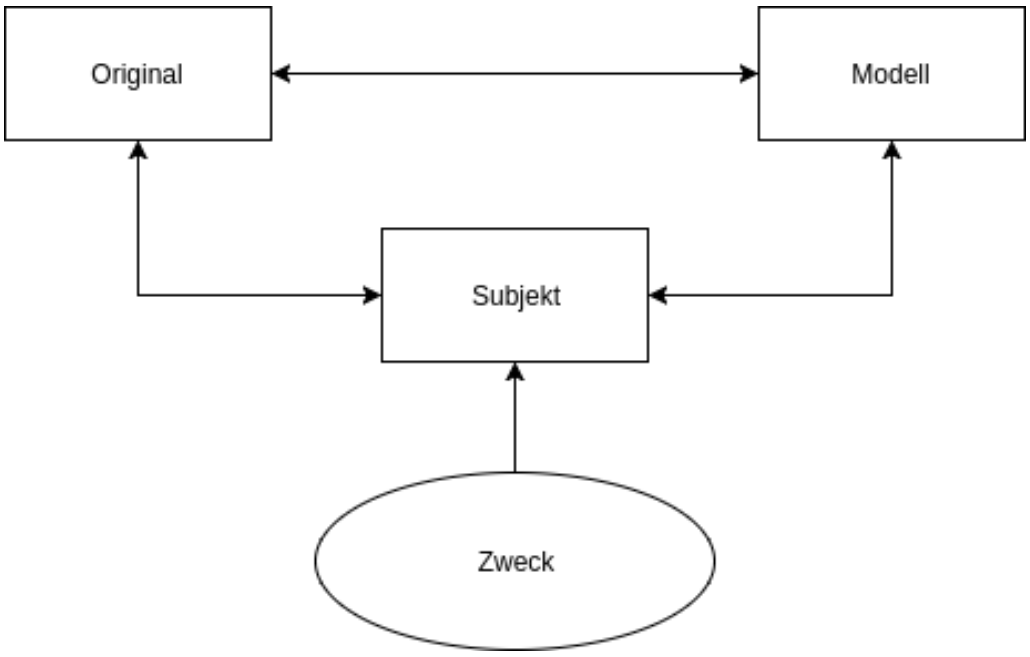

Abbildung 2.1: Allgemeine Modelltheorie [Tho02, S. 28]

Modellbildung

Die Modellbildung kann in zwei verschiedene Verfahren unterteilt werden. Diese sind abhängig von der zur Verfügung stehenden Informationen über die Struktur des Systems. Beim sogenannten „Whitebox-Text“ hat der Modellierer Kenntnis von der inneren Struktur des (realen) Systems und kann auf dieser Basis ein Modell anfertigen. Sollte nur das Verhalten des Systems oder die Interaktion mit dem System als Informationsquelle zur Verfügung stehen, spricht man von einem „Blackbox-Test“.
[Inf]

Allgemein lässt sich der Prozess der Modellbildung in die Phasen

- Abgrenzung
- Reduktion
- Dekomposition
- Aggregation
- Abstraktion

unterteilen. [Inf]

Abgrenzung

In der ersten Phase der Modellbildung werden alle Informationen und Objekte identifiziert, welche für das Modell nicht erforderlich sind.
[Sch13]

Reduktion

Im zweiten Schritt werden Details der verbleibenden Objekte entfernt.
[Sch13]

Dekomposition

Die verbliebenen Objekte werden in Komponenten aufgeteilt.
[Sch13]

Aggregation

Die einzelnen Komponenten werden im vorletzten Schritt der Modellbildung in sinnvolle Module zusammengefasst.
[Sch13]

Abstraktion

Die Module aus der Aggregationsphase werden in Klassen dargestellt.
[Sch13]

Besonderheiten eines Sicherheitsmodell

Sicherheitsmodelle werden für den Zweck geschaffen, ein Modell zu erstellen, dass eine Sicherheitsperspektive auf das Original besitzt. Bei der Modellbildung in den Phasen *Abgrenzung* und *Reduktion* wird der Fokus auf sicherheitsrelevante Aspekte gelegt.

2.3 Smartphone

Dieses Kapitel liefert eine Definition des Smartphones und führt die gängigsten Hardware-Schnittstellen auf.

Definition

Die Definition des Smartphones ist erforderlich, um eine Abgrenzung zu anderen mobilen Endgeräte, wie klassische Mobiltelefone, PDAs oder E-Reader zu ermöglichen.

> *„Mobiltelefon mit erweitertem Funktionsumfang. Dazu zählen neben der Telefonie und Short Message Service (SMS) üblicherweise Zusatzdienste wie Electronic Mail (E-Mail), World Wide Web (WWW), Terminkalender, Navigation sowie Aufnahme und Wiedergabe audiovisueller Inhalte. Auf Smartphones laufen gegenüber herkömmlichen Mobiltelefonen komplexere Betriebssysteme wie etwa Symbian OS, Blackberry OS oder das iPhone OS. Die hierdurch geschaffene Möglichkeit zur Installation weiterer Applikationen durch den Endnutzer verleiht Smartphones einen erweiterbaren und individualisierbaren Funktionsumfang.“[Sju15]*

Schnittstellen

Dieser Abschnitt listet alle gängigen physikalischen und drahtlosen Schnittstellen von Smartphones und deren aktuellen Standards auf. Die Verwendung aktueller Standards ist für die Sicherheit von Smartphones bedeutend, da Schwächen im Standard nicht so einfach behoben werden können.

WLAN / IEEE 802.11

Wireless Local Area Network ist die wohl bekannteste und verbreitetste Technologie zur drahtlosen Kommunikation in Computernetzen. Es sind zwei unterschiedliche Kommunikationsarten mit anderen Teilnehmern möglich. Im Adhoc-Modus kommunizieren die Endgeräte direkt miteinander. Ein Gerät kann dabei Verbindungen zu mehreren Geräten etablieren. Im Infrastruktur-Modus erfolgt die Kommunikation mit einer Basisstation, auch als Access Point bekannt. Der Standard ist in IEEE 802.11 beschrieben. Der jüngste Standard ist 802.11ah [Sut16] (Stand Mai 2017). Für die Datenübertragung werden je nach Standard unterschiedliche Modulationsverfahren verwendet.
[Bra15]

Sicherheit

Es existieren folgende Verschlüsselungsverfahren für die Übertragung:

- WEP: Wired Equivalent Privacy
- WPA: Wi-Fi Protetcted Access
- WPA2: Wi-Fi Protected Access 2

Die Algorithmen verwenden unterschiedliche Verfahren zur Verschlüsselung von Daten. Seit 2017 gelten die Verfahren WEP[1] und WPA als nicht mehr sicher[2] und sollten deswegen nicht mehr verwendet werden.
[Sicg]

Bluetooth / IEEE 802.15.1

Mittels Bluetooth kann eine Datenübertragung auf kurze Entfernungen realisiert werden. Durch die drahtlose Übertragung sollen kurze Kabelverbindungen vermieden werden. Die Kommunikation mit anderen Teilnehmern kann verbindungslos oder verbindungsorientiert sein. Bluetooth ist als Standard in der IEEE 802.15.1 festgelegt. [Bra15] Die aktuellste Version (Stand Mai 2017) ist Bluetooth 5.
[Živ17]

Mobilfunk

Die Ära des Mobilfunks begann bereits Mitte der 1950er Jahre mit der Einführung des „A-Netzes“, welches von der deutschen Bundespost betrieben wurde. In den nächsten Jahrzehnten wurde es durch das B- und C-Netz ersetzt. Einen globalen Standard erreichte mobile Kommunikation im Jahre 1992 durch die Einführung des *Global System for Mobile Communications*.
[Sau15]

GSM

Das *Global System for Mobile Communications (GSM)* ist ein Standard, der eine leitungsvermittelnde Datenübertragung zur Kommunikation nutzt. Der Verbindungsaufbau erfolgt durch das Signalisierungssystem Nr. 7 (kurz: SS7). Dieses Protokoll ist bis heute der international am weitesten verbreitete Standard für Verbindungsaufbau, -erhalt und -abbau von Mobilkommunikation. [Sau15] Das Protokoll, welches 1975 entwickelt wurde, enthält jedoch mehrere Schwachstellen, die Stand 2017 noch zu Angriffen[3] benutzt werden.
[Mou15]

GPRS

Das Protokoll *General Packet Radio Service (GPRS)* stellt eine Erweiterung des GSM-Standards um eine Datenverbindung dar. Der Nachfolger von GPRS ist das Protokoll *Enhanced Data Rates for GSM Evolution (EDGE)*, welches die Datenrate weiter erhöhte.
[Sau15]

UMTS

Das *Universal Mobile Telecommunications System (UMTS)* ist eine Weiterentwicklung von GSM und GPRS, das die Anforderungen an Sprach- und Datenübertragung in einem Protokoll vereint. Das Protokoll *High Speed Packet Access (HSPA)* stellt eine Weiterentwicklung von UMTS dar und ermöglicht weitere Geschwindigkeitssteigerungen bei der Übertragung von Daten.
[Sau15]

[1] Bereits 2005 konnte WEP innerhalb weniger Minuten kompromittiert werden: http://heise.de/-294376

[2] 2008 demonstrierten Sicherheitsforscher einen erfolgreichen Angriff auf WPA innerhalb weniger Minuten: https://heise.de/-216135

[3] SS7 iVm. UMTS: https://www.heise.de/newsticker/meldung/Deutsche-Bankkonten-ueber-UMTS-Sicherheitsluecken-ausgeraeumt-3702194.html

LTE

Long Term Evolution ist der aktuelle Standard für mobile Kommunikation. LTE setzt ähnlich wie im Festnetz bereits fast abgeschlossen auf eine Übertragung mittels Paketvermittlung unter Nutzung des Internet-Protokoll (IP). Dies betrifft die Sprachtelefonie und Datenkommunikation gleichermaßen. Einzige Ausnahme stellt die SMS dar, welche aus Architekturgründen weiterhin über Signalisierungsnachrichten übertragen wird.
[Sau15]

NFC

Near Field Communication (NFC) ist ein Übertragungsstandard, der kontaktlos mittels elektromagnetischer Induktion erfolgt. Im Gegensatz zu klassischen RFID-Systemen, in denen das Lesegerät die Kommunikation initiiert, können bei NFC sowohl Transponder als auch das Lesegerät einen Nachrichtenaustausch initiieren.
[LR10]

GPS

Das *Global Positioning System (GPS)* dessen offizieller Name *Navigation Satellite Timing and Ranging Global Positioning System (NAVSTAR GPS)* lautet, wurde ursprünglich für die militärische Nutzung entwickelt. Die Funktionsweise von GPS basiert auf einer permanenten Mitteilung der aktuellen Position und Zeit jedes Satelliten. Diese Informationen werden auf zwei Frequenzen übertragen. Zur genauen Berechnung benötigt der Empfänger mindestens drei (theoretisch) Signale von unterschiedlichen Satelliten.
[Sch14]

USB

Universal Serial Bus (USB) ist ein Protokoll zur Verbindung von externen Geräten an einen Computer. Der Standard wurde in Version 1.0 1995 veröffentlicht und hatte das Ziel, die zum damaligen Zeitpunkt herrschende Vielfalt an Anschlüssen zur Nutzung von externen Geräten zu vereinheitlichen. Die aktuellste Version trägt offiziell den Namen „USB 3.1 Gen 1“. [USB]
Zahlreiche Android-Geräte nutzen diesen Standard um eine Akkuladung zu ermöglichen und eine Verbindung zwischen Computer und Smartphone herzustellen.

Lightning

Lightning ist ein von Apple entwickelter proprietärer Anschluss für Geräte des gleichnamigen Herstellers. Der Anschluss wird für die Verbindung mit iTunes auf einem Computer oder zum Ladevorgang benötigt.
[Incm]

JTAG

Ein *Joint Test Action Group (JTAG)* Anschluss bietet Entwicklern von Embedded Geräten die Möglichkeit diese Geräte zu testen und debuggen. Durch Weiterentwicklungen kann JTAG auch zur Konfiguration von FPGAs genutzt werden. Außerdem ist ein Zugriff auf den Flashspeicher möglich. [Das+13] In der IT-Forensik wird diese Methode zur Auswertung der auf dem Gerät gespeicherten Nutzerdaten verwendet, da hierfür kein Booten des Gerätes und damit eine eventuelle Modifikation des Systems erforderlich ist.
[AK16]

2.4 Betriebssystem

Dieser Abschnitt stellt die grundlegende Funktionsweise eines Betriebssystems vor und erläutert die Besonderheiten bei gängigen mobilen Betriebssystemen. Um die grundlegenden Aufgaben und Funktionsweisen eines Betriebssystems zu verstehen, zeigt die Abbildung 2.2 in welchen Kontext ein Betriebssystem innerhalb eines Rechners eingeordnet wird.

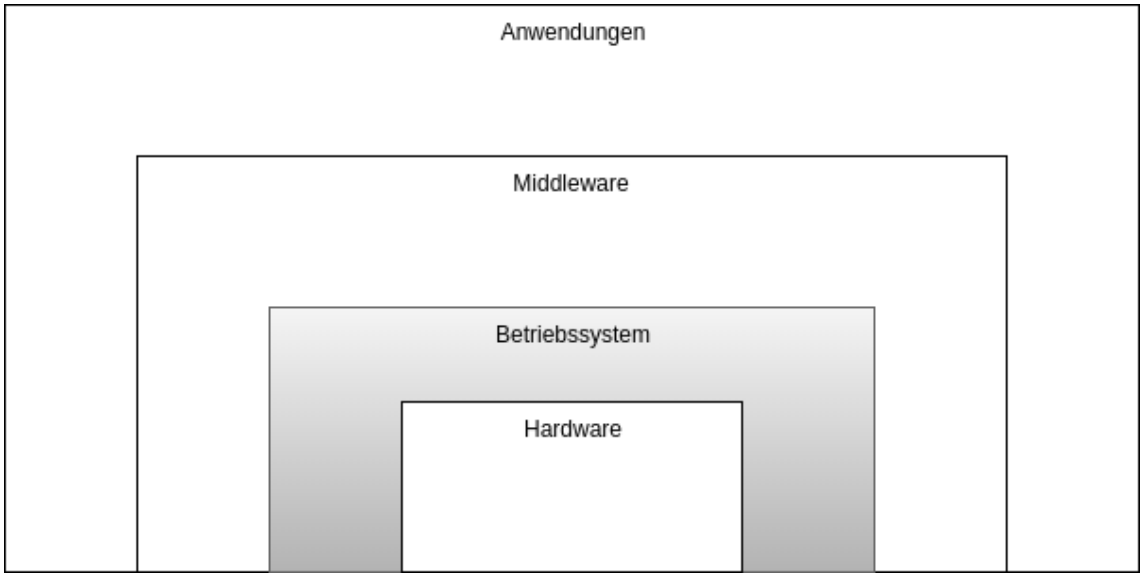

Abbildung 2.2: Positionierung eines klassischen Betriebssystems [Kao, S. 13]

Das Betriebssystem mit seinem Kernel (Betriebssystemkern) stellt eine Art Mittler zwischen Anwendung und Hardware dar. Zu den Hauptaufgaben eines Betriebssystems zählen:

Speicherverwaltung

Die Verwaltung der Speicheradressen von Prozessen auf dem Hauptspeicher wird durch das Betriebssystem erledigt.[Bra17]

Dateisystem

Die Organisation und das Management von Dateien wird in einem Dateisystem festgelegt und ermöglicht Anwendungen, Daten persistent auf der Festplatte abzuspeichern.[Bra17]

Systemaufrufe

Die von einem Betriebssystem angebotenen Systemaufrufe stellen eine Schnittstelle für unprivilegierte Prozesse dar, welche privilegierte Aufgaben erfüllen müssen, beispielsweise das Lesen einer Datei auf der Festplatte.[Bra17]

Prozessverwaltung

In der Prozessverwaltung wird die zeitliche Zuteilung von Hardwareressourcen realisiert.[Bra17]

Interprozesskommunikation

Die Interprozesskommunikation ermöglicht den Austausch von Informationen zwischen Prozessen. Hierzu stehen je nach Betriebssystem unterschiedliche Schnittstellen zur Verfügung.

2.5 Android

Dieses Kapitel liefert einen Einblick in die Architektur von Android und ist unterteilt in Erklärungen über den *Linux Kernel*, den *Hardware Abstraction Layer*, dem *Anwendungsframework (Java API Framework)* und den vorhandenen *Sicherheitsmechanismen* dieses Betriebssystem.

Allgemeines

Android ist ein auf dem Linux Kernel basierendes Open Source Betriebssystem, welches von Google und der Open Handset Alliance entwickelt wird.[IAj]

Die Abbildung 2.3 zeigt hierarchisch den Aufbau der Android Plattform. Die wesentlichen Komponenten werden in den nächsten Abschnitten erklärt.

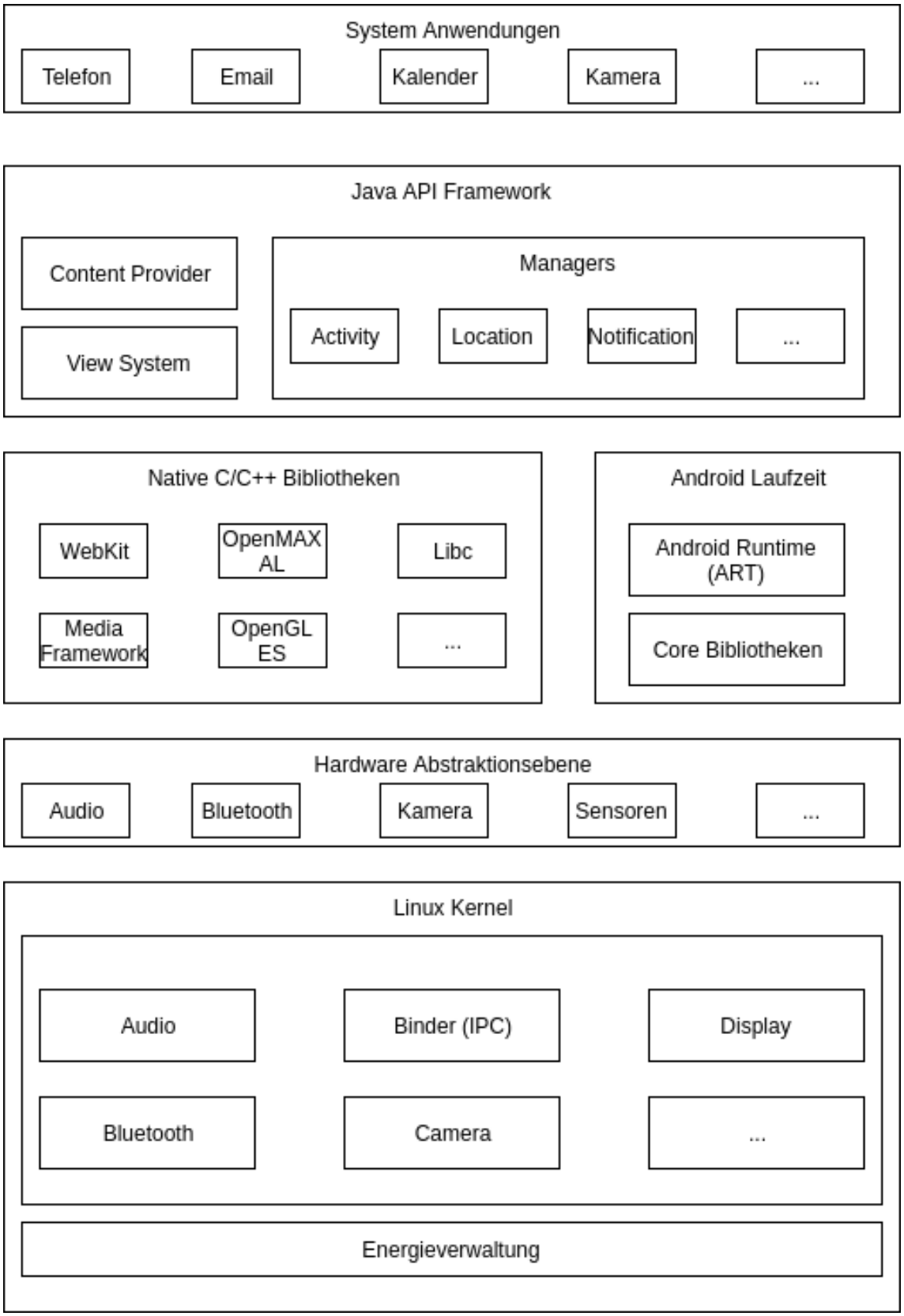

Abbildung 2.3: Android Stack [IAj]

2.5.1 Linux Kernel

Android verwendet als Kernel für sein Betriebssystem den Linux Kernel. Als Gründe für die Wahl dieses Kernel geben die Entwickler Vorteile durch die bereits vorhandenen Sicherheitseigenschaften und die Möglichkeit für Gerätehersteller Hardware-Treiber für einen verbreiteten Kernel zu schreiben, an[IAj]. Als Schlüsselfeatures hinsichtlich der Sicherheit bietet der Kernel:

- Benutzer basiertes Rechtemanagement
- Isolation von Prozessen
- Mechanismen für eine sichere Interprozesskommunikation
- Möglichkeit unsichere und nicht erforderliche Teile des Kernels zu entfernen

[IAo]

Durch das benutzerbasierte Rechtemanagement ist es möglich, dass ein Nutzer keinen Zugriff auf die Dateien, Arbeitsspeicher, CPU Ressourcen und die Schnittstellen eines anderen Nutzers erhält. Android erweitert dieses Rechtemanagement, indem jede Anwendung einem eigenen Nutzer zugewiesen und in einem separaten Prozess ausgeführt wird. Hierdurch ist es einer Anwendung nicht möglich, mit einer anderen Anwendung direkt zu kommunizieren[4]. Dieses Verfahren wird auch als „Application Sandboxing“ bezeichnet. Durch bösartige oder ungewollte Speicherfehler (z.B. Buffer Overflow) kann die Sicherheit des gesamten Systems (Gerät) beeinträchtigt werden. Da Android das Sandboxing auf Betriebssystemebene durchführt, kann eine Kompromittierung des Gerätes nur durch Schwachstellen im Linux Kernel erreicht werden.

Um die Sicherheit des Betriebssystems zu erhöhen, sind auf Kernel-Ebene weitere Schutzmechanismen vorgesehen:

- Systemdaten werden als Read-Only Partition eingebunden (mount)
- Dateisystemaufrufe auf Dateien außerhalb des Applikationskontextes sind nur durch explizite Freigaben möglich
- Security-Enhanced Linux (SELinux) wird verwendet
- Der Bootvorgang wird verifiziert (seit Android 6.0)
- Kryptographische Schnittstellen sind vorhanden
- Standardmäßig ist nur der Kernel und eine Reihe von Systemanwendungen mit Root-Rechten ausgestattet
- Verschlüsselung von Daten ist möglich
- Zugriffsschutz durch Passwortabfragen
- Geräteadministration ist durch native Schnittstelle möglich

[IAo]

[4] Um diese Einschränkung zu eliminieren, bietet Android mehrere Verfahren der Interprozesskommunikation.

2.5.2 Hardware Abstraction Layer (HAL)

Die Hardwareabstraktionsschicht stellt eine Art Vermittler zwischen Hardware und dem restlichen System dar. Damit ist es einfacher das Betriebssystem auf unterschiedliche Geräte zu portieren. Android definiert zur Vereinheitlichung Standardschnittstellen für Hardwaremodule:

- Audio
- Automotive
- Bluetooth
- Kamera
- DRM
- Graphiken
- Eingabe
- Medienwiedergabe
- Peripheriegeräte
- Sensoren
- Speicher
- Television

Die Verwendung von HAL Interfaces ist optional und nicht verpflichtend.
[IAa]

2.5.3 Java API Framework

Das Anwendungsframework bietet diverse Schnittstellen für Entwickler von Applikationen. Das Framework ist in Provider und Manager gruppiert, welche Schnittstellen für eine Funktionalität zur Verfügung stellen. Die wichtigsten und am häufigsten genutzten sind[5][IAa]:

- *Activity Manager:* Zuständig für den Aktivitäts-Lebenszyklus einer Anwendung (Erstellung, Start, Unterbrechung, Fortsetzung, Stopp, Löschen).
- *Content Provider:* Ermöglicht einer Anwendung auf Daten anderer Anwendungen zuzugreifen oder zu veröffentlichen.
- *Notifications Manager:* Ermöglicht Anwendungen Benachrichtigungen und Mitteilungen anzuzeigen.
- *View System:* Schnittstelle zur Erstellung des Frontends einer Anwendung.

In diesem Abschnitt werden die angrenzenden Schichten (siehe Abbildung 2.3), welche direkt oder indirekt durch das *Application Framework* genutzt werden, erklärt.

Binder IPC

Um eine Kommunikation mit anderen Prozessen zu ermöglichen, nutzt Android eine Implementierung des Binder-Frameworks, da der Linux Kernel nur Primitives wie Pipes, Shared Memory oder Sockets anbietet [Eve15]. Auf Anwendungsebene ist die Interprozesskommunikation durch die Schnittstellen Intents, Content Provider oder Broadcast Receiver sichtbar.
[Eve15] [IAa]

[5] Die Dokumentation aller zur Verfügung stehenden Packages kann unter https://developer.android.com/reference/packages.html abgerufen werden.

Systemdienste

Systemdienste stellen modularisierte kontextbezogene Schnittstellen zur Verfügung. Sie sind in die Bereiche *System* (z.B. Benachrichtigungsmanager) und *Medien* (z.B. Mediaplayer) unterteilt.
[IAa]

Native C/C++ Libraries

Manche Komponenten des Androidsystems erfordern nativen Code, der in C/C++ geschrieben wurde (z.B. OpenGL). Für die Entwicklung von Anwendungen ist die API des Java Frameworks vorgesehen. Einige dieser Komponenten werden durch das Android Framework als Schnittstellen angeboten (z.B. Java OpenGL API). Für den Fall, dass keine Schnittstelle vorhanden oder aus anderen Gründen, Code in C oder C++ implementiert werden soll, bietet das Android NDK (Native Development Kit) die Möglichkeit direkt C/C++ Code zu verwenden.
[IAa]

System Apps

Das Betriebssystem Android wird mit einer Reihe an Systemanwendungen ausgeliefert. Es bietet Dienste im Bereich E-Mail, SMS, Kalender, Kontakte und weitere häufig genutzte Anwendungen. Bis auf wenige Ausnahmen hat der Benutzer die Möglichkeit diese Standardanwendungen durch alternative Applikationen zu ersetzen. Des Weiteren besitzen diese Systemapplikationen Nutzungsmöglichkeiten für Entwickler, beispielsweise um Zugriff auf neue SMS-Nachrichten zu erhalten. [IAa]

Android Runtime

Der von Java compilierte Code wird - im Gegensatz zu anderen Programmiersprachen - in einen Bytecode übersetzt, der zur Ausführung einen Interpreter benötigt. Unter Android existierte bis Version 5.0 ausschließlich Dalvik als Laufzeitumgebung. Mit Version 5.0 wurde die Android Runtime (ART) als neue Laufzeitumgebung eingeführt. Sie unterstützt sowohl eine Ausführung in „Ahead-of-time (AOT)"[6] oder „Just in Time (JIT)" [7]. Zusätzlich wurde der Garbage Collector verbessert und die Debuggingmöglichkeit erweitert. [IAa]

2.5.4 Sicherheit

Dieser Abschnitt führt den Leser in die grundlegenden Sicherheitsmechanismen des Betriebssystems Android ein. Bereits im Abschnitt *Linux Kernel* wurde auf die grundlegenden Maßnahmen zur Sicherheit auf Kernel-Ebene eingegangen. Eine detailliertere und umfangreichere Bewertung erfolgt im Sicherheitsmodell (Kapitel 2.4) dieser Arbeit.

Verschlüsselung

Die (symmetrische) Komplettverschlüsselung von Benutzerdaten wurde mit Android Version 4.4 eingeführt. In Version 5.0 wurden zusätzlich Features, wie „fast encryption" eingeführt. Zusätzlich wird der Verschlüsselungsschlüssel in einem sicheren Hardwarespeicher mittels einer Trusted Execution Environment (TEE) aufbewahrt [IAf]. Android Nougat (7.0) unterstützt eine dateibasierte Verschlüsselung der Daten. Dies bedeutet, dass unterschiedliche Schlüssel zur Verschlüsselung der Dateien verwendet werden. [IAe]

[6] Ahead of time bedeutet, dass der Programmcode in Maschinencode bereits vor der Ausführung übersetzt wird.

[7] Just in time bedeutet, dass der Programmcode erst zur Laufzeit in Maschinencode übersetzt wird.

Bootvorgang

Ab Version 4.4 wurde ein verifizierter Bootvorgang eingeführt. Beim Starten des Gerätes initiiert der Bootloader (sofern vorhanden an die TEE) die Integritätsüberprüfung der „boot“ und „recovery“ Partition. Die Überprüfung wird mit einem speziellen Kerneltreiber „dm-verity“ mittels kryptographischen Hashfunktionen blockweise durchgeführt. Anschließend wird der errechnete Wurzelhashwert mit dem vom Gerätehersteller erstellten RSA Schlüssel (OEM Schlüssel) verifiziert, welcher im Speicher des TEE aufbewahrt werden sollte. Android bietet zusätzlich die Möglichkeit weitere Schlüssel zur Integritätsüberprüfung zu verwenden. Je nach Ausgang dieser Überprüfung wird der Benutzer mit einem Hinweis gewarnt oder der Bootvorgang schlägt fehl und das Gerät schaltet sich ab. Zur Information des Anwenders nutzt Android ein Ampelsystem in den Farben Grün, Gelb und Rot. Sollte durch den Gerätehersteller eine Implementierung vorliegen, welche den Gerätestatus (eingeschränkt oder „gerooted“) einbezieht, wird zusätzlich die Warnkategorie Orange eingeführt, die dem Nutzer signalisieren soll, dass die Integrität der Gerätesoftware nicht überprüft werden kann, da der Bootloader entsperrt wurde. Der Status Grün wird angezeigt, wenn die Überprüfung der beiden Partitionen mit dem OEM Schlüssel erfolgreich ist. Sollte die Überprüfung mit dem OEM Schlüssel nicht erfolgreich sein, kann die Überprüfung mit einem weiteren angebotenen Schlüssel - sofern vorhanden - durchgeführt werden. Der Nutzer muss diesem Bootvorgang zustimmen. Sollte keine dieser Validierungen erfolgreich sein, erhält das Gerät den Status Rot und der Nutzer wird durch eine Nachricht informiert, anschließend fährt das Gerät selbständig herunter.[IAq]

Mit der Veröffentlichung von Version 7.0 wurde die Verifizierung des Bootvorgangs verpflichtend eingeführt. Um kleinere Fehler in den Partitionen zu beheben, wurden Fehlerkorrekturfunktionen implementiert, die transparent im Hintergrund ausgeführt werden.[Tol]

Authentifizierung

Nachdem der Bootvorgang erfolgreich durchlaufen ist, bietet Android zur Benutzerauthentifizierung zwei Dienste. Gatekeeper ermöglicht eine Authentifizierung mittels PIN, Muster oder eines Passwortes. Der Fingerprint-Service erfordert einen entsprechenden Hardware-Sensor und nutzt die Merkmale eines Fingerabdruckes zur Authentifizierung des Nutzers. Nach erfolgreicher Authentifizierung wird im Trusted Execution Environment ein sogenannter „AuthToken“ erstellt und an das Android Betriebssystem übermittelt. Der AuthToken enthält die User SID[8], welche mit dem HMAC Schlüssel signiert wird. Der HMAC Schlüssel darf außerhalb der vertrauenswürdigen Zone (TEE) nicht verfügbar sein. [IAd]

Trusted Execution Environment

Trusty OS ist ein eigenes Betriebssystem, welches in einer Trusted Execution Environment ausgeführt wird. Eine Trusted Execution Environment wird verwendet, um Prozesse mit einem erhöhten Sicherheitsbedarf (z.B. Fingerabdruck API, Digital Rights Management (DRM) Software, elektronische Identitäten) in einem sicheren Kontext auszuführen. Dieser Zustand wird durch einen zusätzlichen und vom restlichen System isolierten Mikroprozessor

[8] Die User (*S*ecure *ID*entifier) SID wird beim initialen Festlegen der geheimen Information (PIN, Passwort, Muster) erzeugt.

erreicht. Damit soll der Zugriff auf sensible Daten (z.B. Kennwörter, Informationen zu Fingerabdruck, Zertifikate) über Speicherfehler oder andere Angriffsvektoren vermieden werden. Auf die gespeicherten Informationen kann nur die vertrauenswürdige Zone zugreifen. Um mit den entsprechenden Prozessen im vertrauenswürdigen Bereich vom Hauptprozessor aus kommunizieren zu können, sind Schnittstellen definiert. [IAp]

Applikationssicherheit

Um die Sicherheit von Applikationen zu erhöhen, bietet Android im Manifest zentral die Möglichkeit den Zugriff auf Aktivitäten, Dienste, Empfänger und ContentProvider zu autorisieren. Der folgende Abschnitt führt in die wesentlichen Sicherheitsmechanismen für mobile Anwendungen unter Android ein.

Interprozesskommunikation (IPC)

Interprozesskommunikation ist erforderlich, da unter Android jede Anwendung in einer Sandbox isoliert wird. Um eine derartige Kommunikation durchzuführen, bietet das Betriebssystem mehrere Möglichkeiten:

- Binder: leichtgewichtiger RPC, der als Linux Treiber implementiert ist.
- Service: Dienste können mittels Binder direkt genutzt werden.
- Intents: Möglichkeit über eine Art Nachrichtensystem zu kommunizieren (z.B. Aufruf eines Bildes aus Applikation heraus durch Übergabe der Anwendung, welche das ausgewählte Bild anzeigen soll).[9]
- ContentProvider: Möglichkeit eine bestimmte Sicht auf Daten des Gerätes zu erhalten (z.B. ein ContentProvider, der Kontaktbuch enthält).

[IAb]

Keystore

Android bietet nativ die Möglichkeit Schlüssel zum Verschlüsseln von sensiblen Informationen in einem sicheren Schlüsselspeicher aufzubewahren. Damit besteht für Entwickler die Option, sensible Daten durch eine zusätzliche Verschlüsselung von Unbefugten zu schützen. Alternativ besteht softwareseitig mittels Bibliotheken (z.B. Bouncy Castle) die Möglichkeit geheime Informationen (z.B. Passwörter) verschlüsselt abzuspeichern[CRP14]. In Version 6.0 wurde ein *Keymaster Hardware Abstraction Layer (v1.0)* definiert, um geheime Schlüssel in einem sicheren Hardwarebereich abzuspeichern und nur in dem Trusted Execution Environment auszuführen.[10]
[IAh; IAg]

[9] Beispiel-Aufruf eines Intents: fb://profile?id=15429 (Aufruf des Facebook Profils des Nutzers mit der ID: 15429)

[10] Diese Option steht nur Geräten zur Verfügung, welche hardwareseitig über die erforderlichen Komponenten verfügen.

Signieren von Anwendungen

Alle Anwendungen im Google Play Store müssen eine Codesignierung enthalten, die den Urheber einer Anwendung identifizieren. Seit Version 4.2 wird jede Anwendung zusätzlich vor Veröffentlichung im Play Store einer Überprüfung unterzogen. Vor der Installation einer Anwendung auf einem Gerät, überprüft der Package Manager erneut, ob eine gültige Signatur vorliegt. Dies soll verhindern, dass bösartige Anwendungen unter Umgehung des Play Stores installiert werden können.
[IAb; IAc]

Einführungen in Versionen

Die nachfolgende Aufzählung listet die Einführung von Sicherheitsmechanismen nach Produktversionen auf:

Android 6.0

- Applikationsberechtigungen zur Laufzeit
- Verifizierter Bootvorgang
- Neuer Hardware Abstraction Layer
- Authentifizierung mit Fingerabdruck
- Entfernbare Datenträger können als Speichererweiterung transparent eingebunden werden
- Weiterentwicklung von SELinux
- USB Zugriffskontrolle
- Kommunikationsart festlegen (z.B. HTTPs only)

Android 7.0

- Dateibasierte Verschlüsselung
- Direct Boot (ausgewählte Anwendungen werden bereits ausgeführt, obwohl Gerät noch nicht vollständig entschlüsselt ist)
- Verpflichtender verifizierter Bootvorgang
- Weiterentwicklung von SELinux
- Verbessertes Address Space Layout Randomization (ASLR)
- Weitere Kernelhärtung
- Applikationssignierung in Version 2.0
- (zusätzliche) Konfigurationsmöglichkeiten bei der Netzwerksicherheit

[IAm; IAl]

2.6 iOS

Dieses Kapitel liefert einen Einblick in die Funktionsweise und Architektur von Apples Betriebssystem iOS.

Allgemeines

iOS ist ein von Apple entwickeltes proprietäres Betriebssystem, welches auf den Darwin Kernel basiert. Die Abbildung 2.4 zeigt den Stack des Betriebssystems.

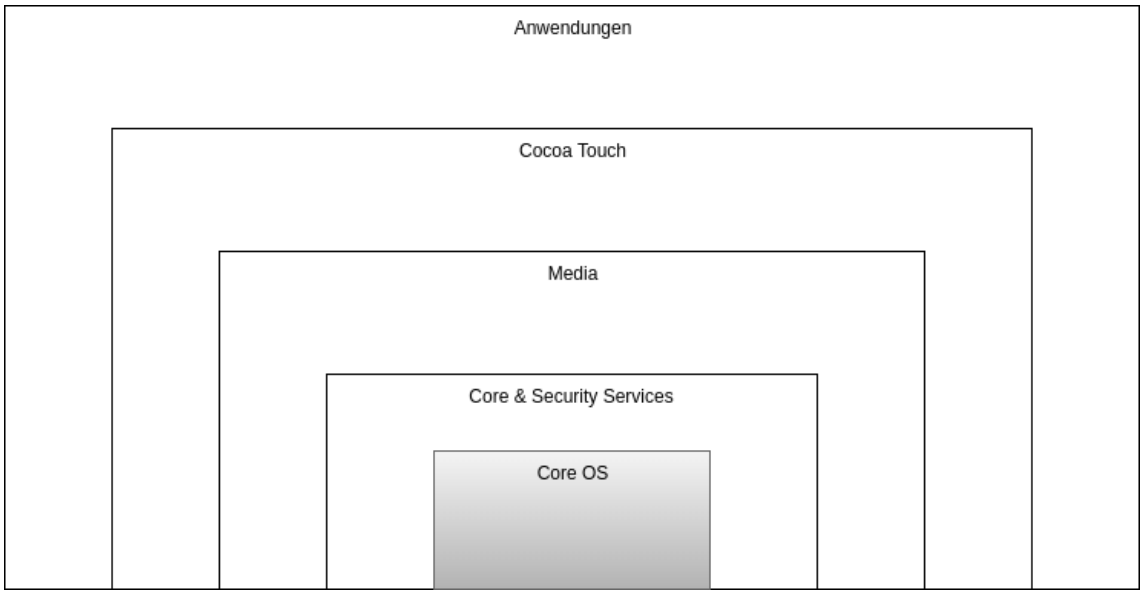

Abbildung 2.4: Architektur von iOS [Spr17]

2.6.1 Kernel (Core OS)

Das (Core) Betriebssystem von iOS basiert auf dem ebenfalls von Apple entwickelten Betriebssystem Darwin, dessen Basis UNIX ist. Als Kernel wird der Mach3-Kernel verwendet. Das Betriebssystem übernimmt als unterste Schicht die Interaktion mit der Hardware. In iOS 7 wurde die bis dato verwendet 32-Bit Architektur durch eine 64-Bit Architektur ersetzt. Alle Systembibliotheken und Schnittstellen unterstützen eine 64-Bit Umgebung, jedoch werden existierende 32-Bit Applikationen weiterhin unterstützt. Apple empfiehlt allen Entwicklern, künftige Applikationen in einer 64-Bit Umgebung zu entwickeln.
[Spr17; Inci]

Core & Security Services

In dieser Ebene befinden sich alle Systembibliotheken, die vom System und von Applikationen gleichermaßen verwendet werden. Exemplarisch sind dies Schnittstellen um mit der iCloud zu interagieren oder um eine Netzwerkkommunikation zu ermöglichen. Zusätzlich sind auf diesem Layer zahlreiche Sicherheitsmechanismen implementiert.
[Spr17; Incj]

2.6.2 Media Layer

Die Media Schicht verwaltet alle Bibliotheken zur Nutzung von Mediendateien (Audio, Video, Bilder). Es sind Bibliotheken zur Bearbeitung derartiger Mediendateien verfügbar, als auch Module zur Wiedergabe und Streaming von Inhalten auf dem Gerät oder anderen kompatiblen Geräten im Netzwerk.
[Spr17, S. 12] [Incb]

2.6.3 Cocoa Touch Layer

Als oberste Softwareschicht von iOS bietet der Cocoa Touch Layer Entwicklern Schnittstellen zu darunter liegenden Bibliotheken an. Der Layer ist unterteilt in das Foundation-Framework, welches Basisklassen (z.B. Arrays, Strings) zur Verfügung stellt und dem

UIKit, das aus einem Benutzerinterface und einem Datenmodell besteht. Ein Teil der Frameworks basiert auf der Cocoa API des Mac OS X Betriebssystems für Notebooks und Personal Computer.
[Spr17; Incg]

2.6.4 Sicherheit

Die Grafik in Abbildung 2.5 zeigt das Zusammenspiel der Sicherheitsmechanismen auf den unterschiedlichen Schichten, um Daten auf dem Gerät vor unberechtigtem Zugriff zu schützen. Der folgende Abschnitt soll dem Leser einen Überblick der in iOS wesentlichen Sicherheitskomponenten vermitteln.

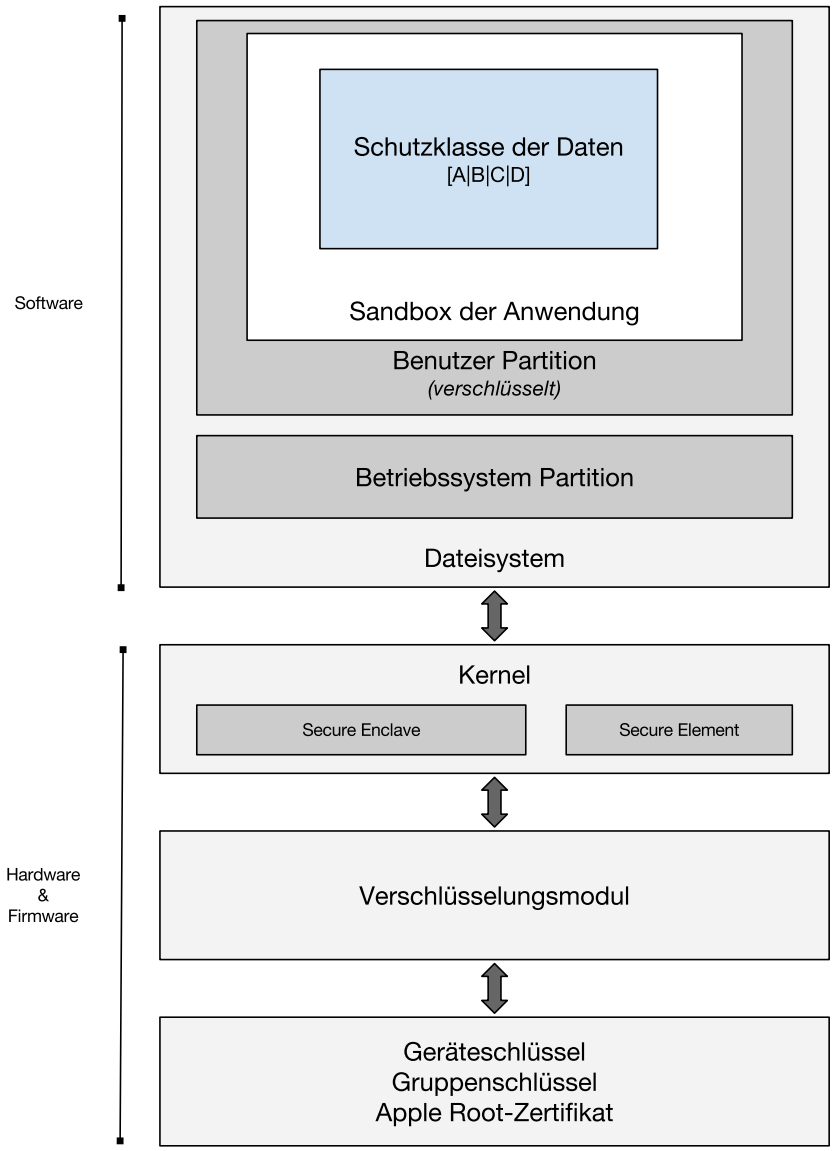

Abbildung 2.5: Sicherheits-Architektur von iOS [Incm]

In den nächsten Paragraphen werden die Kernelemente der Sicherheitsmechanismen von iOS erklärt.

Verschlüsselung

Um die Verschlüsselung der Daten schnell und effizient zu realisieren, verwendet jedes iOS-Gerät ein dediziertes AES 256 Verschlüsselungsmodul, der zwischen Flash-Speicher und Hauptspeicher als Mittler die Kryptographie übernimmt. Alle Daten werden zusätzlich mit dem AES Schlüssel der eindeutigen ID (UID) des Gerätes verschlüsselt. Damit soll verhindert werden, dass ein Datenzugriff erfolgt, indem der Speicher ausgebaut und in einem anderen Geräte eingebaut wird. [Incm]

Gerätecodes

Sobald ein Nutzer einen Gerätecode konfiguriert, wird die Datensicherheit aktiviert. Zur Sicherung kann der Nutzer, sechs- oder vierstellige Ziffern verwenden, alternativ einen alphanumerischen Schlüssel beliebiger Länge. Bei Erzeugung des Gerätecodes wird dieser an die UID des Gerätes gebunden, wodurch ein Brute-Force-Angriff exklusiv auf diesem Gerät erzwungen wird. Bei Falscheingabe wird eine zeitliche Verzögerung zur erneuten Eingabe des Gerätecodes, abhängig von den bisherigen Versuchen, etabliert, welche auch bei einem Neustart des Gerätes erhalten bleibt. [Incm]

Hardwareseitige Verschlüsselung

Um die Verschlüsselung der Daten schnell und effizient zu realisieren, verwendet jedes iOS-Gerät ein dediziertes AES 256 Verschlüsselungsmodul, das als Mittler zwischen Flash-Speicher und Hauptspeicher die kryptographischen Operationen übernimmt. Alle Daten werden zusätzlich mit dem AES Schlüssel der eindeutigen ID (UID) des Gerätes verschlüsselt. Damit soll verhindert werden, dass ein Datenzugriff erfolgt, indem der Speicher ausgebaut und in einem anderen Gerät eingebaut wird.
[Incm]

Dateiverschlüsselung

Zusätzlich zur standardmäßig eingebauten Hardwareverschlüsselung verwenden iOS-Geräte für die Sicherheit der in Dateien enthaltenen Informationen, eine dateibasierte Verschlüsselung. Bei Dateierstellung auf der Partition des Flash-Speichers wird ein individueller 256-Bit Schlüssel erzeugt und an die AES-Engine übergeben. Diese nutzt den erzeugten Schlüssel zur Verschlüsselung der Daten. Jeder generierte Schlüssel wird einer (von mehreren existierenden) Klassen zugeordnet. Welche Klasse ausgewählt wird, richtet sich danach, unter welchen Bedingungen Zugriff auf diese Datei erforderlich ist. Der Dateischlüssel wird mit dem Klassenschlüssel verschlüsselt und in den Metadaten der Datei gespeichert. Die Metadaten werden zusätzlich mit einem für alle Dateien einheitlichen Dateisystemschlüssel verschlüsselt. Dieser Schlüssel wird auf dem Gerät gespeichert und wird nicht verwendet, um die Vertraulichkeit von Daten zu gewähren. Er eignet sich für das schnelle Löschen der auf dem Gerät gespeicherten Informationen, da dieser den Eintrittspunkt einer verschlüsselten Datei darstellt. [Incm] Die nachfolgende Abbildung 2.6 zeigt schematisch die Verwendung der Schlüssel.

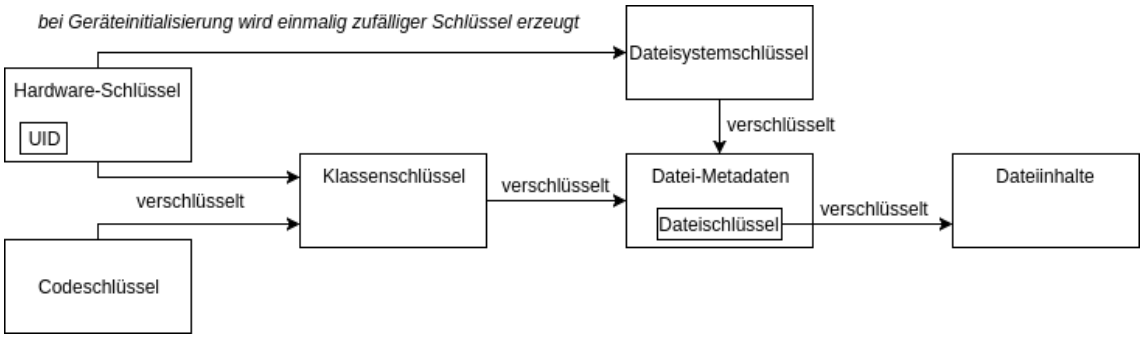

Abbildung 2.6: Verwendung der Schlüssel unter iOS [Incm]

Datensicherheitsklassen

Um sensible Informationen (z.B. Anmelde-Token) innerhalb einer Anwendung sicher speichern zu können, bietet iOS die Möglichkeit - je nach Anwendungsszenario - unterschiedlich starke Sicherheitsklassen zu verwenden:

- *Vollständiger Schutz (A):* Der Klassenschlüssel wird mit einem Schlüssel, der aus UID und Gerätecode besteht, geschützt. Je nach Einstellung wird dieser Schlüssel nach dem Sperren des Gerätes sofort oder nach einer definierten Zeit gelöscht. Nach Entsperrung des Gerätes mit dem Code oder der Touch ID, wird ein neuer Schlüssel erzeugt.
- *Geschützt, außer wenn offen (B):* Es werden temporäre Schlüssel zum Entschlüsseln des Dateischlüssels verwendet, welche für Datenspeicherungen im Hintergrund (zB. Mailanhänge) erforderlich sind.
- *Geschützt bis zur ersten Benutzerauthentifizierung (C):* Diese Klasse verhält sich wie bei einem vollständigen Schutz mit dem Unterschied, dass die erzeugten Schlüssel bei Gerätesperrung nicht vernichtet werden.
- *Kein Schutz (D):* Der Klassenschlüssel wird nur mit der UID geschützt.

[Incm]

Bootvorgang

Beim Starten des Gerätes wird vor dem Booten der auszuführende Code (Bootloader, Kernel, Kernelerweiterungen, Baseband Firmware) auf seine Datenintegrität überprüft. Hierfür werden die verwendeten Binärdateien mit dem Root-Zertifikat von Apple signiert. Der erforderliche öffentliche Schlüssel dieses Zertifikats wird im unveränderlichen Boot ROM vorgehalten. Dieses „Chain of Trust“ Prinzip wird bei allen Komponenten für den Systemstart durchgeführt. Sollte bei einem Modul die Verifizierung fehlschlagen, wird der Benutzer aufgefordert das Gerät mit iTunes zu verbinden und ein Zurücksetzen des Gerätes wird durchgeführt. [Incm]

Autorisierung von Systemsoftware

Alle Softwareaktualisierungen werden von Apple als signierte Updates ausgerollt. Dies ist erforderlich, um die Integritätsüberprüfungen während des Systemstarts erfolgreich abschließen zu können. Um zu verhindern, dass auf einem Gerät eine ältere (signierte) Softwareversion installiert wird, welche unbehobene Schwachstellen enthält, schützt der Prozess „System Software Authorization“ vor derartigen Manipulationen. [Incm]

Eindeutige Geräteidentifikation (UID) & Gerätegruppenidentifikation (GID)

Die UID ist eine eindeutige Geräte-ID, welche bei der Herstellung des Gerätes in die Secure Enclave und den Anwendungsprozessor eingebrannt werden. Die Gerätegruppen-ID wird in die beiden Prozessoren kompiliert. Beide Identifikationen bestehen aus AES-256[11] Bit Schlüssel, welche softwareseitig nicht auslesbar sind. Das AES-Modul, das hardwareseitig Zugriff auf diese IDs hat, übermittelt nur die Resultate der kryptographischen Operationen. Die Gerätegruppen-ID ist bei allen Prozessoren mit dem gleichen Typ identisch (z.B. Apple A8 CPU). Diese wird für weniger sicherheitsrelevante Aufgaben, wie z.B. die Bereitstellung von Systemsoftware verwendet. UID und GID können ebenfalls nicht über Debugging-Schnittstellen, wie beispielsweise JTAG ausgelesen werden. [Incm]

Aktivierungssperre

Bei entsprechender Konfiguration kann das Gerät ausschließlich aktiviert werden, wenn die Anmeldedaten des Eigentümers eingegeben werden. Dies soll bei Verlust oder Diebstahl des Gerätes eine Nutzung des Gerätes durch Unbefugte verhindern. In betreuten Geräten kann dieser Mechanismus ebenfalls aktiviert werden. Damit wird bewirkt, dass der Organisationsverwalter eine erneute Aktivierung durchführen kann, ohne dass die Anmeldedaten des (früheren) Mitarbeiters erforderlich sind. [Incm]

Secure Enclave

Die Secure Enclave wurde in Prozessoren der A7 Reihe erstmals eingebaut. Sie implementiert einen abgetrennten Coprozessor neben dem Hauptprozessor. Zusätzlich ist die Secure Enclave mit einem verschlüsselten Speicher versehen. Die Schlüssel, welche für die Verschlüsselung von Dateien verwendet wurden, werden in der Secure Enclave aufbewahrt und mittels kryptographischer Verfahren bereitgestellt. Damit sollen die Informationen auch im Falle eines kompromittierten Kernels sicher sein. Die Kommunikation zwischen Hauptprozessor und Coprozessor erfolgt durch Interrupts und einem gemeinsamen Speicher. Außerdem übernimmt der Coprozessor die Verarbeitung der Informationen des Fingerabdrucksensors. Die Kommunikation zwischen Sensoren und Coprozessor erfolgt über einen seriellen Bus. Die übermittelten Daten werden dabei Ende-zu-Ende verschlüsselt und sind somit nur für Dritte auf diesem Kanal nicht lesbar. [Incm]

Applikationssicherheit

Dieser Abschnitt informiert über die Hauptmechanismen zur Sicherheit von Anwendungen in diesem Betriebssystem.

[11] American Encryption Standard

Inter-App Kommunikation

Das Betriebssystem iOS bietet diverse Möglichkeiten um einen Datenaustausch mit anderen Anwendungen auf dem Gerät durchzuführen. Mittels der nativen App-Extension „AirDrop“ wird eine Schnittstelle angeboten, welche Dateien anderen Applikationen zur Verfügung stellen kann. Eine weitere Möglichkeit besteht in der Implementierung von eigenen App-Extensionen, welche individuelle Funktionalitäten mit anderen Anwendungen austauschen. Alternativ kann die Kommunikation mit anderen Anwendungen mittels definierte URL Schemata durchgeführt werden. Für einige häufig genutzte Anwendungen (http, maitlo, tel, sms) existieren durch das Betriebssystem bereits vordefinierte Schemata. Die Validierung der in einem selbstdefinierten Schema enthaltenen Daten fällt in die Verantwortung des Entwicklers. Bei Missachtung können Buffer Overflows, URL Commands, Code Insertion oder Social Engineering Angriffe durchgeführt werden. [Incq; Inck] Ein Beispielaufruf eines URL Schemata könnte folgendermaßen aussehen:

```
NSURL *myURL = [NSURL URLWithString:@"socialNetwork://www.
    socialnetwork.com?type=post?title=New␣Post?message=New␣
    Post!"];
[[UIApplication sharedApplication] openURL:myURL];
```

Transport Layer Security (TLS)

iOS unterstützt für den sicheren Transport von Daten auf Protokollebene die Standards TLS v1.0, TLS v1.1, TLSV v1.2 sowie DTLS (Datagram Transporty Layer Security) [12]. Um einen Verbindungsaufbau mit dem Zielserver zu ermöglichen, werden durch iOS Mindeststandards (z.B. TLS 1.2, Forward Secrecy) gesetzt. Diese können durch den Entwickler in der Konfiguration außer Kraft gesetzt werden.
[Incm]

Applikationssignierung

Jede Anwendung wird vor dem Startvorgang überprüft, ob sie mit einem von Apple stammenden Zertifikat signiert wurde. Sollte die Signaturüberprüfung fehlschlagen, wird die Applikation nicht ausgeführt. Diese Signaturverifikation erweitert das „Chain-of-Trust“-Konzept, welches bereits beim Bootvorgang angewendet wird, auf die Applikationsebene und verhindert ein Ausführen manipulierter oder nicht von Apple verifizierter Anwendungen.
[Incm]

Sicherheit von Laufzeitprozessen

Applikationen anderer Anbieter werden in einer Sandbox ausgeführt um zu verhindern, dass Änderungen am Gerät oder an Dateien durchgeführt werden, auf die die Anwendung keinen Zugriff hat. Die Prozesse der Applikationen anderer Anbieter werden alle, unter den Anwendungen von iOS die meisten, mit einem nicht privilegierten Nutzer „mobile“ ausgeführt. Jede Anwendung erhält ein eigenes Home-Verzeichnis für ihre Dateien. Um einen Zugriff auf Dateien außerhalb des Home-Verzeichnisses oder andere Informationen

[12] DTLS: Eine TLS ähnliche Implementierung für UDP Verbindungen

zu erhalten, bietet iOS Dienste zur Interaktion mit anderen Applikationen an. Die erforderlichen Dienste werden über ein festgelegtes Berechtigungssystem vergeben. Um sich vor Schwachstellen in der Speicheradressierung besser zu schützen, verwendet iOS ASLR (Address Space Layout Randomization), indem es beim Start einer Anwendung den Speicherbereich zufällig vergibt.
[Incm]

Touch ID

Mittels des Fingerabdrucksensors kann ein Nutzer unter iOS finanzielle Transaktionen (z.B. Apple Pay) legitimieren und eine Authentifizierung am Gerät vornehmen. Dies erleichtert dem Nutzer die Authentifizierung mit längeren Schlüsseln, da diese nicht bei jeder Anmeldung eingegeben werden müssen. In regelmäßigen Abständen oder für bestimmte Aktionen (z.B. Neustart, keine Entsperrung innerhalb der letzten 48 Stunden) erwartet das System eine händische Eingabe des Schlüssels.
[Incm]

(Sicherheits-) Zertifizierungen

Apples iOS-Geräte werden einer Reihe von Sicherheitszertifizierungen unterzogen. Namentlich sind dies:

- Cryptographic Validation nach FIPS 140-2
- Common Criteria Certification nach ISO 15408
- Commercial Solutions for Classified (CSfC)

[Incm]

In Zusammenarbeit mit Behörden und Regierungsvertretern wurden Handbücher veröffentlicht, in denen festgelegt ist, welche Einstellungen erforderlich sind, um die Gerätesicherheit zu erhöhen.
[Incm]

2.7 Weitere mobile Betriebssysteme

Dieser Abschnitt erläutert weitere mobile Betriebssysteme von namhaften Herstellern und deren aktueller Entwicklungsstand.

Blackberry OS / Blackberry 10

Blackberry 10 ist der offizielle Nachfolger von Blackberry OS und ein Betriebssystem des gleichnamigen Smartphone-Herstellers. Version 10.0 wurde am 30. Januar 2013 veröffentlicht. Die aktuellste Version ist 10.3.3. Der Fokus des Betriebssystems liegt in der Sicherheit des Gerätes und in einer höheren Produktivität im Unternehmensumfeld. Mit Veröffentlichung des Finanzberichtes für das zweite Geschäftsquartal 2017 gab Blackberry einen Strategiewechsel bekannt. Die Smartphone-Entwicklung wird eingestellt und soll künftig an externe Partner ausgelagert werden. Bereits seit 2015 wurden Blackberry-Smartphones mit einem Android Betriebssystem entwickelt und vertrieben. Das aktuellste Blackberry Smartphone „KeyOne“, dessen Hardware vom Hersteller TCL entwickelt wurde, ist ebenfalls mit Android als Betriebssystem veröffentlicht worden.
[Pin; Wal; Lim; Her]

Windows Mobile

Windows Mobile ist ein Betriebssystem, welches von Microsoft entwickelt wird. Die neueste Version von *Microsoft Windows 10 Mobile* wurde im April 2017 freigegeben. Das Betriebssystem unterstützt die ARM-Architektur. Nokia, welches zwischenzeitlich von Microsoft übernommen wurde, stellte die hierfür passenden Geräte in der Nokia Lumia Serie (später Microsoft Lumia) her. 2016 gab Microsoft bekannt, dass die Lumia-Produktion nicht fortgeführt wird.
[Cora; fin]

2.8 Open Web Application Security Project (OWASP)

Das *Open Web Application Security Project* ist eine 2001 gegründete internationale und gemeinnützige Organisation mit dem Ziel die Sicherheit von (Web-)Applikationen zu erhöhen. OWASP Projekte können in Tools, Code und Dokumente kategorisiert werden, sind quelloffen und zur freien Nutzung verfügbar.

Die vier Grundsätze der OWASP lauten:

- Transparenz: Alle Daten der OWASP von den Finanzen bis zum Code werden öffentlich zur Verfügung gestellt.
- Innovation: Die OWASP unterstützt Innovationen und Experimente welche die Software-Sicherheit verbessern sollen.
- Global: Jeder wird ermutigt der OWASP Gemeinschaft beizutreten.
- Integrität: Die OWASP verhält sich neutral und ist frei von wirtschaftlichen oder politischen Einflüssen.

[Coma]

Diese Arbeit nutzt für die Entwicklung des Sicherheitsmodells zur Bestimmung des (aktuellen) Sicherheitslevels eines Smartphones Erkenntnisse und Bewertungen des *OWASP Testing Guide Project*. Für das Testen von mobilen Anwendungen wurden Dokumente und Informationen des *OWASP Mobile Security Project* verwendet.

2.8.1 OWASP Testing Guide Project

Das *OWASP Testing Guide Project* wurde mit dem Ziel gegründet, einen de-facto Standard für das Penetration Testing von Webanwendungen zu schaffen. Version 1.0 des Guides wurde 2004 veröffentlicht, Version 4.0 (2014) ist die aktuellste Fassung des Testing Guides.
Das Dokument ist unterteilt in das OWASP Testing Framework, dass sich konzeptionell mit den Entwicklungsschritten (Definition, Design, Entwicklung, Veröffentlichung) einer (Web-)Anwendung befasst, während im zweiten Part das Penetrationtesten von Webanwendungen (Web Application Security Testing) aus technischer Sicht beschrieben wird.

OWASP Risk Rating Methodology

Die *OWASP Risk Rating Methodology* beschäftigt sich mit der Bewertung von Bedrohungen und deren finanziellen Auswirkungen. Die Risikobewertung kann bereits in frühen Entwicklungsphasen eingesetzt werden, eignet sich jedoch auch in späteren Phasen beispielsweise zur Klassifikation von ermittelten Schwachstellen während eines Penetrationstests.

Die Standardformel zur Risikobewertung lautet:

Risiko = Wahrscheinlichkeit * Auswirkungen [Comf]

Für die Durchführung einer Risikobewertung eignet sich eine Modellierung. Das Modell beginnt mit der Identifikation einer Bedrohung, welche einer Bewertung unterzogen werden soll. Die *(Eintritts-)Wahrscheinlichkeit* in der OWASP Risikobewertung wird in die Analyse des potentiellen Angreifers und in die Schwachstelle selbst untergliedert. Für die *Auswirkungen* entscheidend sind technische Konsequenzen (z.B. Einschränkung der Verfügbarkeit oder fehlende Integrität der Daten) und die finanziellen Schäden, welche bei Ausnutzung der Schwachstelle entstehen. Nachfolgend folgt eine Übersicht der Standardparameter für die verschiedenen Aspekte:

Technische Auswirkungen

- Vertraulichkeit
- Integrität
- Verfügbarkeit
- Nachvollziehbarkeit

Finanzielle Auswirkungen

- Vertraulichkeit
- Integrität
- Verfügbarkeit
- Nachvollziehbarkeit

Angreifer

- Fähigkeiten
- Motivation
- Aufwand
- Anzahl Angreifer

Schwachstelle

- Entdeckung
- Ausnutzung
- Bekanntheit
- Entdeckung

Jeder dieser Parameter erhält eine Bewertung zwischen 0 und 9. Anschließend wird für die Aspekte *technische Auswirkungen* und *finanzielle Auswirkungen* ein Durchschnittswert berechnet, welcher die Auswirkungen darstellt.
Die Aspekte *Angreifer* und *Schwachstelle* werden ebenfalls einer Bewertung unterzogen und deren errechneter Durchschnittswert stellt die Eintrittswahrscheinlichkeit dar.

Die ermittelten Durchschnittswerte werden anschließend in die Kategorien niedrig, mittel und hoch unterteilt (siehe Abbildung 2.7).
[Comf]

Wahrscheinlichkeit & Auswirkungen	
0 - 3	**NIEDRIG**
3 - 6	**MITTEL**
6 - 9	**HOCH**

Abbildung 2.7: Eintrittswahrscheinlichkeit und Auswirkungen [Comf]

Die Kreuzung der Kategorien *Auswirkungen* und *Eintrittswahrscheinlichkeit* ergibt das Gesamtrisiko. Die Abbildung 2.8 stellt die möglichen Bewertungen vor.

Gesamtrisikobewertung = Wahrscheinlichkeit x Auswirkungen				
Auswirkungen	*HOCH*	Mittel	Hoch	Kritisch
	MITTEL	Niedrig	Mittel	Hoch
	NIEDRIG	Info	Niedrig	Mittel
		NIEDRIG	*MITTEL*	*HOCH*
	Wahrscheinlichkeit			

Abbildung 2.8: Risikobewertung [Comf]

2.8.2 OWASP Mobile Security Project

Für die Bewertung des Sicherheitslevels von mobilen Anwendungen veröffentlichte das *OWASP Mobile Security Project* umfassende Informationen. Das Projekt ist unterteilt in die Kategorien:

- Top 10 Mobile Risks
- Mobile Security Checklist
- Mobile Security Testing Guide
- Mobile Tools
- Secure Mobile Development
- Top 10 Mobile Controls
- Mobile Threat Model Project

Top 10 Mobile Risks

Das *OWASP Top 10 Mobile Risks* Project veröffentlicht seit 2014 im 2-Jahres Rhythmus die am häufigsten gefundenen Bedrohungen von mobilen Anwendungen. Grundlage der letzten Veröffentlichung (2016) war eine öffentliche Befragung an der zahlreiche Unternehmen teilnahmen[13].

- M1: Fehlimplementierung von Features oder Sicherheitsmechanismen der Plattform (z.B. TouchID, Keychain, Berechtigungen)
- M2: Unsichere Datenspeicherung auf dem Gerät
- M3: Unsichere Kommunikation (z.B. veraltete SSL Version, Klartext-Kommunikation)
- M4: Unsichere Authentifizierung
- M5: Unzureichende Verschlüsselung
- M6: Unsichere Autorisierung
- M7: Codequalität der (Client-)Applikation
- M8: Codemanipulationen (z.B. hooking)
- M9: Reverse Engineering
- M10: Irrelevante Funktionalität (z.B. Backdoors während Entwicklung, welche im Produktivbetrieb nicht entfernt wurden)

[Comc]

[13] Die Rohdaten für die Auswahl der zehn am häufigsten ermittelten Bedrohungen sind unter diesem Link: https://www.dropbox.com/sh/d143o6tbkdx4w4l/AAAQIpmnCpHCgiBqZkgXPSTKa?dl=0 abrufbar.

Mobile Security Testing Guide

Das *OWASP Mobile Security Testing Guide* Projekt ist mit der Vision angetreten, einen industriellen Standard für die Sicherheit von mobilen Anwendungen zu definieren:

> *„Define the industry standard for mobile application security"[Comd]*

Die *OWASP Mobile Security Checklist* enthält umfangreiche Sicherheitsanforderungen, welche bei der Entwicklung einer mobilen Applikation berücksichtigt werden müssen. Neben einem allgemeinen Part existieren individuelle Anforderungen für die Plattform Android und iOS. Im ersten Quartal 2018 ist eine Buchveröffentlichung geplant.

Mobile Tools

Unter der Kategorie *Mobile Tools* werden Informationen über nützliche Tools zum Testen von Apps veröffentlicht.

Secure Mobile Development

Die *OWASP Secure Mobile Development Guidelines* bieten Entwicklern in diesem Bereich Coding Guidelines zur sicheren Entwicklung einer mobilen Anwendung an. Diese sind unterteilt in die Bereiche:

- Authentifizierung und Passwortmanagement
- Codeobfuskierung
- Kommunikationssicherheit
- Datenspeicherung und -sicherheit
- Zugangsschutz während Bezahlvorgang
- Schutz des Backends
- Sessionverwaltung
- Bibliotheken / Code von Drittanbietern

[Comg]

Top 10 Mobile Controls

Die *OWASP* hat zusammen mit der *European Network and Information Security Agency (NESA)* ein Konzept veröffentlicht, welches die zehn wichtigsten konzeptionellen Maßnahmen und Designprinzipien für eine sichere Entwicklung von Smartphoneanwendungen enthält. Dieses Dokument wurde ebenfalls von der NESA unter dem Namen *„Smartphone Secure Development Guidelines for App Developers"* zur Verfügung gestellt.
[Comh]

Mobile Threat Model Project

Das *OWASP Mobile Threat Model Project* skizziert in seinem Modell mögliche Angriffsfelder gegen mobile Anwendungen. Nachfolgende Grafik 2.9 verdeutlicht die verschiedenen Angriffsvektoren:

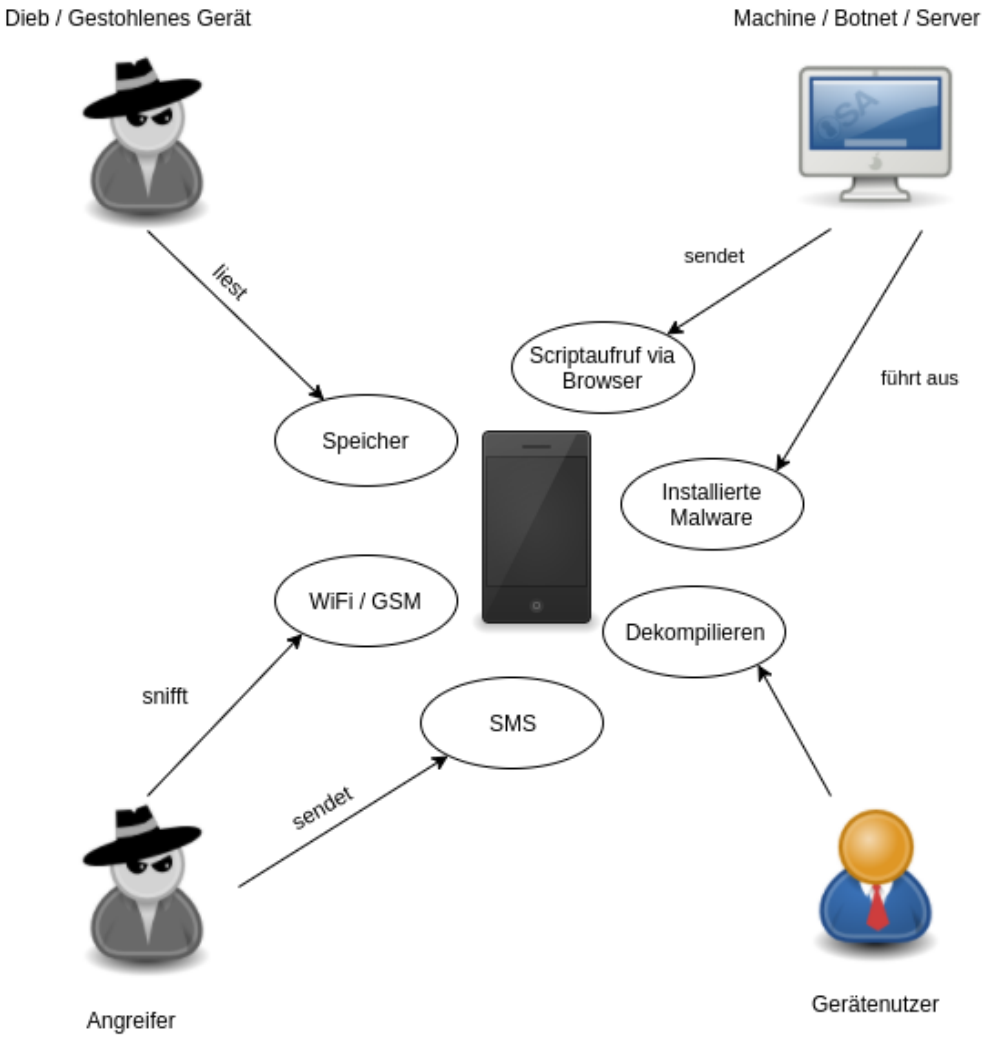

Abbildung 2.9: Mobile Threat Agent identifier and types [Est]

3. Verwandte Arbeiten

Dieses Artikel liefert einen Überblick bereits existierender Sicherheitsmodelle in der Informatik. Bei der Recherche wurde primär nach Modellen für mobile Betriebssysteme und Endgeräte gesucht. Hierbei zeigte sich, dass bestehende Modelle häufig indirekt auch für Teilbereiche von mobilen Geräten genutzt werden (können).

3.1 Formale Sicherheitsmodelle

Mittels formaler Modelle kann bewiesen werden, dass eine Funktion oder ein Modul, welche als logische Anforderung formuliert wurde, eine bestimmte Funktion erfüllt. Im Gegensatz zu Programmtests (z.B. Unit Tests), welche nur garantieren, dass die gegebene Funktion alle Anforderungen des Tests erfüllt, kann ein formales Modell die Korrektheit eines Moduls nachweisen. [Kle09]

In der Informatik existieren allgemein verfügbare, formale Sicherheitsmodelle vorwiegend im Bereich der Informationskontrolle von Betriebssystemen. Diese Modelle können unterteilt werden in zugriffskontroll-, informationsfluss- und transaktionsbasierte Modelle.

3.1.1 Zugriffskontrollbasierte Modelle

Zugriffskontrollbasierte Modelle können hinsichtlich der Art von Berechtigungen in Zugriffsmatrixmodelle und eigenschaftsbasierte Modelle unterteilt werden. Zugriffsmatrixmodelle regeln den Zugriff (Rechte) von Subjekten auf Objekte (z.B. Benutzer A liest Datei X). Eigenschaftsbasierte Modelle verwenden Rollen um den Zugriff auf Objekte zu regeln (z.B. RBAC0-Modell [1]). Daneben existieren Modelle (z.B. Schematic Protection Modell), welche Funktionalitäten aus beiden Modellen übernehmen.
[Kep13]

Bell-LaPadula

Unter diese Rubrik fällt das *Bell-LaPadula Sicherheitsmodell*, welches als das erste vollständig formalisierte Modell gilt. Dieses Modell gewährleistet die Vertraulichkeit von

[1] RBAC0-Modell: Role Based Access Control Modell

Informationen und findet seine Anwendung in Bereichen, welche eine konsequente Trennung von Zuständigkeitsbereichen und Informationen besitzen. Dies ist beispielsweise im Militär vorzufinden. Das Modell setzt dabei den Fokus auf die Gewährleistung, dass es nicht möglich ist, Zugriff auf Informationen aus einer höheren Berechtigungsklasse zu erhalten und dass Objekte höherer Berechtigungsklassen nicht die Möglichkeit besitzen, Informationen mit einer niedrigeren Berechtigungsstufe zu teilen. SELinux, eine Erweiterung des Linux-Kernels, nutzt für die Zugriffskontrolle das Bell-LaPadula-Modell[Ric].

Biba

Das *Sicherheitsmodell von Biba* ermöglicht im Gegensatz zum Bell-LaPadula Modell die Wahrung der Integrität von Daten. Dazu wird vor jedem Zugriff überprüft, dass es Akteuren aus höheren Ebenen nicht möglich ist, Informationen einer niedrigeren Ebene zu lesen („No-Read-Down'") , während es Akteuren niedrigerer Ebenen nicht erlaubt ist, Informationen in höhere Berechtigungsebenen zu schreiben („No-Write-UP'").
[BSP]

Brewer-Nash-Modell (Chinese-Wall)

Das Brewer-Nash Sicherheitsmodell hat seine Ursprünge in der Finanzindustrie. Es soll Interessenskonflikte von beispielsweise Bankberatern mit Geheiminformationen unterbinden.
[WM16]

3.1.2 Informationsflussbasierte Modelle

Denning

Das *Denning-Modell* ist ein informationsflussbasiertes Modell, welches das zugriffsorientierte *Bell-LaPadula* Modell um dynamische Zustände und Zustandsübergänge erweitert, mit dem Ziel, unautorisierte Informationsflüsse zu unterbinden.
[Kep13; PP03]

3.1.3 Transaktionsbasierte Modelle

Gleichgewichtsmodell

Das Gleichgewichtsmodell, welches den transaktionsbasierten Modellen zugeordnet wird, stellt eine Möglichkeit dar, über einen dezentralen Kanal (z.B. Internet) verbindlich zu kommunizieren. Ausgangssituation dieses Modells ist, dass ein Akteur sein eigenes Verhalten beeinflussen kann, jedoch nicht das seines Gegenübers. Um beispielsweise im digitalen Handel eine Verbindlichkeit herzustellen, ist ein Mechanismus erforderlich der es erlaubt, dass entweder beide Akteure erfolgreich sind oder keiner von beiden.
[Rüd]

3.2 Sicherheitsmodell auf Basis von ISMS

Für den Einsatz von Smartphones in Unternehmen existiert ein erweitertes Sicherheitsmodell auf Basis eines Information Security Management System (ISMS). Dieses Sicherheitsmodell liefert einen rudimentären Überblick möglicher Angriffsvektoren gegen Smartphones und leitet daraus Anforderungen an die Sicherheit von Smartphones ab.

Diese Anforderungen werden in die vier Werte *Authentizität*, *Anonymität*, *Integrität* und *Vertraulichkeit* unterteilt. Diese Werte stellen die Grundlage einer Reihe von Maßnahmen dar, welche ein Unternehmen etablieren soll. Die Maßnahmen werden in technische (z.B. Zugriffsschutz und Verschlüsselung gespeicherter Inhalte) und organisatorische (z.B. Etablierung eines Managements für Sicherheitsvorfälle) Aspekte unterteilt.
[PYJ14]

3.3 Sicherheitsmodell für mobile Betriebssysteme

Die *Information Assurance* der *National Security Agency (NSA)*[2] veröffentlichte 2013 ein Dokument, welches Sicherheitsanforderungen für mobile Betriebssysteme definiert. Das Modell fußt auf einer Beschreibung möglicher Bedrohungen auf Basis der häufigsten Angriffsvektoren gegen Geräte mit einem mobilen Betriebssystem. Diese Bedrohungen resümieren in definierte Sicherheitsziele dieses Modells, welche anschließend in den Sicherheitsanforderungen technisch detailliert beschrieben werden.
[NSA13]

3.4 Fazit

Die ausführliche Recherche nach bereits existierenden Sicherheitsmodellen für Smartphones oder mobile Endgeräte ergab, dass bis dato kein Modell existiert, welches die Gesamtsicherheit eines Gerätes bewertet. Der Fokus *formaler Sicherheitsmodelle* liegt in Zugriffskontrollstrategien für Betriebssysteme auf Datei- und Prozessebene. Diese bilden somit eine detaillierte Perspektive auf eine von vielen Sicherheitsanforderungen an ein Smartphone. Deswegen eignet sich die Verwendung dieser Modelle gut, um die Zugriffskontrolle für das Betriebssystem zu realisieren, es stellt aber nur eine von zahlreichen Sicherheitsfeatures innerhalb des Ökosystems Smartphone dar.

Im Gegensatz zu den formalen Sicherheitsmodellen besitzt das von den Autoren Jong Hyuk Park, Ki Jung Yi und Young-Sik Jeong 2014 veröffentlichte *Sicherheitsmodell auf Basis eines Information Security Management Systems (ISMS)* ein Modell, welches die Gesamtsicherheit eines Smartphones bewertet, es beschreibt aber nach Meinung des Autors nur sehr oberflächlich die Bedrohungen von Smartphones. Die Analyse dieser Gefährdungen beinhaltet ausgehende Gefahren, die von einer Massenverbreitung von Schadcode, durch die Vielzahl an Nutzern mit eigenem Smartphone reichen, über die Entwicklung eines Kernel Rootkits, bis hin zu einem durch bösartigen Code verursachten Schaden durch sogenannter „Zombie'Geräte, welche von einem Kommandoserver gesteuert werden. Der Schaden fällt im Vergleich zu herkömmlichen Computern höher aus, da zusätzliche Schnittstellen wie GPS, Mikrofon oder Fotoaufnahmen missbraucht werden können.

Das von der NSA 2013 veröffentlichte Dokument zu den *Sicherheitsanforderungen eines mobilen Betriebssystems* beschreibt technisch sehr detailliert, welche Anforderungen zwingend implementiert werden müssen. Dies hat zur Folge, dass dieses Dokument Mängel in der Aktualität besitzt. So wird beispielsweise SHA-1 in diesem Dokument als MÖglichkeit zur Erstellung kryptographischer Schlüssel dargestellt, während 2005 ein Angriff durch

[2] NSA: Ein Auslandsgeheimdienst der Vereinigten Staaten von Amerika.

chinesische Kryptologen vorgestellt wurde, der die Anzahl erforderlicher Berechnungen zum Auffinden einer Kollision deutlich reduzierte. 2017 stellten Forscher von Google und dem Forschungsinstitut CWI Amsterdam eine weitere Angriffsmöglichkeit vor, welche die Anzahl der Berechnungen nochmals reduzierte[Ste+17]. Zudem werden Sicherheitsaspekte der Hardware und des Managements solcher Geräte in diesem Dokument nicht berücksichtigt, da der Fokus auf das Betriebssystem gerichtet ist.

Abschließend lässt sich konstatieren, dass bereits verwendbare Sicherheitsmodelle für mobile Endgeräte existieren, diese jedoch nur Teilaspekte berücksichtigen. Die Anforderung dieser Masterarbeit ist die Erarbeitung eines Sicherheitsmodells, welches die Sicherheit von Hardware, Betriebssystem, Management und der verwendeten Applikationen eines ausgewählten Smartphones beurteilen soll. Da diese Anforderung von keinem dieser Modelle erfüllt wird, ist die Einführung eines neuen Sicherheitsmodells für Smartphones erforderlich.

4. Sicherheitsmodell für Smartphones

Dieses Kapitel widmet sich zu Beginn mit der Vorstellung eines geeigneten Prozesses zur Erstellung des Sicherheitsmodells. Nach der Festlegung auf ein Vorgehensmodell, folgt die Darstellung der Entwicklung des Sicherheitsmodells gemäß des vorgesehenen Prozesses. Die Evaluation des Sicherheitsmodells erfolgt abschließend durch die Anwendung an drei fiktiven Nutzergruppen mit jeweils zwei unterschiedlichen Smartphones.

4.1 Vorgehensmodell

Die im Grundlagenteil vorgestellte *OWASP Risk Rating Methodology* ermöglicht die Bewertung einer konkreten Bedrohung hinsichtlich Eintrittswahrscheinlichkeit und Auswirkungen. Zur Identifikation aller Bedrohungen ist zusätzlich ein Vorgehensmodell erforderlich. Hierzu eignet sich das von Professor Dr.-Ing. Hans-Joachim Hof an der Technischen Hochschule Ingolstadt im Wintersemester 2016/2017 vorgestellte Modell zur Bedrohungs- und Risikoanalyse.

Das Modell besteht aus einem sechsstufigen Prozess, der zu Beginn die Werte für Smartphones identifiziert, anschließend folgt ein Überblick der Architektur, welcher eine Zerteilung des Systems in kleinere Komponenten erlaubt. Im Verlauf erfolgt eine Analyse und Bewertung der Bedrohungen, welche abschließend durch geeignete Gegenmaßnahmen minimiert oder verhindert werden. Der Ablauf wird in Abbildung 4.1 graphisch erläutert.[Hof]

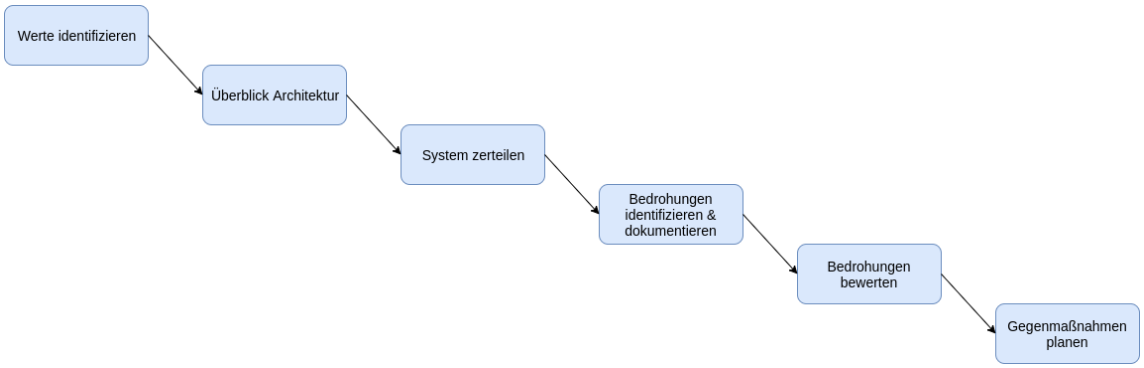

Abbildung 4.1: Vorgehensmodell zur Bedrohungs- und Risikoanalyse[Hof, S. 13]

Das in Abbildung 4.1 dargestellte Vorgehen wird in dieser Arbeit als Grundlage für die Bestimmung des (aktuellen) Sicherheitslevels eines Smartphones genutzt, bedarf für die Entwicklung eines auf Smartphones abgestimmten Bestimmung des Sicherheitslevels jedoch einer Anpassung. Die Anpassungen sind in der fünften und sechsten Phase des Prozesses erforderlich. Während das Vorgehensmodell die identifizierten Bedrohungen einer Bewertung unterzieht, werden diese im Sicherheitsmodell für Smartphones in Anforderungen umformuliert, um eine Analyse der Implementierungen der Hersteller zu ermöglichen, da eine Bewertung der Bedrohungen ohne Kenntnisse der Infrastruktur und des Source-Codes der Hersteller, nicht durchführbar ist. Durch die Analyse der Implementierungen ist anschließend eine Recherche nach öffentlich verfügbaren Schwachstellen möglich. Letztere werden anschließend einer *Bewertung unterzogen*, welche neben dem Ausmaß der Schwachstelle und den technischen Auswirkungen, das Angreiferumfeld und die finanziellen Auswirkungen der Nutzergruppe umfasst. Hierzu eignet sich die Verwendung des *OWASP Risk Rating* besonders, da alle oben genannten Bewertungsaspekte berücksichtigt werden. Die Verwendung alternativer Bewertungsmodelle für Schwachstellen, wie beispielsweise das *Common Vulnerability Scoring System (CVSS)*, besitzen den Nachteil, dass eine isolierte Bewertung aus rein technischer Sicht durchgeführt wird. Die Anforderung an dieses Sicherheitsmodell sieht jedoch eine Bewertung der Schwachstelle in Verbindung mit den Auswirkungen für die Anwendergruppe vor.

Im Vorgehensmodell sind im letzten Schritt die Ergreifung geeigneter *Gegenmaßnahmen* vorgesehen, welche ebenfalls erweitert werden müssen. Das Ziel dieses Modells ist eine Darstellung des aktuellen Sicherheitslevels eines Smartphones und geeignete Gegenmaßnahmen beschränken sich auf die Empfehlung hinsichtlich der Verwendung des ausgewählten Gerätes. Die Darstellung des Gesamt-Sicherheitslevels in einem numerischen Wert erscheint nicht sinnvoll, da im Modell Anforderungen vorhanden sind, welche bei Nichterfüllung nicht durch andere Anforderungen „ausgeglichen“ werden können.

Die ursprüngliche Abbildung 4.1 wird nachfolgend angepasst dargestellt:

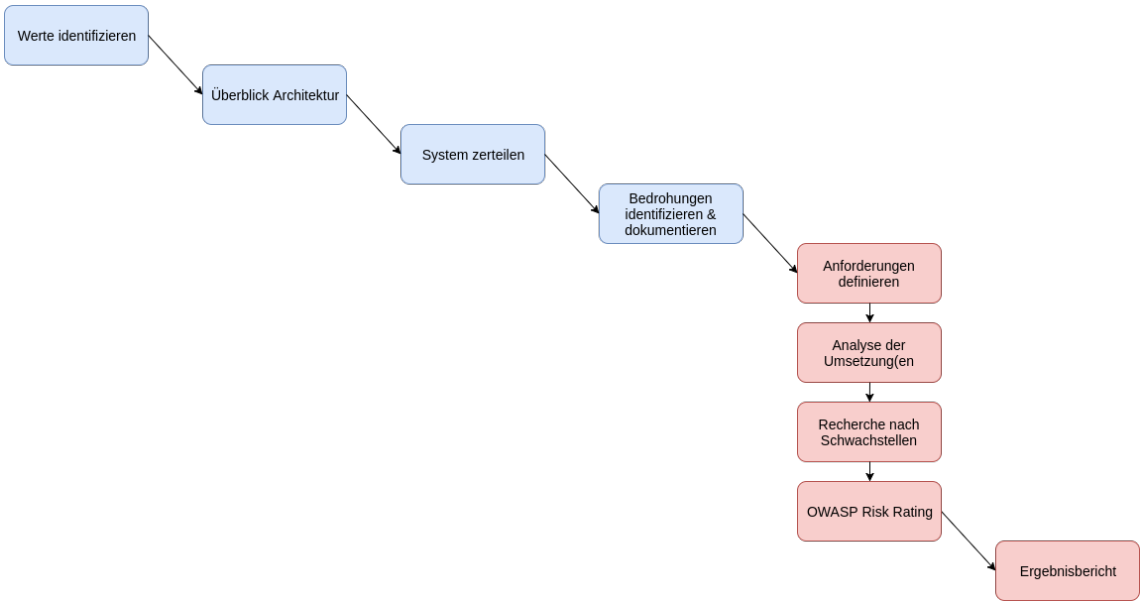

Abbildung 4.2: Angepasstes Vorgehensmodell zur Bedrohungs- und Risikoanalyse

4.2 Vorgehen zur Bedrohungs- und Risikoanalyse

Dieser Abschnitt wendet die Phasen des angepassten Vorgehensmodells zur Bestimmung des (aktuellen) Sicherheitslevels an.

4.2.1 Werteidentifikation

Gemäß dem Vorgehensmodell zur Risiko- und Bedrohungsanalyse folgt zu Beginn dieses Modells eine Auflistung der Werte. Hierzu existieren in der Literatur verschiedene Varianten, um die Werte und damit die Schutzziele eines Modells oder einer Anwendung abzubilden.

4.2.1.1 Evil User Stories

Die Verwendung von *Evil User Stories* oder auch *Attacker Stories* hat seine Ursprünge in *Software Development Life Cycle (SDLC)* und agilen Methoden zur Softwarentwicklung (z.B. Scrum). Nutzergeschichten (User Stories) beschreiben allgemein in natürlicher Sprache und in wenigen Sätzen eine Anforderung. Nachdem eine *User Story* definiert wurde, werden im nächsten Schritt alle Szenarien eruiert, welche ein Angreifer nutzen könnte, um diese Anforderung zu attackieren. Diese Szenarien werden in *Evil User Stories* festgehalten.[Fit15]
Die OWASP illustriert in ihrem *OWASP Application Security Verification Standard Project* folgendes Beispiel[Comb]:

Nutzergeschichte (Product Backlog):

Als ein Mitarbeiter kann ich nach anderen Mitarbeitern durch Eingabe des Nachnamens suchen.

Evil User Stories:

- Evil User Story 1: Als ein Angreifer kann ich durch Aufruf einer manipulierten URL, Zugriff auf Daten und Funktionen erhalten, zu denen ich keine Berechtigung habe.
- Evil User Story 2: Als ein Angreifer kann ich durch das Mitsenden von bösartigem Inhalt in einer Anfrage, Zugriff auf Daten und Funktionen erhalten, zu denen ich keine Berechtigung habe.
- Evil User Story 3: Als ein Angreifer kann ich durch Manipulation des HTTP Headers, Zugriff auf Daten und Funktionen erhalten, zu denen ich keine Berechtigung habe.
- Evil User Story 4: Als ein Angreifer kann ich alle Daten lesen und gegebenenfalls manipulieren, welche durch die Anwendung eingelesen und ausgegeben werden.

In der weiteren Planung müssen bei jedem Sprint diese *Evil User Stories* berücksichtigt werden.

Für die Verwendung im Sicherheitsmodell für Smartphones ist eine Adaption erforderlich. Nach der Identifikation der Bedrohungen und einer Definition der Anforderungen (Kapitel 4.2.4) müssen zu jeder Anforderung *Evil User Stories* identifiziert und bei der Bewertung der Anforderungen berücksichtigt werden.

Aus Sicht des Autors stellt dieses Vorgehen eine Möglichkeit zur Werteidentifikation dar, es birgt jedoch den großen Nachteil, dass ein sehr hoher Aufwand zur Identifikation aller Angreiferszenarios entsteht.

4.2.1.2 Security Stories

Eine Alternative zu *Evil User Stories* stellt die zusätzliche Definition von normalen *User Stories* mit Sicherheitsanforderungen dar. Dieses Verfahren findet bei der Verwendung von *Security Stories* statt. Diese Nutzergeschichten können anschließend in den Sprints eingeplant und priorisiert werden. Im Unterschied zu einer gewöhnlichen *User Story* werden diese Nutzergeschichten nicht aus der Perspektive eines Nutzers oder Kunden beschrieben, sondern richten sich direkt an das Entwicklerteam.[UM15]

- Beispiel: Als ein (Software-)Architekt möchte ich sicherstellen, dass sensitive Daten so geschützt werden, dass nur autorisierte Nutzer Zugriff haben.[Bir]

Um die Definition derartiger Sicherheitsanforderungen zu erleichtern, veröffentlichte das *Software Assurance Forum for Excellence in Code (SafeCODE)* einen Artikel mit zahlreichen Vorlagen[Ast+12].

Nach Meinung des Autors stellt dieses Vorgehen eine sinnvolle Möglichkeit der Werteidentifikation dar, führt jedoch ebenso zu dem Nachteil, dass für den Umfang eines Smartphones die Anzahl an *Security Stories* eine sehr hohe Anzahl derartiger Nutzergeschichten produzieren würde und eine praktische Anwendung dieser Stories nur mit einem sehr hohen zeitlichen Aufwand realisierbar wäre.

4.2.1.3 Grundwerte der Informationssicherheit

Die klassische Werteidentifikation auf Basis der drei Grundwerte in der Informationssicherheit definiert nachfolgende Werte[Osc03; Sicf]. Die Definition dieser und weiterer geeigneter Werte eignet sich für dieses Sicherheitsmodell im Besonderen, da die vorangegangenen Vorgehensweisen (Evil User Stories, Security Stories) eine Vielzahl an Varianten erlauben und aufgrund der Größe des Sicherheitsmodells zur Bestimmung des (aktuellen) Sicherheitslevels eines Smartphones praktisch nicht durchführbar wären.

- Vertraulichkeit
- Verfügbarkeit
- Integrität

Eine Definition dieser Werte fand bereits im Grundlagenteil (Kapitel 2.1) statt und bedarf keiner weiteren Erklärung.

Bei der Verarbeitung personenbezogener Daten, existieren neben den oben genannten Grundwerten zusätzliche Schutzziele[Bud+14; Ber; Sch+16]:

Authentizität

> *„Authentizität bedeutet, dass Daten jederzeit ihrem Ursprung zugeordnet werden können."[Ber]*

Revisionsfähigkeit

> *„Revisionsfähigkeit bezieht sich auf die Organisation des Verfahrens. Sie ist gewährleistet, wenn Änderungen an Daten nachvollzogen werden können."[Ber]*

Transparenz

> *„Transparenz ist gewährleistet, wenn das IT-Verfahren für die jeweils Sachkundigen in zumutbarer Zeit mit zumutbarem Aufwand nachvollziehbar ist. Für Betroffene muss die Verarbeitung ihrer Daten vollständig nachvollziehbar sein. In der Regel setzt dies eine aktuelle und angemessene Dokumentation voraus."[Ber]*

Datensparsamkeit

> *„Personenbezogene Daten dürfen nur erhoben und verarbeitet werden, so lange sie für die Erfüllung der Aufgaben erforderlich sind. Werden personenbezogene Daten nicht mehr benötigt, sind sie zu löschen. Es muss stets begründet werden, warum die Daten benötigt werden."[Ber]*

Zweckbindung

> *„Personenbezogene Daten dürfen nur für den Zweck verwendet werden, zu dem sie erhoben wurden. Werden personenbezogene Daten für diesen Zweck nicht mehr benötigt, sind sie zu löschen."[Ber]*

Intervenierbarkeit

> *„Die Technik muss es ermöglichen, dass die Rechte von Betroffenen jederzeit gewahrt werden können."[Ber]*

Informationelles Selbstbestimmungsrecht

> *„Betroffene haben das Recht, selbst über die Preisgabe und Verwendung ihrer Daten zu entscheiden."[Ber]*

Für dieses Modell eignet sich die Aufnahme der Werte *Transparenz* und *Intervenierbarkeit*. Die Werte *Authentizität*, *Revisionsfähigkeit*, *Datensparsamkeit*, *Zweckbindung* und *informationelles Selbstbestimmungsrecht* betreffen die Erhebung und Verarbeitung von Daten und sind damit irrelevant für das Sicherheitsmodell, da die zu schützenden Daten auf einem Smartphone bereits erhoben wurden und eine Bewertung des Datenschutzes aus Perspektive der Privatsphäre nicht zum Umfang dieses Modells gehört.

4.2.2 Architekturüberblick

Im Grundlagenteil wurde bereits eine Definition des Smartphones vorgenommen. Nachdem 99,6 Prozent aller Smartphones als Betriebssystem Android oder iOS verwenden (Stand 4. Quartal 2016, siehe Abbildung 4.3), beschränkt sich die Auswahl der Plattformen auf diese beiden Hersteller. Sollten sich an dieser Verteilung in der Zukunft Änderungen ergeben, kann dieses Sicherheitsmodell ohne hohen Aufwand um eine weitere Architektur ergänzt werden.

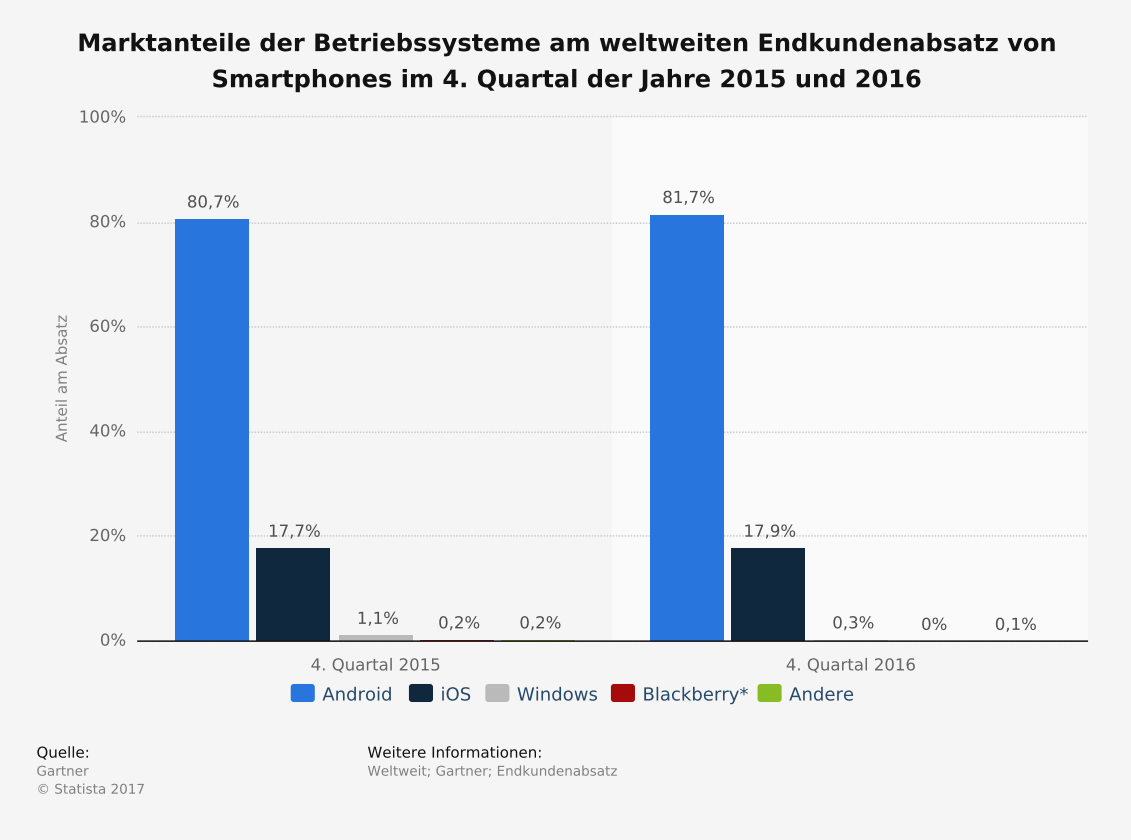

Abbildung 4.3: Marktanteile der Betriebssysteme im 4. Quartal 2015 und 2016 [Ins]

4.2.3 Systemzerteilung

Zur leichteren Identifikation der Bedrohungen eignet sich, wie im Modell zur Bedrohungs- und Risikoanalyse angegeben, im nächsten Schritt eine Zerteilung des Systems. Ausgehend von einer Unterteilung in die Komponenten Hardware und Software, lässt sich eine weitere Verfeinerung durchführen:

- Hardware
  - Schnittstellen
  - Trusted Execution Zone
- Betriebssystem
  - Allgemein
  - Systemsicherheit
  - Applikationssicherheit
  - Datensicherheit
- Management
  - Allgemein
  - Laufzeitanalyse
  - Mobile Device Management
  - Transparenz
- Anwendung(en)

Die Beurteilung des Sicherheitslevels einer Anwendung kann nicht generisch für einen Gerätetypen vorgenommen werden, sondern unterliegt einem individuellen Penetrationstest. Die Durchführung eines Penetrationstest für mobile Applikationen wird in Kapitel 5 erörtert. Aus diesem Grund wird die Beurteilung des (aktuellen) Sicherheitslevels auf Anwendungsebene separat vorgenommen.

4.2.4 Bedrohungen identifizieren

Die Recherche aktueller Bedrohungsszenarien ergab folgende Grafik (Abbildung 4.4):

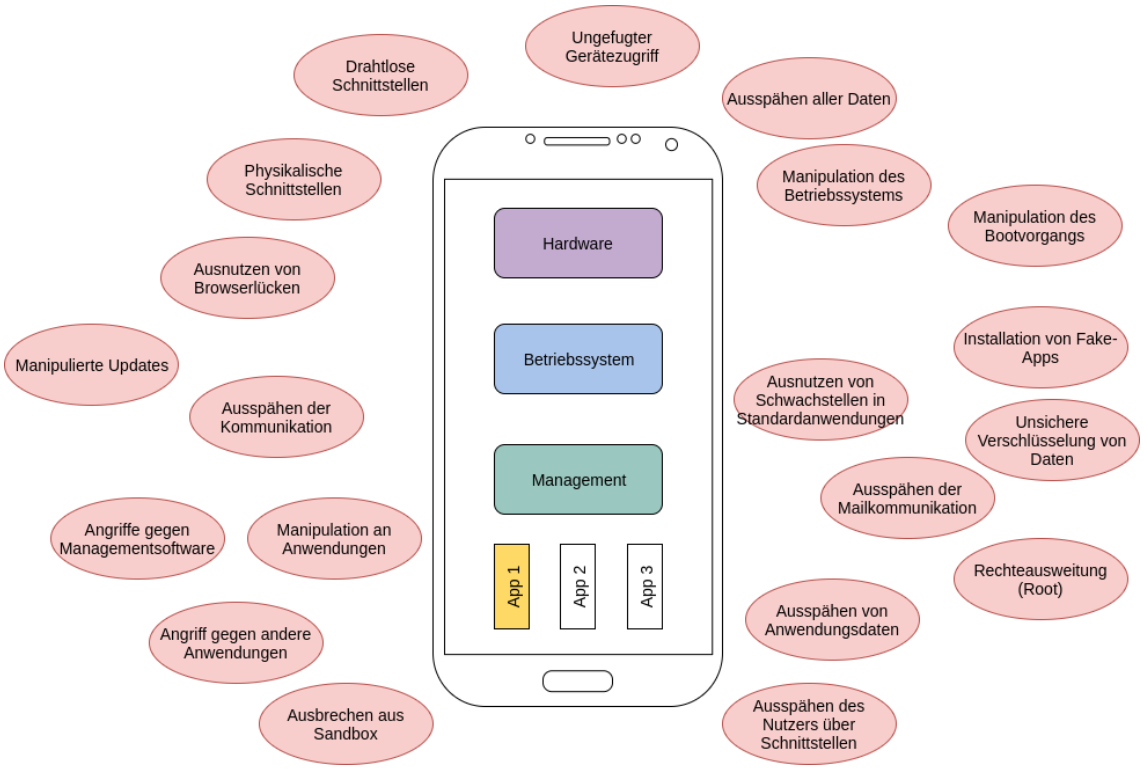

Abbildung 4.4: Identifikation von Bedrohungen eines Smartphones

Um eine möglichst hohe Abdeckung aller vorhandenen Bedrohungen zu erreichen, wurde eine intensive Befragung der Kolleginnen und Kollegen bei der Firma Optima Business Information Technology und Professor Dr.-Ing. Hans-Joachim Hof vorgenommen. Die Identifikation der Risiken basiert auf umfangreichen Quellen[HE12; FBT13; GMA14; McA; Lab].

4.2.5 Anforderungen definieren

Auf Basis der Bedrohungsanalyse aus Kapitel 4.2.4 wurden insgesamt 41 Anforderungen identifiziert.

Hardware

Diese Kategorie listet alle Anforderungen auf, welche auf Hardwareebene umgesetzt werden müssen.

Schnittstellen

- R-HW.01 Physikalische Schnittstellen müssen sicher vor unautorisierter Nutzung sein.
- R-HW.02 Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Trusted Execution Zone (Hardwaregetrennter Bereich)

- R-HW.03 Das Gerät muss über hardwaregetrennte Module verfügen, welche sicherheitsrelevante Aufgaben übernehmen.
- R-HW.03.1 Das Gerät muss Hardwareunterstützung zur Verschlüsselung von Daten ermöglichen.
- R-HW.03.2 Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.
- R-HW.03.3 Kostenintensive Vorgänge oder sensible Daten müssen in hardwaregetrenntem Bereich verarbeitet werden.
- R-HW.04 Vor dem Systemstart muss eine hardwareunterstützte Integritätsüberprüfung durchgeführt werden.4

Sollte die Anforderung (R-HW.03.1) nicht in Hardware implementiert sein, kann alternativ eine Realisierung auf Betriebssystemebene (R-OS.01 in Verbindung mit R-OS.14) überprüft werden. Gleiches gilt für die Anforderung R-HW.03.2 , welche durch die Anforderung R-OS.14 abgedeckt wird und die Anforderung R-HW.04 , welche durch die Anforderung R-OS.04 alternativ implementiert werden kann.
Es gilt jedoch zu beachten, dass bei fehlender Realisierung dieser Anforderungen in Hardware, eine Umsetzung auf Betriebssystemebene eine Alternative darstellt, diese jedoch keinen gleichwertigen Ersatz darstellt. Ziel eines Hardwareherstellers für Smartphones muss eine Erfüllung der Hardwareanforderungen sein.

Betriebssystem

Diese Kategorie listet alle Anforderungen auf, welche durch das Betriebssystem umgesetzt werden müssen.

Allgemein

- R-OS.01 Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.
- R-OS.02 Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

Systemsicherheit

- R-OS.03 Die Updates des Betriebssystems müssen authentisch sein.
- R-OS.04 Der Bootvorgang und das Laden aller für den Betrieb erforderlichen Komponenten müssen auf deren Integrität überprüft werden.
- R-OS.05 Das Betriebssystem soll auf unterschiedlichen Hardware Plattformen verwendbar sein.
- R-OS.06 Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

Anwendungssicherheit

- R-OS.07 Das Betriebssystem muss Anwendungen in sicherem Umfeld ausführen und vor Zugriffen von außen schützen.
- R-OS.08 Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.
- R-OS.09 Eine Anwendung darf nur sicher mit berechtigten externen Schnittstellen kommunizieren.
- R-OS.10 Eine Anwendung darf nur möglichst feingranulare Berechtigungen für den Zugriff auf andere Anwendungen erhalten.
- R-OS.11 Das Betriebssystem muss eine sichere Netzwerkkommunikation für Anwendungen anbieten.

Datensicherheit

- R-OS.12 Das Betriebssystem muss individuellen Speicher einer Anwendung vor externen Zugriff schützen.
- R-OS.13 Das Betriebssystem muss eine Schnittstelle anbieten, um sensitive Daten besonders sicher aufzubewahren.
- R-OS.14 Das Betriebssystem muss alle Benutzerdateien verschlüsselt aufbewahren.

Management

Diese Kategorie listet alle Anforderungen auf, welche das Management eines Gerätes betreffen.

Allgemein

- R-MGMT.01 Das Gerät muss Schutzmechanismen anbieten, die einen ungeprüften und unautorisierten Zugriff verhindern.
- R-MGMT.02 Es muss ein Prozess zur Überprüfung von Anwendungen vor Veröffentlichung in einem App-Store vorhanden sein.
- R-MGMT.03 Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.
- R-MGMT.04 Der Gerätenutzer darf Anwendungen ausschließlich aus sicheren Quellen installieren können.
- R-MGMT.05 Das Gerät muss nativ eine sichere Mailkommunikation anbieten.

Laufzeitanalyse

- R-MGMT.06 Das Gerät muss nach dem Systemstart und im laufenden Betrieb auf Sicherheitsaktualisierungen überprüft werden.
- R-MGMT.07 Manipulationen an installierten Anwendungen müssen erkannt werden.
- R-MGMT.08 Das Betriebssystem soll eine Möglichkeit bieten, Geräte auf denen ein Root-Zugriff etabliert wurde, zu erkennen.
- R-MGMT.09 Es muss eine vertrauenswürdige Instanz geben, welche nachträglich schadhafte Anwendungen deinstallieren kann.

Mobile Device Management

- R-MGMT.10 Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.
- R-MGMT.11 Das Gerät muss die Integration von Sicherheitsrichtlinien ermöglichen.
- R-MGMT.12 Es muss eine individuelle Möglichkeit geben, nur ausgewählte Anwendungen auf einem Gerät installieren zu können.
- R-MGMT.13 Mobile Device Management Lösungen müssen einen Zugriff aus der Ferne erhalten können.
- R-MGMT.14 Es muss eine zentrale Möglichkeit geben, Produkt- und Sicherheitsaktualisierungen auszurollen.
- R-MGMT.15 Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

Transparenz

- R-MGMT.16 Der verwendete Code innerhalb eines Gerätes muss offen liegen, um eine transparente Validierung der Sicherheit zu ermöglichen.
- R-MGMT.17 Hersteller und Anbieter von Betriebssystemen sollen Transparenz bei Sicherheitsproblemen zeigen.
- R-MGMT.18 Gerätehersteller und Betriebssystemanbieter müssen (Sicherheits-) Aktualisierungen in einem definierten Zeitraum anbieten.
- R-MGMT.19 Das Gerät, bestehend aus Hardware- und Softwarekomponenten, muss durch legitimierte Stellen zertifiziert werden.
- R-MGMT.20 Es muss für Behörden auf nationaler Ebene die Möglichkeit geben, Spezifikationen am Gerät und Betriebssystem individuell vorzunehmen.

4.2.6 Analyse der Umsetzung(en) & Recherche nach Schwachstellen

Eine Analyse der technischen Implementierungen kann erst nach Auswahl eines konkreten Smartphones durchgeführt werden. Die Auswertung der Implementierungen und die Recherche nach Schwachstellen der zur Evaluation des Modells ausgewählten Smartphones sind in Anhang B, C, D und E aufgeführt.

4.2.7 OWASP Risk Rating

Die Bewertung des Gesamtrisikos einer Schwachstelle wird gemäß der *OWASP Risk Rating Methodology* in zwei Bereichen vorgenommen. Die *Eintrittswahrscheinlichkeit* wird durch die *Ausnutzbarkeit der Schwachstelle* und die *Fähigkeiten des Angreifers* berechnet. Um den zweiten Part des Gesamtrisikos zu ermitteln, werden die *technischen* und *finanziellen Auswirkungen* betrachtet. Um diese bestimmen zu können, ist eine vorherige Definition des Nutzers oder der Nutzergruppe eines Smartphones erforderlich.

4.2.7.1 Eintrittswahrscheinlichkeit

Die Eintrittswahrscheinlichkeit wird im *OWASP Risk Rating* weiter unterteilt in eine Beurteilung des *Angreiferumfelds* und in die Ausnutzbarkeit der *Schwachstelle*.

Angreifer

Das Umfeld des *Angreifers* wird durch eine Einstufung der Kategorien *Fähigkeiten*, *Motivation*, *Aufwand* und *Angreifer(gruppe)* ermittelt. Die Basis für die Einschätzung des Angreifers stellt eine Analyse des Umfelds einer Benutzergruppe dar. Dabei wird angenommen, dass Angriffe, beispielsweise gegen Nachrichtendienste, nur mit einer Vielzahl an sehr professionellen Angreifern und einer sehr hohen Motivation durchgeführt werden können, da Nachrichtendienste diverse Schutzmechanismen vorsehen und deren Mitarbeiter entsprechend geschult sind. Bei einem kleinen Betrieb wird der Aufwand für einen Angriff als geringer betrachtet, da mit einer hohen Wahrscheinlichkeit die Schutzmechanismen vor Angriffen nicht oder nur teilweise vorhanden sind und die Aufklärung der Mitarbeiter über IT-Sicherheit kaum vorhanden ist.
Das Bundesamt für Sicherheit ermittelt einen normalen Schutzbedarf für die Anwendergruppe eines generischen Kleinunternehmens.[Sica]

Die Parameter zur Bewertung des Umfeld eines Angreifers werden nachfolgend beschrieben:

Fähigkeiten Die Bewertung des technischen Umfelds eines Angreifers basiert auf seinen Fähigkeiten. So wird die Gefahr eines Angreifers ohne technische Fähigkeiten (1) am Geringsten angesehen, während ein Penetrationtester (9) das größte Risiko darstellt. Weitere Unterscheidungen werden in erweiterte technische Fähigkeiten (3), fortgeschrittene technische Fähigkeiten (5) oder Netzwerk- und Programmierkenntnisse (6) vollzogen.

Motivation Die Bewertung der Motivation wird in dieser Bewertung kategorisiert in hohe Motivation (9), mittelmäßige Motivation (4) und keine Motivation (1). Das Vorhandensein einer hohen Motivation bedeutet dabei, dass der Angriff gezielt gegen das Smartphone durchgeführt wird. Eine „mittelmäßige“ Motivation wird definiert als „kein gezielter Angriff gegen ein individuelles Smartphone, jedoch das gezielte Ausnutzen von Schwachstellen um die Sicherheit zu gefährden“, mit dem Ziel beispielsweise die Vertraulichkeit der Daten anzugreifen und damit monetäre Vorteile zu erreichen. Als keine Motivation wird das Ausnutzen von Schwachstellen angesehen, mit dem Ziel Schaden - ohne eigenen Profit - anzurichten.

Aufwand Der Aufwand betrachtet den Umfang an Ressourcen oder Zugriffe, welche erforderlich sind, um diesen Angriff erfolgreich durchzuführen. Adaptiert auf den Kontext eines Smartphones wird eine Unterscheidung in physikalischer Zugriff (0), spezieller Zugriff oder Ressourcen (4), einmaliger Zugriff (7) oder kein Zugriff (9) erforderlich vorgenommen.

Angreifergröße Die Angreifergröße analysiert das Umfeld und dessen Anzahl genauer. So geht, bezogen auf die Anzahl der Angreifer, die geringste Gefahr von Entwicklern (2) aus. Die größte Bedrohung stellen aufgrund ihrer Masse die anonymen Internetnutzer (9) dar. Eine weitere Verfeinerung findet in Intranetnutzer (4), Partner (5) oder authentifizierte Nutzer (6) statt.

Schwachstelle

Die Analyse der Schwachstelle erfolgt in den Bereichen *Entdeckung*, *Ausnutzung*, *Bekanntheit* und *Erkennung*. Um den Besonderheiten eines Sicherheitsmodells für Smartphones nachzukommen, werden zusätzlich die Merkmale *Angriffsvektor*, *Nutzerinteraktion*, *Umfang* und *Transparenz* eingeführt.

Entdeckung Die Entdeckung der Schwachstelle wird unterteilt in praktisch unmöglich (1), schwierig (3), leicht (7) und automatisierte Tools (9).

Ausnutzung Diese Kategorie bewertet die Schwierigkeit der Ausnutzung einer Schwachstelle und wird eingeteilt in theoretisch (1), schwierig (3), leicht (5) oder automatisierte Tools (9).

Bekanntheit Die Bekanntheit der Schwachstelle wird eingestuft in unbekannt (1), versteckt (4), offensichtlich (6) und öffentlich (9).

Erkennung Die Erkennung eines Angriffs wird unterteilt in die Stufen aktive Erkennung (1), der Angriff wird protokolliert und es erfolgt Mitteilung (3), der Angriff wird protokolliert und es erfolgt keine Mitteilung (7) und der Angriff läuft unprotokolliert (9) ab.

Angriffsvektor Der *Angriffsvektor* beeinflusst den Schweregrad einer Schwachstelle im mobilen Bereich im Gegensatz zu klassischen Risikobewertungen, da Smartphones aufgrund ihrer Architektur über Schnittstellen (z.B. Bluetooth, WLAN, NFC) verfügen, welche eine Kommunikation mit anderen Geräten oder Netzwerken erlauben. Diese Schnittstellen existieren bei klassischen Bedrohungsbewertungen eines Servers in aller Regel nicht. Es findet eine Unterscheidung in physikalisch (1), lokal (4), angrenzend (6) oder Netzwerk (9) statt.

Nutzerinteraktion Die Berücksichtigung der *Nutzerinteraktion* ermöglicht eine Bewertung in der Erfordernis eines Social Engineering Angriff, durch beispielsweise Versand einer Nachricht mit einem präparierten Link, zur Ausnutzung der Schwachstelle oder ob der Empfang einer manipulierten Datei beziehungsweise Nachricht ausreicht. Diese Einschätzung kann mit erforderlich (3) oder nicht erforderlich (7) bewertet werden.

Umfang Die Eigenschaft *Umfang* beschreibt, inwiefern die Schwachstelle Auswirkungen auf die betroffene Komponente hat oder ob durch diese Schwachstelle andere Komponenten korrumpiert werden können. Diese Bewertung wird in lokal (3) oder übergreifend (7) eingeteilt.

Transparenz Der Wert *Transparenz* wird neben der zur Verfügung gestellten Dokumentation des verwendeten IT-Verfahrens (siehe Definition in Kapitel 4.2.1) um die Eigenschaft der Bewertung des Herstellers im Umgang mit der Bereitstellung von Informationen zu einer Schwachstelle erweitert. Dies lässt Rückschlüsse zu, ob der Hersteller Priorität in die Sicherheit seiner Geräte setzt. Dieser Wert wird unterteilt in Sourcecode veröffentlicht und dokumentiert (1), Dokumentation mit Hintergrundinformationen (3), Dokumentation ohne Hintergrundinformationen (4) oder keine Information (6).

Die nachfolgende Tabelle zeigt eine Beispielauswertung der Eintrittswahrscheinlichkeit einer Schwachstelle:

Bewertung:

HW.02	Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Angreifer	
Fähigkeiten:	6 - Netzwerk- und Programmierfähigkeiten
Motivation:	4 - Mittelmäßige Motivation
Aufwand:	4 - Spezialrechte erforderlich
Angreifergröße:	9 - Anonyme (Internet-) Nutzer
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	6 - Angrenzend
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer:	5.75 - Mittel
Schwachstelle:	6.25 - Hoch
Gesamt:	5.75 - Mittel

4.2.7.2 Auswirkungen

Der Part *Auswirkungen* bewertet das Ausmaß der Verletzungen der in Kapitel 4.2.1 vorgestellten Werte. Ausgehend von der Identifikation der Werte in Kapitel 4.2.1 wird die Risikobewertung zusätzlich um die Kategorie *Intervenierbarkeit* erweitert.

Technische Auswirkungen

Dieser Abschnitt erläutert die Einstufungen hinsichtlich der technischen Auswirkungen einer Schwachstelle.

Vertraulichkeit Die Vertraulichkeit der Daten wird unterteilt in eine Verletzung von keinen Daten (0), wenigen nicht-sensitiven Daten (2), umfangreiche nicht-sensitive Daten (5), wenige kritische Daten (6), umfangreiche kritische Daten (7) oder alle Daten (9). Sensitive Daten umfassen alle personenbezogenen Daten und kritische Daten erweitern sensitive Daten um einen noch höheren Geheimhaltungsgrad, wie beispielsweise Passwörter.

Integrität Die Integrität von Daten wird eingeteilt in geringfügige Manipulationen (1), kleine schwerwiegende Manipulationen (3), umfangreiche kleinere Manipulationen (5), umfangreiche schwerwiegende Manipulationen (7) oder die komplette (Daten-) Manipulation (9).

Verfügbarkeit Die Verfügbarkeit von Diensten ist kategorisiert in kleine untergeordnete Dienste (1), kleinere Hauptdienste unterbrochen (5), zahlreiche Hauptdienste unterbrochen (7) oder der Totalausfall aller Dienste (9).

Verantwortliche Die Kategorie Verantwortliche stuft die Chancen der Identifikation der Urheber eines Angriffs ein, in komplett identifizierbar (1), möglicherweise identifizierbar (7) oder anonym (9).

Intervenierbarkeit Der Wert *Intervenierbarkeit* beurteilt die vorhandenen Mechanismen des Gerätes infolge einer Angriffsdetektion nach einer Unterbindung von Angriffen und einer Gewährleistung des Datenschutzes. Eine Einteilung wird in die Kategorien vollständige Datenhoheit (0), teilweise Datenhoheit (3) oder keine Datenhoheit (7) getroffen.

Finanzielle Auswirkungen

Dieser Part beschäftigt sich mit den aus einer Ausnutzung der Schwachstelle entstehenden Konsequenzen in finanzieller Hinsicht.

Materieller Schaden Der materielle Schaden wird eingeteilt in geringer als Kosten zur Behebung der Schwachstelle (1), kleine Auswirkungen auf jährlichen Umsatz (3), signifikante Auswirkungen auf jährlichen Umsatz (7) oder Insolvenz (9).

Ansehen Der Schweregrad des Verlustes an Ansehen wird unterschieden in niedriger Ansichtsverlust (1), Verlust betroffener Kunden (4), Verlust von Großkunden (6), Verlust des Firmenrufs (7) oder Markenverlust (9).

Gesetzesverstoß Die Verletzung von Gesetzen ist kategorisiert in keine Verstöße (0), kleine Verstöße (2), einzelne Verstöße (5) oder öffentlichkeitswirksame Verstöße (7).

Privatsphäre Dieser Punkt liefert eine Einschätzung der Anzahl an Personen, deren Privatsphäre durch diese Schwachstelle verletzt wurde. Es erfolgt eine Abstufung in kein Betroffener (0), ein Betroffener (2), mehrere Betroffene (3), hunderte Betroffene (5), tausende Betroffene (7) oder Millionen Betroffene (9).

Um die Auswirkungen einordnen zu können, ist wie zu Beginn dieses Abschnitts bereits erwähnt, eine vorherige Definition der Nutzergruppe erforderlich. Die Beispieldefinition erfolgt später in der Anwendung des Modells.

Die nachfolgende Tabelle zeigt exemplarisch die Einstufung der finanziellen und technischen Auswirkungen.

Bewertung:

HW.02	**Geschäftsführer**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	9 - Anonym / Nicht identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	3.33 - Mittel

Behördliche Auswirkungen

Die Bewertung der *finanziellen Auswirkungen* richtet sich in erster Linie an Unternehmen jeglicher Größe. Nachdem das *OWASP Risk Rating* eine Erweiterung und Anpassung der Risikobewertung zulässt, wurde für Behörden ein eigenes Bewertungsprofil erstellt. Dieses Bewertungsprofil kann bei Behörden anstelle der Einschätzung der *finanziellen Auswirkungen* verwendet werden. Die Identifikation der Messgrößen der *behördlichen Auswirkungen* basiert im Wesentlichen auf die vom *Bundesamt für Sicherheit in der Informationstechnik* definierten *Schutzbedarfskategorien*.
[Sicc]

Verstöße gegen Gesetze Die Verletzung von Gesetzen ist kategorisiert in keine Verstöße (0), kleine Verstöße (2), einzelne Verstöße (5), oder öffentlichkeitswirksame Verstöße (7).[Sicc]

Informationelles Selbstbestimmungsrecht Die Beeinträchtigung des informationellen Selbstbestimmungsrecht ist eingeteilt in die Kategorien keine Nachteile für Betroffene (0), Betroffene könnten in deren gesellschaftlichen Stellung oder wirtschaftlichen Verhältnisse benachteiligt werden (3), es ist mit einer erheblichen Beeinträchtigung in der gesellschaftlichen Stellung oder der wirtschaftlichen Verhältnisse von Betroffenen zu rechnen (6) oder es besteht Gefahr für Leib und Leben oder die persönliche Freiheit der Betroffenen (9).[Sicc]

Persönliche Unversehrtheit Die persönliche Unversehrtheit der Betroffenen wird unterteilt in keine Gefahr (0), möglicherweise Beeinträchtigung (3), eine Beeinträchtigung kann nicht ausgeschlossen werden (6) oder es besteht Gefahr für Leib und Leben (9).[Sicc]

Beeinträchtigung der Aufgabenerfüllung Die Beeinträchtigung der Aufgabenerfüllung wird eingestuft in keine Beeinträchtigung (0), eine akzeptable Beeinträchtigung (3), eine nicht tolerierbare Beeinträchtigung Einzelner (6) oder eine nicht tolerierbare Beeinträchtigung Aller (9).[Sicc]

Negative Innen- oder Außenwirkung Die Wirkung im Innen- oder Außenbereich wird kategorisiert in kein Ansehensverlust (0), ein geringer / interner Ansehensverlust (3), ein breiter Ansehens- oder Vertrauensverlust (6) oder ein landes- bzw. bundesweiter Ansehens- und Vertrauensverlust (9).[Sicc]

Finanzielle Auswirkungen Die finanziellen Auswirkungen werden unterteilt in kein Schaden (0), ein Schaden kleiner 25.000 Euro (3), ein Schaden zwischen 25.000 und 2.5 Millionen Euro (6) oder ein Schaden größer 2.5 Millionen Euro.[Sicc]

4.2.7.3 Gesamtrisiko einer Bedrohung

Nachdem in der Risikobewertung alle Attribute mit Werten eingestuft wurden, können die Eintrittswahrscheinlichkeit und die Auswirkungen separat berechnet werden. Das Ergebnis wird jeweils als numerischer Wert angegeben, welcher mit den Kategorien *Info*, *Niedrig*, *Mittel*, *Hoch* und *Kritisch* abgebildet wird (siehe Abbildung 4.5). Die Kreuzung der beiden Ergebnisse bestimmt das Gesamtrisiko.

Gesamtrisikobewertung = Wahrscheinlichkeit x Auswirkungen				
Auswirkungen	*HOCH*	Mittel	Hoch	Kritisch
	MITTEL	Niedrig	Mittel	Hoch
	NIEDRIG	Info	Niedrig	Mittel
		NIEDRIG	*MITTEL*	*HOCH*
	Wahrscheinlichkeit			

Abbildung 4.5: Matrix des Gesamtrisikos

4.2.8 Gegenmaßnahmen planen

Der letzte Part des Prozesses zur Erstellung eines Risiko- und Bedrohungsmodells sieht die Ergreifung geeigneter Gegenmaßnahmen vor. Die Anforderung an das Sicherheitsmodell sieht jedoch eine Bewertung des Sicherheitslevels vor, wofür die Dokumentation von Gegenmaßnahmen nicht erforderlich ist. Stattdessen wird im letzten Schritt ein Bericht erstellt, der die Ergebnisse der Auswertung vorstellt und zusammenfasst.

4.3 Implementierung des Sicherheitsmodells

Nach Durchführung der verschiedenen Phasen ergibt sich das in Abbildung 4.6 dargestellte Sicherheitsmodell. Damit ist eine Gesamtbewertung des aktuellen Sicherheitslevels eines Smartphones darstell- und vergleichbar. Die Risikobewertung geschieht auf Basis der öffentlich recherchierbaren Schwachstellen von Komponenten eines Smartphones, welche für die Erfüllung der Anforderung zuständig sind.

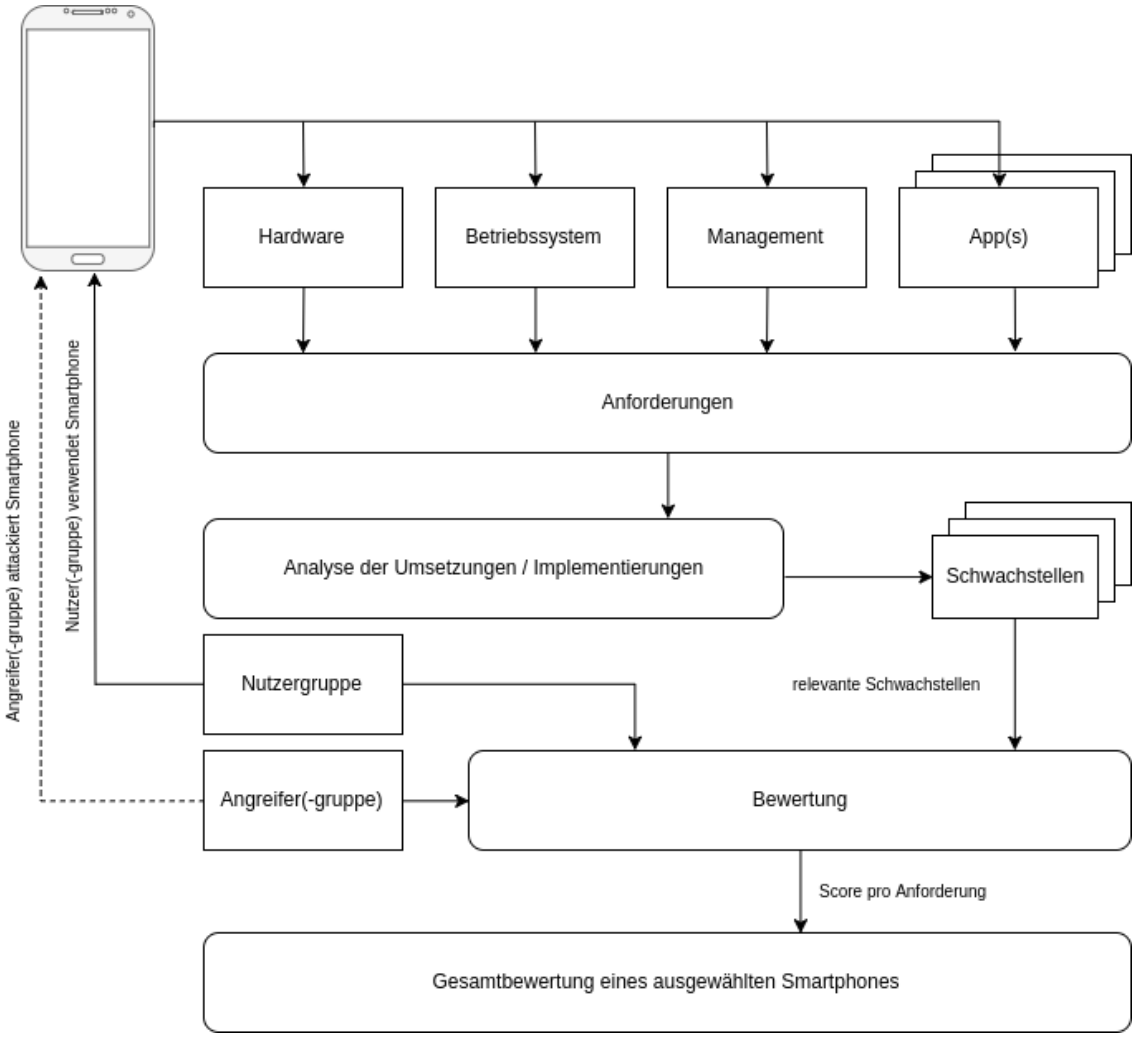

Abbildung 4.6: Implementierung des Sicherheitsmodells für Smartphones

Die Anwendung des Modells beginnt mit der Auswahl eines Smartphones und der Festlegung einer Nutzergruppe, welche dieses Smartphones verwendet. Anschließend kann das Modell angewendet werden. Die Sicherheitsanforderungen des Modells sind unterteilt in die Bereiche *Hardware*, *Betriebssystem*, *Management* und *Applikationen*. Jede dieser Anforderung (Insgesamt 41 Anforderungen) wird einer Analyse der technischen Umsetzung unterzogen. Auf Basis dieser Analyse findet im Verlauf eine Recherche nach öffentlich verfügbaren Schwachstellen statt. Die ermittelten und relevanten Schwachstellen werden einer Bewertung unterzogen. Diese berücksichtigt Aspekte der Ausnutzbarkeit der Schwachstelle, die technischen Auswirkungen, eine Bewertung des Angreifertyps und eine Einschätzung der finanziellen Auswirkungen in Bezug auf die Nutzergruppe. Nach der

Kalkulation kann eine Einteilung des Befundes in *Information*, *Niedrig*, *Mittel*, *Hoch* oder *Kritisch* vorgenommen werden.

4.4 Anwendung des Sicherheitsmodells

In diesem Abschnitt wird das Sicherheitsmodell für drei unterschiedliche Nutzergruppen mit jeweils zwei Smartphones angewendet. Zur besseren Vergleichbarkeit werden zwei Smartphones unterschiedlicher Hersteller mit verschiedenen Betriebssystemen ausgewählt.

4.4.1 Auswahl der Smartphones

Als Testobjekte werden in dieser Arbeit für die Plattformen Android und iOS nachfolgende Geräte verwendet. Um eine faire Bewertung des aktuellen Sicherheitslevels zu erhalten, fällt die Wahl auf zwei im gleichen Jahr erschienenen Smartphones.

1	Smartphone 1
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Erscheinungsdatum:	16. September 2016

2	Smartphone 2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Erscheinungsdatum:	10. März 2016

4.4.2 Unternehmen 1 - Geschäftsführer

In diesem Abschnitt wird das Sicherheitsmodell auf die *Beispielgruppe Geschäftsführer* angewendet um das Sicherheitslevel für die oben ausgewählten Smartphones zu bestimmen. Diese Beispielgruppe definiert einen Geschäftsführer einer Firma mit rund 25 Angestellten, welcher ein Gerät für geschäftliche Zwecke nutzen möchte.

Es sollen folgende Diensten genutzt werden:

- Lesen und Schreiben von (verschlüsselten) geschäftlichen E-Mails
- Zugriff auf Personalakten der Mitarbeiter über proprietäre Applikation

Die Nutzung der oben genannten Dienste ermöglicht einem Angreifer den Empfang und Versand von vertraulichen E-Mails. Der Inhalt aller Nachrichten wird verschlüsselt und erhöht damit den Aufwand für einen Angreifer den privaten Schlüssel zu extrahieren. Als weitere Bedrohung wird die Verletzung der Privatsphäre der Firmenangestellten betrachtet,

die zu einem Verstoß des Bundesdatenschutzgesetzes führt.

Die *maximalen* finanziellen Auswirkungen für diese Anwendergruppe durch Ausnutzen einer Schwachstelle, welche eine Korrumpierung des Gerätes ermöglicht, werden wie folgt bewertet:

1	Geschäftsführer
Finanzielle Auswirkungen	
Materieller Schaden:	3 - Kleine Auswirkungen auf jährlichen Umsatz
Ansehen:	4 - Verlust betroffener Kunden
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Auswirkungen	
Finanzielle:	3.00 - Mittel

Damit ergibt sich ein numerischer Durchschnittswert von 3.00 für die Bewertung der maximalen finanziellen Auswirkungen. Diese Auswirkungen fallen in die Kategorie *mittel*.

Hinsichtlich der Beurteilung potentieller Angreifer dieser Anwendergruppe ergibt sich folgende Einschätzung:

1	Geschäftsführer
Angreifer	
Fähigkeiten:	6 - Netzwerk- und Programmierfähigkeiten
Motivation:	4 - Mittelmäßige Motivation
Aufwand:	4 - Spezialrechte erforderlich
Angreifergröße:	9 - Anonyme (Internet-) Nutzer
Eintrittswahrscheinlichkeit	
Angreifer:	5.75 - Mittel

Das Ergebnis der Einschätzung potentieller Angreifer (numerischer Durchschnittswert: 5.75) mit einer *mittleren* Bedrohung, deckt sich mit der Beurteilung des Bundesamts für Sicherheit in der Informationstechnik für kleine bis mittelständische Unternehmen (siehe Überschrift *Angreifer* in Kapitel 4.2.7).

Der Aspekt *Angreifer* wird generisch für alle Bewertungen von Schwachstellen für diese Anwendergruppe verwendet. Die Beurteilung der Bereiche *finanzielle Auswirkungen*, *tech-*

nische Auswirkungen und das Ausmaß der *Schwachstelle* findet individuell im Anhang B und C dieser Arbeit statt.

4.4.2.1 Exemplarische Bewertung einer Anforderung

Zur Demonstration des Modells wird eine Anforderung exemplarisch eingefügt. Um den Lesefluss nicht zu beeinträchtigen, werden alle weiteren Anforderungen, die Analyse der Umsetzung und die Bewertung vorhandener Schwachstellen, in den Anhang B (Firma 1) und C ausgelagert.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Beschreibung: Drahtlose Schnittstellen (Bluetooth / WLAN) sollen aktuelle Standards zur sicheren Übertragung von Daten verwenden.

Allgemeine Hintergrundinformationen: Der Bedarf an WLAN Hotspots um das Smartphone mit dem Internet zu verbinden, kann allein durch die riesige Anzahl an angebotenen Hotspots an öffentlichen Plätzen begründet werden. Durch die Nutzung eines öffentlichen Hotspots mit dem Smartphone ergeben sich multiple Bedrohungen für die Sicherheit des Gerätes. Obwohl die Kommunikation zwischen Smartphone und Access Point auf Ebene 1 (Bitübertragungsschicht) verschlüsselt abläuft, können auf Ebene 3 (Netzwerkschicht) unverschlüsselte IP-Pakete abgehört werden. Hierzu sind weitere Sicherungsmaßnahmen erforderlich, beispielsweise die Etablierung einer VPN-Verbindung.

Umsetzung iPhone 7: Der jüngste WLAN Standard 802.11ah (siehe Kapitel 2.3) wird nicht unterstützt. Das iPhone 7 unterstützt den Standard 802.11ac[Incn]. Zur Verschlüsselung auf Ebene 1 werden alle aktuellen Standards (2017) unterstützt.

Schwachstelle: Die als *Broadpwn* vorgestellte Schwachstelle im Treiber des Chipherstellers Broadcom, ermöglicht einem Angreifer, durch ein modifiziertes WLAN Paket, Schadcode mit Systemrechten auszuführen. Diese Lücke wurde in Version 10.3.3 behoben[1].
Am 27.09.2017 wurde von einem Forscher des Project Zero (Google) ein Exploit veröffentlicht, welcher abermals eine Schwachstelle in der Broadcom-Firmware ausnutzt. Damit ist die volle Kontrolle des WLAN-Moduls möglich[2][Zer].

Angriffsszenario: Durch die Konfiguration des Exploits mit den erforderlichen Parametern, ist ein Angreifer in der Lage, Schadcode (Payload) mit Systemrechten auszuführen. Damit ist eine Kompromittierung aller auf dem Gerät gespeicherten Daten möglich. Inwiefern ein Angreifer Informationen aus der Keychain des Gerätes extrahieren kann, konnte nicht recherchiert werden. Im AppStore Cydia[3] existieren Tools, welche auf Geräten mit Root-Rechten die komplette Keychain auslesen können. Deswegen kann angenommen werden, dass durch Nutzung dieses Exploits sensible Daten aus der Keychain entschlüsselt werden können. Abhängig von der gewählten Datensicherheitsklasse (siehe Kapitel 2.6.4) können Daten der Klasse C und D extrahiert werden.

[1] https://www.heise.de/mac-and-i/meldung/Broadpwn-Kritische-iPhone-WLAN-Schwachstelle-gestopft-3779461.html

[2] https://www.heise.de/mac-and-i/meldung/iPhone-7-Exploit-fuer-kritische-WLAN-Luecke-veroeffentlicht-3845431.html

[3] Cydia ist ein inoffizieller App Store, welcher auf gejailbreakten Gerät installiert werden kann und diverse Tools zur Umgehung von Sicherheitsmechanismen des Betriebssystems enthält.

Bewertung der Eintrittswahrscheinlichkeit:

HW.02	**Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	6 - Angrenzend
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer:	5.75 - Mittel
Schwachstelle:	6.25 - Hoch
Gesamt:	6.00 - Hoch

Bewertung der Auswirkungen:

HW.02	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	3.33 - Mittel

Gesamtrisiko der Anforderung R-HW.02 :

Die Eintrittswahrscheinlichkeit wird mit 6.00 als hoch angesehen. Zusammen mit den mittleren Auswirkungen (numerischer Durchschnittswert: 3.33) ergibt sich ein hohes Gesamtrisiko. Welche Konsequenzen diese Auswirkungen auf die Gesamtbewertung des Sicherheitslevels für das iPhone 7 hat, wird im Fazit des Abschnitts 4.4.2.2 erläutert.

Umsetzung Samsung Galaxy S7: Die populäre Schwachstelle *Broadpwn (CVE-2017-9417)* betraf ebenfalls diverse Android-Geräte (u.a. Samsung Galaxy S7). Mit dem Sicherheitsupdate vom 01.08.2017 wurde diese Lücke von Samsung durch ein Update am 30.08.2017 für dieses Gerät geschlossen.
Im September 2017 tauchten weitere kritische Schwachstellen in der Firmware von Broadcom für WLAN Module auf, welche das Einschleusen und Ausführen von Schadcode ermöglichen (CVE-2017-11120, CVE-2017-11121, CVE-2017-7065)[Prog].
Am 13. September veröffentlichten Sicherheitsforscher der Firma Armis unter dem Titel *BlueBorne* eine Reihe an Schwachstellen in der Implementierung von Bluetooth-Schnittstellen, welche unter Android eine Ausführung von Schadcode mit Systemrechten ermöglicht.[SV17]

Angriffsszenario: Für die Schwachstelle *BlueBorne* existiert ein öffentlich verfügbarer Exploit der die Ausführung von Schadcode mit privilegierten Rechten ermöglicht. Damit ist ein Angreifer in der Lage alle gespeicherten Daten auf dem Smartphone zu korrumpieren. Die Entschlüsselung von Daten, welche mittels des Keystores verschlüsselt sind, ist nach jetzigem Stand nicht möglich.

Bewertung der Eintrittswahrscheinlichkeit:

HW.02	**Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	6 - Angrenzend
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer:	5.75 - Mittel
Schwachstelle:	6.25 - Hoch
Gesamt:	6.00 - Hoch

Bewertung der Auswirkungen:

HW.02	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065 (Broadcom), CVE-2017-0781 (BlueBorne)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	3.33 - Mittel

Gesamtrisiko der Anforderung R-HW.02 :

Die Eintrittswahrscheinlichkeit wird mit 6.00 als hoch angesehen. Zusammen mit den mittleren Auswirkungen (numerischer Durchschnittswert: 3.33) ergibt sich ein hohes Gesamtrisiko. Welche Konsequenzen diese Auswirkungen auf die Gesamtbewertung des Sicherheitslevels für das Samsung Galaxy S7 hat, wird im Fazit des Abschnitts 4.4.2.2 erläutert.

4.4.2.2 Auswertung

Dieser Abschnitt stellt die Bewertungen der einzelnen Anforderungen übersichtlich dar und ermöglicht einen schnellen Vergleich der beiden Smartphones.
Zur Erleichterung des Leseflusses folgt vorab erneut eine Legende der Bewertungsfarben:

Farbkategorie	Beschreibung
	Anforderung wird nicht umgesetzt.
	Anforderung wird umgesetzt.
	Es konnte keine öffentliche Schwachstelle recherchiert werden.
	Das Risiko wird sehr niedrig / theoretisch eingestuft.
	Das Risiko wird niedrig eingestuft.
	Das Risiko wird mittel eingestuft.
	Das Risiko wird hoch eingestuft.
	Das Risiko wird kritisch eingestuft.

Die Kombination von Eintrittswahrscheinlichkeit und Auswirkungen werden in Abbildung 4.5 erläutert.

	iPhone 7			Samsung Galaxy S7		
Anforderungen	Schwachstelle	Auswirkungen	Gesamt	Schwachstelle	Auswirkungen	Gesamt
HW.01	-			-		
HW.02	6.00	3.33	hoch	6.00	3.33	hoch
HW.03						
HW.03.1	-					info
HW.03.2	-			5.13	4.22	mittel
HW.03.3	-			-		
HW.04	-			-		
OS.01	5.56	4.44	mittel	4.31	2.89	niedrig
OS.02	4.63	1.44	niedrig	4.63	1.44	niedrig
OS.03	-			5.63	2.22	niedrig
OS.04						

OS.05			info			
OS.06	4.88	1.67	niedrig	4.88	3.00	mittel
OS.07	-			-		
OS.08	-			4.88	3.00	mittel
OS.09	-			-		
OS.10	-			-		
OS.11						
OS.12	-			-		
OS.13						
OS.14				-		
MGMT.01	-			-		
MGMT.02						
MGMT.03	4.63	1.56	niedrig	4.88	3.00	mittel
MGMT.04						
MGMT.05	4.56	0.89	niedrig			info
MGMT.06	-			-		
MGMT.07	-					info
MGMT.08			info	-		
MGMT.09			info	-		
MGMT.10	5.19	3.67	mittel	-		
MGMT.11						
MGMT.12	-			-		
MGMT.13	-			-		
MGMT.14	-					info
MGMT.15	4.25	2.22	niedrig	4.25	2.11	niedrig
MGMT.16			info			
MGMT.17						
MGMT.18			info			
MGMT.19						
MGMT.20			info			

Fazit iPhone 7:

Das iPhone 7 besitzt aktuell (18. September 2017) eine hohe Gefährdung (R-HW.02) durch eine Schwachstelle im WLAN Treiber. Zur Vermeidung eines erfolgreichen Angriffs wird die Aktivierung des WLAN Moduls derzeit nicht empfohlen. Sollte sich dieses nicht vermeiden lassen, so ist darauf zu achten, dass eine Aktivierung des Moduls nicht an öffentlichen Plätzen geschieht. Daneben existieren zwei weitere Gefährdungen mit einem mittleren Risiko und fünf mit einem niedrigen Gesamtrisiko. Die Gefährdung durch die Schwachstelle im Lightning-Anschluss (R-OS.01) kann durch Verwendung eines ausreichend langen alphanumerischen Passwortes als Geräteschutz auf ein Minimum reduziert werden. Die Schwachstelle (R-MGMT.10), welche eine Übernahme eines verwalteten Gerätes zu einem bösartigen Server durch Aufruf einer Konfigurations-URL ermöglicht, kann durch eine entsprechende Sensibilisierung der Mitarbeiter / des Geschäftsführers minimiert werden.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen des Sicherheitsmodells liegen daran, dass Apple seinen Quelltext für das Betriebssystem nicht veröffentlicht und damit eine Bewertung der Sicherheit nur aufgrund der zur Verfügung gestellten Dokumentation durchgeführt werden kann.

Fazit Samsung Galaxy S7:

Das Samsung Galaxy S7 besitzt aktuell (19. September 2017) eine hohe Gefährdung (R-HW.02) durch eine Schwachstelle im WLAN-Treiber von Broadcom. Zur Minimierung eines erfolgreichen Angriffs wird die Aktivierung des WLAN Moduls derzeit nicht empfohlen. Sollte sich dieses nicht vermeiden lassen, so ist darauf zu achten, dass eine Aktivierung des Moduls nicht an öffentlichen Plätzen erfolgt. Daneben existieren vier weitere Gefährdungen mit einem mittleren Risiko und vier mit einem niedrigen Risiko. Die erste Schwachstelle in der ARM-Architektur, welche die Extraktion von privaten Schlüsseln aus der Trustzone ermöglicht, setzt einen physikalischen Zugriff auf das Gerät und die Möglichkeit den Stromsparmodus zu aktivieren voraus. Die Schwachstelle in der Anwendung *SVoice*, welche das Ausführen von Schadcode mit privilegierten Rechten ermöglicht, kann durch eine Deaktivierung des Sprachassistenten vermieden werden. Die Schwachstelle im MediaFramework kann durch einen Nichtaufruf von Musikdateien von nicht vertrauenswürdigen Absendern vermieden werden.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen liegen in einem Fehlen eines dedizierten Kryptographiemoduls und der fehlenden nativen Unterstützung von Mailverschlüsselung durch gängige Verfahren (S/MIME oder PGP). Letzteres lässt sich durch die Installation einer Drittanbieter Anwendung lösen.

4.4.3 Unternehmen 1 - Mitarbeiter

Die *Beispielgruppe Mitarbeiter* beurteilt das Sicherheitslevel eines Smartphones, welches von einem Mitarbeiter - ohne Führungsverantwortung - einer Firma mit rund 25 Mitarbeiter für berufliche Zwecke genutzt wird.

Die Geräte sind mit folgenden Diensten verknüpft:

- Lesen und Schreiben von (verschlüsselten) geschäftlichen E-Mails

Die Nutzung der oben genannten Dienste ermöglichen einen Angreifer den Empfang und Versand von vertraulichen E-Mails. Der Inhalt aller Nachrichten wird verschlüsselt und erhöht damit den Aufwand für einen Angreifer den privaten Schlüssel zu extrahieren.

Die maximalen finanziellen Auswirkungen für diese Anwendergruppe durch Ausnutzen einer Schwachstelle, welche eine Korrumpierung des Gerätes ermöglicht, werden wie folgt bewertet:

2 Mitarbeiter	
Finanzielle Auswirkungen	
Materieller Schaden:	3 - Kleine Auswirkungen auf jährlichen Umsatz
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Auswirkungen	
Finanzielle:	1.50 - Niedrig

Damit ergibt sich ein numerischer Durchschnittswert von 1.50 für die Bewertung der finanziellen Auswirkungen. Diese Auswirkungen fallen in die Kategorie *niedrig*.

Hinsichtlich der Beurteilung potentieller Angreifer dieser Anwendergruppe ergibt sich folgende Einschätzung:

2 Mitarbeiter	
Angreifer	
Fähigkeiten:	6 - Netzwerk- und Programmierfähigkeiten
Motivation:	4 - Mittelmäßige Motivation
Aufwand:	4 - Spezialrechte erforderlich
Angreifergröße:	9 - Anonyme (Internet-) Nutzer
Eintrittswahrscheinlichkeit	
Angreifer:	5.75 - Mittel

Die Bewertung des potentiellen Angreifers ergibt damit einen numerischen Durchschnittswert von 5.75, was eine *mittlere* Bedrohung darstellt.

Der Aspekt *Angreifer* wird generisch für alle Bewertungen von Schwachstellen für diese Anwendergruppe verwendet. Die Beurteilung der Bereiche *finanzielle Auswirkungen*, *technische Auswirkungen* und die *Schwachstelle* findet individuell im Anhang B und D dieser Arbeit statt.

4.4.3.1 Exemplarische Bewertung einer Anforderung

Eine Demonstration der Analyse einer Anforderung und der Bewertung einer Schwachstelle wurde in der Beispielgruppe *Geschäftsführer* bereits erläutert. Die Einschätzung der Eintrittswahrscheinlichkeit einer Bedrohung durch eine Schwachstelle im Smartphone ist abhängig vom Unternehmen, welches für die Beispielgruppen *Geschäftsführer* und *Mitarbeiter* identisch ist. Damit können die Bewertungen der Schwachstellen im Anhang B für diese Gruppe (Firma 1) ohne Anpassungen übernommen werden. Die Bewertung der technischen und finanziellen Auswirkungen erfolgt separat in Anhang D dieser Arbeit.

4.4.3.2 Auswertung

Dieser Abschnitt stellt die Bewertungen der einzelnen Anforderungen übersichtlich dar und ermöglicht einen schnellen Vergleich der beiden Smartphones.

Zur Erleichterung des Leseflusses folgt vorab erneut eine Legende der Bewertungsfarben:

Farbkategorie	Beschreibung
	Anforderung wird nicht umgesetzt.
	Anforderung wird umgesetzt.
	Es konnte keine öffentliche Schwachstelle recherchiert werden.
	Das Risiko wird sehr niedrig / theoretisch eingestuft.
	Das Risiko wird niedrig eingestuft.

Das Risiko wird mittel eingestuft.

Das Risiko wird hoch eingestuft.

Das Risiko wird kritisch eingestuft.

Die Kombination von Eintrittswahrscheinlichkeit und Auswirkungen werden in Abbildung 4.5 erläutert.

	iPhone 7			Samsung Galaxy S7		
Anforderungen	Schwachstelle	Auswirkungen	Gesamt	Schwachstelle	Auswirkungen	Gesamt
HW.01	-			-		
HW.02	6.00	2.89	mittel	6.00	2.89	mittel
HW.03						
HW.03.1	-					info
HW.03.2	-			5.13	3.56	mittel
HW.03.3	-			-		
HW.04	-			-		
OS.01	5.56	4.00	mittel	4.31	2.44	niedrig
OS.02	4.63	1.33	niedrig	4.63	1.44	niedrig
OS.03	-			5.63	2.22	niedrig
OS.04						
OS.05			info			
OS.06	4.88	1.67	niedrig	4.88	2.78	niedrig
OS.07	-			-		
OS.08	-			4.88	2.78	niedrig
OS.09	-			-		
OS.10	-			-		
OS.11						
OS.12	-			-		
OS.13						

OS.14				-		
MGMT.01	-			-		
MGMT.02						
MGMT.03	4.63	1.56	niedrig	4.88	2.78	niedrig
MGMT.04						
MGMT.05	4.56	0.89	niedrig			info
MGMT.06	-			-		
MGMT.07	-					info
MGMT.08			info	-		
MGMT.09			info	-		
MGMT.10	5.19	3.11	mittel	-		
MGMT.11						
MGMT.12	-			-		
MGMT.13	-			-		
MGMT.14	-					info
MGMT.15	4.25	2.22	niedrig	4.25	2.11	niedrig
MGMT.16			info			
MGMT.17						
MGMT.18			info			
MGMT.19						
MGMT.20			info			

Fazit iPhone 7:

Das iPhone 7 besitzt aktuell (18. September 2017) für die Nutzergruppe Mitarbeiter keine kritischen oder hohen Gefährdungen. Es bestehen insgesamt drei mittlere und fünf niedrige Gefährdungen. Ein Angriff durch Ausnutzen der Schwachstelle im WLAN Modul wird durch Deaktivierung vermieden. Sollte sich dieses nicht vermeiden lassen, so ist darauf zu achten, dass eine Aktivierung des Moduls nicht an öffentlichen Plätzen geschieht. Die Gefährdung durch die Schwachstelle im Lightning-Anschluss (R-OS.01) kann durch Verwendung eines ausreichend langen alphanumerischen Passwortes als Geräteschutz auf ein Minimum reduziert werden. Die Schwachstelle (R-MGMT.10), welche eine Übernahme eines verwalteten Gerätes zu einem bösartigen Server durch Aufruf einer Konfigurations-URL ermöglicht, kann durch eine entsprechende Sensibilisierung der

Mitarbeiter erreicht werden.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen des Sicherheitsmodells liegen daran, dass Apple seinen Quelltext für das Betriebssystem nicht veröffentlicht und damit eine Bewertung der Sicherheit nur aufgrund der zur Verfügung gestellten Dokumentation durchgeführt werden kann.

Fazit Samsung Galaxy S7:

Das Samsung Galaxy S7 besitzt aktuell (19. September 2017) für die Nutzergruppe Mitarbeiter keine kritischen oder hohen Gefährdungen. Es bestehen insgesamt zwei mittlere und sieben niedrige Gefährdungen. Ein Angriff durch Ausnutzen der Schwachstelle im WLAN Modul wird durch Deaktivierung vermieden. Sollte sich dieses nicht vermeiden lassen, so ist darauf zu achten, dass eine Aktivierung des Moduls nicht an öffentlichen Plätzen geschieht. Die Schwachstelle in der ARM-Architektur, welche die Extraktion von privaten Schlüsseln aus der Trustzone ermöglicht, setzt einen physikalischen Zugriff auf das Gerät und die Möglichkeit den Stromsparmodus zu aktivieren voraus. Die Schwachstelle in der Anwendung *SVoice*, welche das Ausführen von Schadcode mit privilegierten Rechten ermöglicht, kann durch eine Deaktivierung des Sprachassistenten vermieden werden. Die Schwachstelle im MediaFramework kann durch einen Nichtaufruf von Musikdateien von nicht vertrauenswürdigen Absendern vermieden werden.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen liegen in einem Fehlen eines dedizierten Kryptographiemoduls und der fehlenden nativen Unterstützung von Mailverschlüsselung durch gängige Verfahren (S/MIME oder PGP). Letzteres lässt sich durch die Installation einer Drittanbieter Anwendung lösen.

4.4.4 Unternehmen 2 - Konzernchef

Die *Beispielgruppe Konzernchef* beurteilt das Sicherheitslevel eines Smartphones, welches von einem Konzernchef eines Unternehmens mit einer Größe von 10.000 Mitarbeitern für berufliche Zwecke genutzt wird.

Das Gerät ist mit folgenden Diensten verknüpft:

- Lesen und Schreiben von (verschlüsselten) geschäftlichen E-Mails
- Zugriff auf interne Webanwendungen via VPN Verbindung
- Möglichkeit der Freigabe von Geldbeträgen bis 100.000 Euro über mobile Anwendung

Die Nutzung der oben genannten Dienste ermöglichen einem Angreifer den Empfang und Versand von vertraulichen E-Mails. Der Inhalt aller Nachrichten wird verschlüsselt und erhöht damit den Aufwand für einen Angreifer den privaten Schlüssel zu extrahieren. Durch das Eindringen in die internen Webanwendungen kann eine Vielzahl an vertraulichen Informationen abgegriffen werden. Ebenso können Rechnungen bis zu einem Gesamtbetrag von 100.000 Euro freigegeben werden.

Die maximalen finanziellen Auswirkungen für diese Anwendergruppe durch Ausnutzen einer Schwachstelle, welche eine Korrumpierung des Gerätes ermöglicht, werden wie folgt bewertet:

2 Mitarbeiter	
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	6 - Verlust von Großkunden
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Auswirkungen	
Finanzielle:	6.75 - Hoch

Damit ergibt sich ein numerischer Durchschnittswert von 6.75 für die Bewertung der finanziellen Auswirkungen. Diese Auswirkungen fallen in die Kategorie *hoch*.

Hinsichtlich der Beurteilung potentieller Angreifer dieser Anwendergruppe ergibt sich folgende Einschätzung:

3 Konzernchef	
Angreifer	
Fähigkeiten:	6 - Netzwerk- und Programmierfähigkeiten
Motivation:	4 - Mittelmäßige Motivation
Aufwand:	4 - Spezialrechte erforderlich
Angreifergröße:	9 - Anonyme (Internet-) Nutzer
Eintrittswahrscheinlichkeit	
Angreifer:	7.75 - Hoch

Die Bewertung des potentiellen Angreifers ergibt damit einen numerischen Durchschnittswert von 7.75, was eine *hohe* Bedrohung darstellt.

4.4.4.1 Exemplarische Bewertung einer Anforderung

Der Aspekt *Angreifer* wird generisch für alle Bewertungen von Schwachstellen für diese Anwendergruppe verwendet. Die Beurteilung der Bereiche *finanzielle Auswirkungen*, *technische Auswirkungen* und die *Schwachstelle* findet individuell im Anhang B (Firma 2) und E dieser Arbeit statt.

4.4.4.2 Auswertung

Dieser Abschnitt stellt die Bewertungen der einzelnen Anforderungen übersichtlich dar und ermöglicht einen schnellen Vergleich der beiden Smartphones.

Zur Erleichterung des Leseflusses folgt vorab erneut eine Legende der Bewertungsfarben:

Farbkategorie	Beschreibung
	Anforderung wird nicht umgesetzt.
	Anforderung wird umgesetzt.
	Es konnte keine öffentliche Schwachstelle recherchiert werden.
	Das Risiko wird sehr niedrig / theoretisch eingestuft.
	Das Risiko wird niedrig eingestuft.
	Das Risiko wird mittel eingestuft.
	Das Risiko wird hoch eingestuft.
	Das Risiko wird kritisch eingestuft.

Die Kombination von Eintrittswahrscheinlichkeit und Auswirkungen werden in Abbildung 4.5 erläutert.

	iPhone 7			Samsung Galaxy S7		
Anforderungen	Schwachstelle	Auswirkungen	Gesamt	Schwachstelle	Auswirkungen	Gesamt
HW.01	-			-		
HW.02	7.00	5.22	hoch	7.00	5.22	hoch
HW.03						
HW.03.1	-					info
HW.03.2	-			6.13	5.89	hoch
HW.03.3	-			-		
HW.04	-			-		
OS.01	6.56	6.67	kritisch	5.31	3.56	mittel
OS.02	5.63	1.44	niedrig	5.63	1.44	niedrig
OS.03	-			6.63	2.22	mittel
OS.04						
OS.05			info			
OS.06	5.88	1.67	niedrig	5.88	5.33	mittel
OS.07	-			-		
OS.08	-			5.88	5.00	mittel
OS.09	-			-		
OS.10	-			-		
OS.11						
OS.12	-			-		
OS.13						
OS.14				-		
MGMT.01	-			-		
MGMT.02						
MGMT.03	5.63	1.56	niedrig	5.88	5.33	mittel
MGMT.04						
MGMT.05	5.56	1.11	niedrig			info

MGMT.06	-			-		
MGMT.07	-					info
MGMT.08			info	-		
MGMT.09			info	-		
MGMT.10	6.19	5.22	hoch	-		
MGMT.11						
MGMT.12	-			-		
MGMT.13	-			-		
MGMT.14	-					info
MGMT.15	5.25	2.33	niedrig	5.25	2.33	niedrig
MGMT.16			info			
MGMT.17						
MGMT.18			info			
MGMT.19						
MGMT.20			info			

Fazit iPhone 7:

Das iPhone 7 besitzt aktuell (18. September 2017) eine kritische Gefährdung durch die Schwachstelle im Lightning-Anschluss, da hierdurch eine komplette Entschlüsselung aller Daten durchgeführt werden kann. Diese Gefahr kann durch Verwendung eines ausreichend langen alphanumerischen Passwortes als Geräteschutz auf ein Minimum reduziert werden. Daneben existieren zwei hohe Gefährdungen. Diese sind in einer Schwachstelle des WLAN Moduls und der Verwaltung eines Gerätes durch einen bösartigen Managementserver verortet. Um die Schwachstelle im Treiber des WLAN Moduls zu vermeiden, wird die grundsätzliche Deaktivierung des WLAN Moduls empfohlen, bis ein Update durch den Hersteller vorliegt, welche diese Lücke behebt. Die Gefahr ein Gerät durch Aufruf einer Konfigurations-URL eines anderen (bösartigen) Managementserver einem Man in the Middle Angriff auszusetzen, kann derzeit nur durch eine entsprechende Schulung der Mitarbeiter / Geschäftsführers minimiert werden. Daneben existieren fünf weitere niedrige Gefährdungen.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen des Sicherheitsmodells liegen daran, dass Apple seinen Quelltext für das Betriebssystem nicht veröffentlicht und damit eine Bewertung der Sicherheit nur aufgrund der zur Verfügung gestellten Dokumentation durchgeführt werden kann.

Fazit Samsung Galaxy S7:

Das Samsung Galaxy S7 besitzt aktuell (19. September 2017) zwei hohe Gefährdung (R-HW.02).Um die Schwachstelle im Treiber des WLAN Moduls zu vermeiden, wird die

grundsätzliche Deaktivierung des WLAN Moduls empfohlen, bis ein Update durch den Hersteller vorliegt, welche diese Lücke behebt. Die Schwachstelle in der ARM-Architektur, welche die Extraktion von privaten Schlüsseln aus der Trustzone ermöglicht, setzt einen physikalischen Zugriff auf das Gerät und die Möglichkeit den Stromsparmodus zu aktivieren voraus. Eine Minimierung der Datenextraktion wird durch eine Verschlüsselung der Benutzerdateien, welche grundsätzlich aktiviert sein sollte, erreicht. Daneben existieren fünf mittlere Gefährdungen. Eine Verschlüsselung der Dateien verhindert auch das Auslesen von Daten durch ein Dumpen des Flashspeichers über die Debugging-Schnittstelle. Ein Downgrade der Betriebssystemversion und dem Ausnutzen von früheren Schwachstellen wird ebenfalls nur durch eine grundsätzliche Verschlüsselung der Benutzerdateien vermieden. Die Schwachstelle in der Anwendung *SVoice*, welche das Ausführen von Schadcode mit privilegierten Rechten ermöglicht, kann durch eine Deaktivierung des Sprachassistenten vermieden werden. Die Schwachstelle im MediaFramework kann durch einen Nichtaufruf von Musikdateien von nicht vertrauenswürdigen Absendern vermieden werden.
Die fehlenden Umsetzungen (schwarze Markierung) der Anforderungen liegen in einem Fehlen eines dedizierten Kryptographiemoduls und der fehlenden nativen Unterstützung von Mailverschlüsselung durch gängige Verfahren (S/MIME oder PGP). Letzteres lässt sich durch die Installation einer Drittanbieter Anwendung lösen.

5. Testen von mobilen Applikationen

Dieses Kapitel ermöglicht einen Einblick in das Testen von mobilen Anwendungen und liefert als Ergebnis ein systematisches Vorgehen eines Penetrationstests für eine mobile Anwendung. Zur Dokumentation eines Tests wird auf Grundlage der Anforderungen eine Checkliste erstellt.

5.1 Motivation

Die Motivation für die Durchführung eines Penetrationstest einer mobilen Anwendung kann verschiedene Gründe haben:

Die zunehmende Verwendung von mobilen Anwendungen in sicherheitskritischen Bereichen, bietet Angreifern die Möglichkeit Schwachstellen in einer Anwendung als neuen Angriffsvektor für die Kompromittierung der Kommunikation und lokal gespeicherter Daten auszunutzen oder die Sicherheit des Gesamtsystems Smartphone zu gefährden. Durch die Vielzahl an Schnittstellen und Kommunikationsmöglichkeiten mit der Umgebung (GPS, Kamera, Bluetooth, ...), erhöht sich das Interesse eines Angreifers Schwachstellen zu detektieren.

Des Weiteren existieren in vielen Großkonzernen interne Richtlinien, welche obligatorisch einen Penetrationstest einer mobilen Applikation vorsehen, gemäß dem Grundsatz, dass die Sicherheitskette nur so stark ist, wie ihr schwächstes Glied[Sic11].

5.2 Abgrenzung

Aus zeitlichen Gründen wird das Vorgehen eines Code-Reviews für mobile Anwendungen nicht thematisiert. Code-Reviews stellen zwar die beste Möglichkeit zur Identifikation aller Schwachstellen dar, erfordern jedoch einen deutlich höheren Zeitaufwand und sind unter Einbezug aller Bibliotheken oder vorgefertigten Implementierungen praktisch unmöglich durchzuführen. Zusätzlich scheuen viele Unternehmen die Herausgabe des Quelltextes oder diese wird durch interne Richtlinien untersagt.

In der Wirtschaft setzte sich für das Penetrationtesten von mobilen Anwendungen der *Gray-Box Test* durch. Bei einem *Gray-Box Test* wird die Binärdatei einer Anwendung und zusätzlich die Dokumentation über Schnittstellen und Funktionsweisen der Anwendung zur Verfügung gestellt.

5.3 Vorgehen

Die Grundlagen für die Entwicklung eines systematischen Vorgehens bei einem Penetrationstest von mobilen Anwendungen bilden der *OWASP Mobile Security Testing Guide* und der interne *OPTIMA Mobile Security Standard*.

Um den Aufbau einer mobilen Anwendung zu verstehen, ist eine Einführung in die Arten mobiler Apps erforderlich. Es bestehen derzeit drei Arten der Entwicklung von Anwendungen:

Native Applikation:

Eine native Anwendung wird in der von der Plattform unterstützten Programmiersprache (Android: Java, iOS: Swift oder Objective-C) implementiert. Die Funktionalitäten des Betriebssystems oder die Interaktion mit Hardware-Schnittstellen wird über vordefinierte Schnittstellen realisiert und können direkt verwendet werden. Der Nachteil dieser Anwendungen ist, dass für jede Plattform ein separater Quelltext geschrieben werden muss.
[Mo17]

Mobile Webapplikation:

Mobile Webanwendungen sind Webseiten, welche für den Aufruf auf mobile Endgeräte angepasst werden. Für gewöhnlich werden sie in HTML5 entwickelt. Mittlerweile existieren teilweise die Möglichkeiten der Interaktion mit (Hardware-)Schnittstellen (z.B. GPS, Kamera). Der Aufruf mobiler Webapplikationen erfolgt über eine URL. Die Vorteile dieses Anwendungstyps sind eine von der Plattform unabhängige Implementierung und eine direkte Veröffentlichung von Änderungen, da keine Überprüfung in den AppStores erforderlich ist.
[Mo17]

Hybride Applikation:

Dieser Anwendungstyp stellt eine Mischung der beiden vorangegangenen Typen dar. Die Darstellung und Logik von hybriden Applikationen wird über WebViews realisiert. Im Gegensatz zu mobilen Webanwendungen werden diese in einer eingebetteten WebView ausgeführt. Die Interaktion mit nativen Schnittstellen wird über Module des Frameworks für hybride Applikationen ermöglicht.
[Mo17]

5.4 Testbereiche

Ein Angreifer wird bei der Suche nach Schwachstellen einer Anwendung folgende Kernbereiche untersuchen:

- Lokale Datenspeicherung
- Kommunikation mit Backend
- Authentifizierung und Autorisierung
- Interaktion mit Plattform (Betriebssystem)

Diese Bereiche werden im *OWASP Mobile Security Testing Guide* als Hauptbereiche deklariert, für die zusätzliche Schutzmechanismen implementiert werden müssen. Die Autoren dieses Guides haben sich dazu entschieden, das umstrittene *Anti-Tampering* und *Anti-Reversing* als Anforderung an eine mobile Anwendung aufzunehmen, explizit jedoch mit dem Zusatz, dass es eine Ergänzung der Sicherheitsmaßnahmen darstellt und stattdessen auf andere Mechanismen nicht verzichtet werden darf.
[Comd]

5.4.1 Anforderungen

Durch Analyse der im vorangegangenen Kapitel 5.4 ermittelten Kernbereiche, lässt sich eine weitere Verfeinerung der Testbereiche durchführen:

- Allgemein
- Datenspeicherung
- Datenschutz und -sicherheit
- Kryptographie
- Authentifizierung
- Sessionverwaltung
- Netzwerkkommunikation
- Plattforminteraktion
- Anwendungsgenerierung
- Codequalität
- *Besonderheiten in Android*
- *Besonderheiten in iOS*

[Mue17]

Auf Basis des *OWASP Mobile Application Security Verification Standard (MASVS)* wurden zu jedem Testbereich konkrete Anforderungen identifiziert:

Allgemein

1. Die verwendeten Bibliotheken sind aktuell.
2. Das Backend verifiziert, dass die aktuellste Version der Anwendung installiert ist.
3. Sicherheitsmechanismen werden nicht clientseitig, sondern durch Backend erzwungen.

Datenspeicherung

1. Sensitive Daten werden lokal sicher gespeichert.
2. Sensitive Daten sind nicht in unverschlüsselten Backups enthalten.
3. Sensitive Daten werden nicht in Logdateien der Anwendungen geschrieben.

Datenschutz und -sicherheit

1. Das Cachen von Tastatureingaben ist in Textfeldern mit sensitiven Daten deaktiviert.
2. Die Verwendung der Zwischenablage ist bei Textfeldern mit sensitiven Inhalt nicht erlaubt.
3. Sensitive Informationen werden nicht über Interprozesskommunikation ausgetauscht.
4. Screenshots von Views mit sensitiven Daten sind nicht zugelassen.
5. Sensitive Daten werden bei Unterbrechung der Anwendung von Views entfernt.
6. Sensitive Daten werden nicht länger gespeichert als erforderlich.
7. Die Anwendung erfordert ein Minimum an Geräteschutz.

Kryptographie

1. Die Anwendung verwendet keine fest codierten Schlüssel zur symmetrischen Verschlüsselung.
2. Die Applikation nutzt erprobte Implementierungen von Verschlüsselungsalgorithmen.
3. Die Parameter der verwendeten Algorithmen sind nach Industriestandards konfiguriert.
4. Die Anwendung nutzt keine kryptographischen Algorithmen, welche veraltet oder unsicher sind.
5. Die Anwendung generiert ausreichend zufällige Zufallszahlen.

Authentifizierung

1. Die Anwendung nutzt eine akzeptable Form der Authentifizierung.
2. Das Backend erzwingt Passwortrichtlinie.
3. Das Backend sieht Mechanismen vor, um Nutzer bei einer Vielzahl an Fehlversuchen temporär zu sperren.
4. Die Verwendung von biometrischer Authentifizierung ist nicht eventbasiert, sondern entschlüsselt Informationen.
5. Das Backend implementiert und erzwingt einen zweiten Faktor zur Authentifizierung.
6. Transaktionen mit sensitiven Daten erfordern zusätzliche Authentifizierung.
7. Die Anwendung informiert Nutzer über alle Loginaktivitäten.

Sessionverwaltung

1. Das Backend verwendet zufällig generierte Token zur Authentifizierung von Clientanfragen, ohne das Mitsenden der Zugangsdaten eines Benutzers.
2. Das Backend invalidiert Token bei einem Logout des Nutzers.
3. Jeder Token wird durch das Backend nach einer vordefinierten Inaktivitätszeit für ungültig erklärt.

Netzwerkkommunikation

1. Die Übertragung von Daten findet durchgehend verschlüsselt statt und eine Verschlüsselung wird durch die Anwendung erzwungen.
2. Die Konfiguration der TLS Verbindung erfolgt nach aktuellen Standards.
3. Das Zertifikat des Backends wird bei Verbindungsaufbau verifiziert und es wird nur Zertifikaten vertraut, welche durch eine gültige Zertifizierungsstelle signiert wurden.
4. Die Anwendung wird mit Zertifikat des Backends ausgeliefert und akzeptiert ausschließlich sichere Verbindungen bei Verwendung des voreingestellten Zertifikates (*Certificate Pinning*) .
5. Die Anwendung nutzt für kritische Operationen und Transaktionen keinen einzelnen unsicheren Kommunikationskanal (z.B. nur SMS) zur Durchführung.
6. Die Anwendung kann ausschließlich mit vordefinierter Liste an Backends kommunizieren.

Plattforminteraktion

1. Die Anwendung fordert nur ein Minimum an Berechtigungen an.
2. Nutzereingaben und Eingaben von externen Quellen / Schnittstellen werden validiert.
3. Die Anwendung nutzt ausschließlich sichere Implementierungen für die Interprozesskommunikation mit anderen Anwendungen.
4. Die Anwendung exportiert sensitive Daten über URL Schemata nur bei ausreichender Sicherung.
5. Javascript ist bei Verwendung von WebViews ohne triftigen Grund deaktiviert.
6. WebViews werden so konfiguriert, dass keine kritischen Handler (z.B. tel://, file://) aufgerufen werden dürfen.
7. Die Anwendung nutzt eine sichere Serialisierung.
8. Die Anwendung erkennt die Ausführung auf einem Gerät mit Root-Berechtigung.
9. Die Anwendung erfordert ein aktuelles Betriebssystem.

Anwendungsgenerierung

1. Die Anwendung ist vorbereitet und signiert mit einem gültigen Zertifikat.
2. Die Anwendung wurde im *„release mode“* kompiliert.
3. Die Anwendung wurde obfuskiert.

Codequalität

1. Debugging Code wurde entfernt.
2. Die Anwendung protokolliert Nachrichten ohne Debugginformationen.
3. Die Anwendung behandelt alle möglichen Exceptions.
4. Der Quelltext enthält keine sensitiven Daten.
5. Die Fehlerbehandlung in Sicherheitsmechanismen verweigert den Zugriff standardmäßig.
6. Es soll sichergestellt werden, dass der Speicher bei nicht verwaltetem Code sicher verwaltet wird.
7. Freie Sicherheitsfeatures sind aktiviert.

iOS

1. ASLR[1] ist aktiviert.
2. Schutz vor Buffer Overflow mittels Stack Smashing wird verwendet.

Android

1. *Activities* sollen sicher implementiert werden.
2. Unsichere Parameter im Manifest dürfen nicht konfiguriert werden.
3. Eine Nutzung des externen Speichers darf nur mit zusätzlichen Sicherheitsmechanismen verwendet werden.

5.4.2 Durchführung

Zur Durchführung eines Penetrationstest wurden für jede konkrete Anforderung aus Kapitel 5.4.1 eine oder mehrere Aufgaben definiert, die eine Untersuchung für die Plattformen Android und iOS ermöglichen. Die Aufgaben wurden um Vorbedingungen erweitert, welche besondere Voraussetzungen für die Analyse anzeigen:

- Mobiles Endgerät mit privilegierten Rechten (Root / Jailbreak)
- Anwendung ohne Jailbreak-Erkennung / SafetyNET-Implementierung

[1] Address Space Layout Randomisation

- Anwendung ohne Zertifikatspinning

Für jede dieser Aufgaben wurde in der Wissensdatenbank von *OptimaBIT* ein Artikel verfasst, welcher den Angriffsvektor bei Nichterfüllung einer konkreten Anforderung skizziert, welche Tools für die Untersuchung verwendet werden können und wie diese Tools konfiguriert werden müssen.[2]

Nachfolgend wird beispielhaft ein Artikel aus der Wissensdatenbank vorgestellt:

Anforderung:

Sensible Daten werden auf dem Gerät sicher gespeichert.

Hintergrundinformationen:

Die Speicherung von Daten auf einem Smartphone kann aus Entwicklersicht durch eine Vielzahl an Möglichkeiten realisiert werden. Der von Apple in den Entwicklerguides empfohlene Weg zur Speicherung von sensiblen Daten, ist die Verwendung der Keychain Schnittstelle. Diese ermöglicht eine sichere und verschlüsselte Speicher von Informationen im hardwaregetrennten Bereich (Secure Enclave) des Gerätes.

Testdurchführung:

Um die von einer Anwendung in die Keychain gespeicherten Einträge auslesen zu können, bietet sich die Nutzung des Tools *Objection (siehe Kapitel 5.5)* an. Mittels Objection ist es möglich Befehle von einem Computer im Kontext der Anwendung auszuführen. Um diese Interaktion zu ermöglichen, ergänzt Objection die Binärdatei einer Anwendung um eine Bibliothek, welche durch Öffnen eines Sockets Befehle von einem Computer empfangen und ausführen kann. Dies hat den Vorteil, dass zur Analyse einer Anwendung auf einem iOS-Gerät, kein Jailbreak erforderlich ist.

Die Erweiterung und Installation einer manipulierten Binärdatei wird mit den folgenden Befehlen durchgeführt:

```
// 1. Ermitteln der Identity des Developer Zeritifikats
security find-identity -p codesigning -v

// 2. Patchen der Binaerdatei mittels Objection
objection patchipa --source my-app.ipa --codesign-signature
    0C2E8200Dxxxx

// 3. Entpacken der Binaerdatei
unzip <my-injected-app>.ipa

// 4. Installation der Binary und Starten der Anwendung im
    Debug Modus
```

[2] Insgesamt 93 verfasst Artikel (Stand: 30.10.2017.

```
ios-deploy --bundle --source Payload/my-injected-app.app -d

// 5. Aufruf einer interaktiven Shell zwischen Computer und
   iOS Geraet
objection explore

// 6. Dumpen der Anwendungsbezogenen Keychaineintraege
ios keychain dump --json
```

Listing 5.1: Erklärung zum Dumpen der Keychain

Eine Anwendung die sensible Daten verarbeitet, muss nach dem Dumpen der Keychain die entsprechenden Daten ausgeben. Sollte im Dump der Keychain die gesuchten Informationen nicht enthalten sein, muss geprüft werden, wo und wie die Anwendung stattdessen diese Daten speichert.

5.4.3 Bewertung

Die Bewertung gefundener Schwachstellen unterliegt einer internen *OPTIMA-Bewertungsmatrix*. Alternativ ist eine Bewertung nach dem *Common Vulnerability Scoring System (CVSS)* möglich. Eine Einführung in diese Bewertungsmatrix wurde bereits im Grundlagenteil vorgenommen.

5.5 Toolchain

Dieser Abschnitt liefert einen Überblick aktuell nutzbarer Open Source Lösungen für das Testen von mobilen Anwendungen:

Apktool

Dieses Tool[3] liefert eine Unterstützung für das Reverse Engineering von Android Anwendungen.

BadIntent

Das Tool *BadIntent*[4] bietet im Zusammenspiel mit einem HTTP(S)-Proxy (z.B. Burp) die Möglichkeit, unter Android Interprozesskommunikation durch einen HTTP-Request anzustoßen / zu automatisieren.

Drozer

Drozer[5] ist ein Android Testing Framework, welches die Suche nach Schwachstellen automatisiert. Es besteht aus einem Set von Modulen, welche verschiedene Angriffsszenarien durchführen können.

JD-GUI

JD-GUI[6] ist ein Java Decompiler und kann zur Dekompilation von Android Anwendungen verwendet werden.

[3] https://github.com/iBotPeaches/Apktool
[4] https://github.com/mateuszk87/BadIntent
[5] https://github.com/mwrlabs/drozer
[6] https://github.com/java-decompiler/jd-gui

Mobile Security Framework (MobSF)

Das Framework *MobSF*[7] bietet für Android und iOS Anwendungen statische und dynamische Analysen der Binärdateien an.

Clutch

Clutch[8] ermöglicht auf einem gejailbreakten Gerät das Entschlüsseln der Binärdatei von installierten Anwendungen. iOS verschlüsselt Teile der Binärdatei einer Anwendung bei Veröffentlichung über den AppStore, wodurch einige Untersuchungen (Kapitel 5.4.1) nicht durchgeführt werden können. Bei einer erfolgreichen Entschlüsselung der Binärdatei via *Clutch* können diese Untersuchungen getestet werden.

Dumpdecrypted

Dumpdecrypted[9] bietet wie *Clutch* die Möglichkeit, verschlüsselte Binärdateien zu entschlüsseln.

IPAInstaller

Der *IPAInstaller*[10] ermöglicht die Installation von iOS-Anwendungen auf ein iOS-Gerät über die Kommandozeile und umgeht damit den aufwendigeren Weg über die Software iTunes.

Keychain-dumper

Der *Keychain-dumper*[11] ermöglicht den Export der Keychain von einem gejailbreakten iOS-Gerät in eine JSON-Datei.

Burp Suite Mobile Assistent

Die über Burp auf gejailbreakten iOS-Geräten verfügbare Anwendung[12] ermöglicht das Deaktivieren von Zertifikatspinning ausgewählter Anwendung.

Objection

Objection[13] ermöglicht auf nicht gejailbreakten Geräten, das Auslesen der Keychaineinträge einer Anwendung, die Simulation eines Jailbreaks und das Durchsuchen des Home-Verzeichnisses einer Anwendung, durch Code-Injection.

iphone-dataprotection

Iphone-dataprotection[14] ist ein forensisches Tool, welches zur Entschlüsselung verschlüsselter iOS-Backups verwendet werden kann. Nachdem in verschlüsselten iOS-Backups die komplette Keychain eines Gerätes gespeichert wird, kann diese über das Tool ausgelesen werden.

[7] https://github.com/MobSF/Mobile-Security-Framework-MobSF.git
[8] https://github.com/KJCracks/Clutch
[9] https://github.com/stefanesser/dumpdecrypted
[10] http://hshersy.com/?l=ipainstaller.html
[11] https://github.com/ptoomey3/Keychain-Dumper
[12] https://support.portswigger.net/customer/en/portal/articles/2798916-installing-burp-suite-mobile-assistant
[13] https://github.com/sensepost/objection
[14] https://github.com/dinosec/iphone-dataprotection

libplist

Libplist[15] konvertiert iOS-plist Dateien in eine XML-Datei und ist damit für Menschen lesbar.

needle

Needle[16] ist das iOS-Äquivalenz zum Android-Framework Drozer. Es besteht aus einem Set von Modulen, welche verschiedene Angriffsszenarien durchführen können.

Frida

Frida[17] ist eine Anwendung, welche Codeinjection von Anwendungen unter anderem für Android und iOS ermöglicht. Das Tool enthält einen Javascript-Parser, welcher in Javascript geschriebene Snippets (z.B. Hooken der Jailbreak-Erkennung) in die Assemblersprache der jeweiligen Plattform übersetzt.

[15] https://github.com/libimobiledevice/libplist
[16] https://github.com/mwrlabs/needle
[17] https://www.frida.re

6. Zusammenfassung und Ausblick

Die Entwicklung eines Sicherheitsmodells zur Bestimmung des (aktuellen) Sicherheitslevels eines Smartphones ermöglicht erstmals die Bewertung der Gesamtsicherheit eines Smartphones. Zusätzlich wurde damit auch das Problem gelöst, dass klassische Penetrationstests von mobilen Anwendungen nur die Schwachstellen einer Anwendung identifizieren und damit keine Aussage über die Sicherheit der Daten auf einem Smartphone treffen können. Das Modell selbst trennt dabei die Bewertung des Gerätes in einen generischen und einen individuellen Teil. Der generische Teil besteht dabei aus der Hardware, der aktuellen Betriebssystemversion, des Managements sowie den vorinstallierten Standardanwendungen und kann einer allgemein Evaluation unterzogen werden. Der individuelle Teil umfasst einen Penetrationstest derjenigen Anwendungen, welche durch die Nutzergruppe verwendet werden.
Die Nachhaltigkeit des Sicherheitsmodells liegt darin, dass alle Anforderungen auf konzeptueller Ebene definiert wurden. Zusätzlich ist die Verwendbarkeit des Modells durch die individuelle Betrachtung der technischen Implementierungen jedes Gerätes in der Umsetzungsanalyse auch in den nächsten Jahren sichergestellt. Das Erscheinen eines neuen Akteurs bei Hardwareherstellern oder bei mobilen Betriebssystemen stellt für das Modell ebenfalls kein Problem dar, da die zugrundeliegende Architektur erst in der Umsetzungsanalyse betrachtet wird.
Für die Risikobewertung der von einer Schwachstelle ausgehenden Bedrohung wurde das *OWASP Risk Rating* Modell ausgewählt, da es eine Bewertung der Ausnutzbarkeit einer Schwachstelle vornimmt und ebenso eine Klassifizierung der technischen und finanziellen Auswirkungen einer bestimmten Anwendergruppe berücksichtigt. Zusätzlich erlaubt das Modell eine Erweiterung der Risikobewertung um weitere Einflussfaktoren. Für die vorliegende Masterarbeit wurde diese Erweiterungsmöglichkeit genutzt, um bei der Einstufung des Schweregrades einer Schwachstelle individuelle Einflussfaktoren von Smartphones zu berücksichtigen.
Um dem hohen Sicherheitsbedarf von beispielsweise Behörden und Organisationen mit Sicherheitsaufgaben (BOS) nachzukommen, wurden im Managementteil des Modells Anforderungen zur Transparenz der Hersteller definiert.

Nach der Evaluation des Modells durch drei unterschiedliche Nutzergruppen mit jeweils zwei verschiedenen Smartphones zeigte sich, dass aktuell kein Smartphone existiert,

welches über keine Schwachstellen verfügt. Hinsichtlich des Schweregrads der offenen Schwachstellen konnten in der aktuellen Auswertung bei Smartphones mit dem Betriebssystem iOS weniger hohe Gefährdungen im Vergleich zu Smartphones mit einer Androidumgebung ermittelt werden.
Eine mögliche Ursache dieser Ergebnisse könnte in der restriktiven Politik von Apple im Umgang mit Schwachstellen und bei der Veröffentlichung von Informationen über Art und Umfang der Schwachstelle liegen. Im Dokument zu den Sicherheitsupdates von Apple wird der Leser darauf hingewiesen, dass für Apple der Schutz der Kunden eine höhere Priorität genießt und Sicherheitsprobleme erst nach einer gründlichen Untersuchung und der Bereitstellung von Patches bekannt gegeben werden. Dies erschwert die Suche nach aktuellen Schwachstellen, da eine intensive Recherche von Drittquellen erforderlich war. Für die Analyse der Umsetzung von Anforderungen aus dem Sicherheitsmodell konnte nur auf den von Apple veröffentlichten Sicherheitsguide für iOS zurückgegriffen werden. Deswegen ist für die Bewertung des Sicherheitslevels eines iOS-Gerätes das Vertrauen in das Unternehmen Apple für die Entwicklung sicherer Geräte von hoher Bedeutung. Für das Vertrauen in das Unternehmen sprechen Entscheidungen in der Vergangenheit, so zum Beispiel die Weigerung einer Anordnung des FBI Folge zu leisten und eine Schwachstelle in das Betriebssystem einzubauen, um die Anzahl an Fehlversuchen bei der Eingabe des Geräte-PINs zu erhöhen[Wil]. Zusätzlich ist das Unternehmen nach ISO 27001 zertifiziert, jede iOS Version wird vor Auslieferung von der Common Criteria untersucht und diverse Krypto-Module sind auf die Einhaltung bestimmter Standards (FIPS 140-2) validiert.

Samsung als Hersteller des zweiten evaluierten Smartphones, nutzt das freie Betriebssystem Android. Dies birgt den Nachteil, dass die Veröffentlichung einer Android-Aktualisierung vor dem Ausrollen auf Samsung-Geräte eine Anpassung an die konkrete Hardware benötigt, was in jedem Fall zu einem zeitlichen Nachteil gegenüber Apple's Geräten führt. Hinsichtlich der Transparenz im Umgang mit Sicherheitslücken stellt das Betriebssystem Android im Vergleich zu Apple's iOS der Allgemeinheit wesentlich mehr Informationen zur Verfügung. Mit Einführung der monatlichen Sicherheitsupdates als Reaktion auf die *Stagefright* Schwachstelle, kann auf jedem als Betriebssystem Android nutzenden Smartphone eine Überprüfung des aktuellen Stands der Sicherheitsupdates, sowie eine Identifikation der aktuellen Schwachstellen des Geräts vorgenommen werden. Im Gegensatz dazu, wird unter iOS bei Veröffentlichung eines neuen Patches nur eine Auflistung von Informationen über die behobenen Schwachstellen und eine rudimentäre Beschreibung der Auswirkungen auf das Gerät publiziert.

Beim Vergleich der gefundenen und behobenen Schwachstellen unter Android und iOS lässt sich konstatieren, dass für das Betriebssystem iOS im Bearbeitungszeitraum der Masterarbeit weniger Schwachstellen veröffentlicht wurden. Durch die vergleichsweise geringe Vielfalt von iOS-Geräten gegenüber Android-Geräten ist der Aufwand für die Behebung gering und das Zeitfenster vom Bekanntwerden einer Schwachstelle bis zur Bereitstellung einer Sicherheitsaktualisierung kürzer.

Das Modell bietet zahlreiche Einsatzmöglichkeiten. Unternehmen oder Behörden, welche vor der Entscheidung stehen, Smartphones für eine Vielzahl an Nutzern anzuschaffen, können dieses Modell nutzen, um vor einer Kaufentscheidung das derzeitige Sicherheitslevel potentieller Geräte zu bestimmen.

Neben einer Massenbeurteilung von Geräten bietet das Modell auf die Möglichkeit einzelne Geräte während des Lebenszyklus mehrfach einer Bewertung zu unterziehen, um den Verlauf des Sicherheitslevels zeitlich darzustellen. Damit können Rückschlüsse gezogen werden, wie schnell der betroffene Hersteller grundsätzlich auf Sicherheitsvorfälle regiert, ob sich eine zeitliche Verlängerung der Updatezyklen mit zunehmender Alterung des Gerätes feststellen lässt und wie sich das Sicherheitslevel mit zunehmender Zeit entwickelt. Zukünftige Untersuchungsansätze können auf der Basis des in dieser Arbeit geschaffenen Sicherheitsmodells einen Vergleich zwischen der Anzahl geschlossener Schwachstellen und der Zeitspanne des Bekanntwerdens einer Schwachstelle bis zur Behebung ziehen. Bei Anwendung dieses Modells auf eine Vielzahl an Geräten kann anschließend eine Auswertung nach Hersteller vorgenommen werden, mit dem Zielt der Identifikation von Anbietern, welche der Sicherheit ihrer Geräte eine höhere Bedeutung beimessen. Durch weitergehende Analysen kann zusätzlich der Frage nachgegangen werden, ob ein Zusammenhang zwischen dem Sicherheitslevel eines Smartphones und dem Kaufpreis besteht.

Nicht zuletzt durch die immense Zunahme von Malware, die immer öfter nach Infektion eine Rechteausweitung hin zu Rootrechten anstrebt[McA17; Kas16], richtet sich der Appell dieser Arbeit an die Hersteller von mobilen Endgeräten, der Sicherheit von Software einen höheren Stellenwert einzuräumen. Hersteller müssen sich den zunehmenden Herausforderungen mobiler Malware durch eine zeitnahe Behebung der Schwachstellen annehmen, um den Angreifern, die in der fehlenden oder verzögerten Behebung kritischer Schwachstellen immer mehr eine gute Angriffsfläche für ihre Anwendungen sehen, den Nährboden zu entziehen.

Ebenso sollte die Politik mehr in die Verantwortung gezogen werden. Ein sinnvoller Schutz vor verspäteten oder ausbleibenden Sicherheitsupdates, sowie ein wichtiger Fortschritt zur Erhöhung der Sicherheit mobiler Endgeräte, wäre eine gesetzliche Anpassung und Erweiterung der Produkthaftung für Software. Dies würde die Hersteller gesetzlich zum zeitnahen Handeln verpflichten.

A. Kategorien der Risikobewertung

Dieser Anhang zählt alle vorhandenen Bewertungen der Kategorien der Risikobewertung auf. Eine ausführliche Erklärung fand bereits im Kapitel 4.2.7 statt.

A.1 Technische Auswirkungen

Vertraulichkeit

- 0 - Keine Daten
- 2 - Kaum sensitive Daten
- 5 - Kaum kritische Daten, aber umfassende sensitive Daten
- 6 - Wenige kritische Daten
- 7 - Umfangreiche kritische Daten
- 9 - Alle Daten

Verfügbarkeit

- 1 - Kleine untergeordnete Dienste
- 5 - Kleinere Hauptdienste unterbrochen
- 7 - Zahlreiche Hauptdienste unterbrochen
- 9 - Alle Dienste unterbrochen

Integrität

- 1 - Geringfügige Manipulationen
- 3 - Kleine schwerwiegende Manipulationen
- 5 - Umfangreiche kleinere Manipulationen
- 7 - Umfangreiche schwerwiegende Manipulationen
- 9 - Komplette (Daten-) Manipulation

Verantwortlichkeit

- 1 - Komplett identifizierbar
- 7 - Möglicherweise identifizierbar
- 9 - Anonym / Nicht identifizierbar

Intervenierbarkeit

- 0 - Vollständige Datenhoheit
- 3 - Teilweise Datenhoheit
- 7 - Keine Datenhoheit

A.2 Finanzielle Auswirkungen

Materieller Schaden

- 1 - Geringer als Kosten zur Behebung der Schwachstelle
- 3 - Kleine Auswirkungen auf jährlichen Umsatz
- 7 - Signifikante Auswirkungen auf jährlichen Umsatz
- 9 - Insolvenz

Ansehen

- 1 - Niedriger Ansichtsverlust
- 4 - Verlust betroffener Kunden
- 6 - Verlust von Großkunden
- 7 - Verlust des (Firmen-)Rufs
- 9 - Markenverlust

Gesetzesverstoß

- 0 - Keine Verstöße
- 2 - Kleine Verstöße
- 5 - Einzelne Verstöße
- 7 - Öffentlichkeitswirksame Verstöße

Privatsphäre

- 0 - Kein Betroffener
- 2 - Einzelner
- 3 - Mehrere Betroffene
- 5 - Hunderte Betroffene
- 7 - Tausende Betroffene
- 9 - Millionen Betroffene

A.3 Angreifer

Fähigkeiten

- 1 - Keine Technikkenntnisse
- 3 - Erweiterte Technikkenntnisse
- 5 - Fortgeschrittene Fähigkeiten
- 6 - Netzwerk- und Programmierkenntnisse
- 9 - Penetrationtester

Motivation

- 1 - Niedrige Motivation
- 4 - Mittelmäßige Motivation
- 9 - Hohe Motivation

Aufwand

- 0 - Vollzugriff erforderlich
- 4 - Spezialrechte erforderlich
- 7 - Einmaliger Zugriff erforderlich
- 9 - Kein Zugriff erforderlich

Anzahl Angreifer (Gruppe)

- 2 - Entwickler
- 4 - Intranetnutzer
- 5 - Partner
- 6 - Authentifizierte Nutzer
- 9 - Anonyme (Internet-) Nutzer

A.4 Schwachstelle

Entdeckung

- 1 - Praktisch unmöglich
- 3 - Kompliziert
- 7 - Leicht
- 9 - Automatisierte Tools verfügbar

Ausnutzung

- 1 - Theoretisch
- 3 - Kompliziert
- 5 - Leicht
- 9 - Automatisierte Tools verfügbar

Bekanntheit

- 1 - Unbekannt
- 4 - Verborgen
- 6 - Offensichtlich
- 9 - Öffentlich

Transparenz

- 1 - Sourcecode einsehbar und Dokumentation
- 3 - Dokumentation mit Hintergrundinformationen
- 4 - Dokumentation ohne Hintergrundinformationen
- 6 - keine Information

Erkennung

- 1 - Aktive Erkennung
- 3 - Logging mit Mitteilung
- 7 - Logging ohne Mitteilung
- 9 - Unprotokolliert

Angriffsvektor

- 1 - Physikalisch
- 4 - Lokal
- 6 - Angrenzend
- 9 - Netzwerk

Nutzerinteraktion

- 3 - Erforderlich
- 7 - Nicht erforderlich

Umfang

- 3 - Lokal
- 7 - Übergreifend

B. Sicherheitsanforderungen und deren Umsetzung in iOS und Android

Dieser Anhang beschreibt die Anforderungen aus dem Sicherheitsmodell detaillierter und liefert Hintergrundinformationen, welche Gefahren bei einer fehlenden Umsetzung möglich sind. Anschließend werden die konkreten technischen Implementierungen auf den Smartphones *Samsung Galaxy S7 (Android 7.0 - Sicherheitsupdates 01.08.2017, SM-G930F)* sowie *iPhone 7 (iOS 10.3.3)* vorgestellt, evaluiert und einer Bewertung unterzogen.

Anmerkung des Autors: Aus Darstellungsgründen wird bei jeder Implementierung, welche über Schwachstellen verfügt, nur die jeweils schwerwiegendste Schwachstelle bewertet.

B.1 Hardware

Dieser Abschnitt behandelt alle hardwarerelevanten Anforderungen und ist unterteilt in die Kategorien *Schnittstellen* und *Trusted Execution Zone (Hardwaregetrennter Bereich)*.

Schnittstellen

R-HW.01

Anforderung: Physikalische Schnittstellen müssen sicher vor unautorisierter Nutzung sein.

Beschreibung: Die physikalischen Schnittstellen (USB / Lightning, JTAG) sollen vor einer unautorisierten Nutzung geschützt werden, um zu verhindern, dass Unbefugte Daten extrahieren oder Schadsoftware installieren können.

Allgemeine Hintergrundinformationen: Physikalische Schnittstellen stellen für Angreifer, welche im Besitz des Gerätes sind, eine Möglichkeit dar, Zugriff auf das Dateisystem, oder den Arbeitsspeicher zu erlangen.

Umsetzung iPhone 7: Das iOS-Betriebssystem überprüft vor der Kommunikation über ein Lightning-Kabel, ob es von Apple signiert wurde, andernfalls ist nur ein Laden des Gerätes oder die Nutzung des analogen Audiosignals möglich[Incm, S. 24]. Die Kommunikation zwischen iTunes und einem iOS Gerät erfolgt durch ein proprietäres Protokoll, zu dessen Funktionsweise keine Dokumentation gefunden werden konnte.

Schwachstelle: Eine Möglichkeit des Zugriffs auf die zur Verschlüsselung der Benutzerdaten verwendeten hardwareseitigen Schlüssel über Debugging-Schnittstellen konnte nicht eruiert werden. Ebenfalls ist kein Vorgehen bekannt, um über eine Debugging-Schnittstelle Zugang zum Flash-Speicher zu erhalten.

Bewertung:

HW.01	Physikalische Schnittstellen müssen sicher vor unautorisierter Nutzung sein.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Mittels der USB Schnittstelle können folgende Dienste genutzt werden:

- Dateisystemzugriff auf Partition *sdcard*
- Android Debug Bridge (adb) nach vorheriger Aktivierung
- adb sideload zur Installation von signierten (System-)updates

Schwachstelle: Für das vorliegende Gerät konnte während der Bearbeitungszeit dieser Arbeit keine Schwachstelle ermittelt werden, welche einen unautorisierten Zugriff oder Manipulation von Benutzerdateien durch physikalische Schnittstellen ermöglicht.

Bewertung:

HW.01	**Physikalische Schnittstellen müssen sicher vor unautorisierter Nutzung sein.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Beschreibung: Drahtlose Schnittstellen (Bluetooth / WLAN) sollen aktuelle Standards zur sicheren Übertragung von Daten verwenden.

Allgemeine Hintergrundinformationen: Der Bedarf an WLAN Hotspots um das Smartphone mit dem Internet zu verbinden, kann allein durch die riesige Anzahl an angebotenen Hotspots an öffentlichen Plätzen begründet werden. Durch die Nutzung eines öffentlichen Hotspots mit dem Smartphone ergeben sich multiple Bedrohungen für die Sicherheit des Gerätes. Obwohl die Kommunikation zwischen Smartphone und Access Point auf Ebene 1 (Bitübertragungsschicht) verschlüsselt abläuft, können auf Ebene 3 (Netzwerkschicht) unverschlüsselte IP-Pakete abgehört werden. Hierzu sind weitere Sicherungsmaßnahmen erforderlich, beispielsweise die Etablierung einer VPN-Verbindung.

Umsetzung iPhone 7: Der jüngste WLAN Standard 802.11ah (siehe Kapitel 2.3) wird nicht unterstützt. Das iPhone 7 unterstützt den Standard 802.11ac[Incn]. Zur Verschlüsselung auf Ebene 1 werden alle aktuellen Standards (2017) unterstützt.

Schwachstelle: Die als *Broadpwn* vorgestellte Schwachstelle im Treiber des Chipherstellers Broadcom, ermöglicht einem Angreifer, durch ein modifiziertes WLAN Paket, Schadcode mit Systemrechten auszuführen. Diese Lücke wurde in Version 10.3.3 behoben[1].
Am 27.09.2017 wurde von einem Forscher des Project Zero (Google) ein Exploit veröffentlicht, welcher abermals eine Schwachstelle in der Broadcom-Firmware ausnutzt. Damit ist die volle Kontrolle des WLAN-Moduls möglich[2][Zer].

Angriffsszenario: Durch die Konfiguration des Exploits mit den erforderlichen Parametern, ist ein Angreifer in der Lage, Schadcode (Payload) mit Systemrechten auszuführen. Damit ist eine Kompromittierung aller auf dem Gerät gespeicherten Daten möglich. Inwiefern ein Angreifer Informationen aus der Keychain des Gerätes extrahieren kann, konnte nicht recherchiert werden. Im AppStore Cydia[3] existieren Tools, welche auf Geräten mit Root-Rechten die komplette Keychain auslesen können. Deswegen kann angenommen werden, dass durch Nutzung dieses Exploits sensible Daten aus der Keychain entschlüsselt werden können. Abhängig von der gewählten Datensicherheitsklasse (siehe Kapitel 2.6.4) können Daten der Klasse C und D extrahiert werden.

[1] https://www.heise.de/mac-and-i/meldung/Broadpwn-Kritische-iPhone-WLAN-Schwachstelle-gestopft-3779461.html

[2] https://www.heise.de/mac-and-i/meldung/iPhone-7-Exploit-fuer-kritische-WLAN-Luecke-veroeffentlicht-3845431.html

[3] Cydia ist ein inoffizieller App Store, welcher auf gejailbreakten Gerät installiert werden kann und diverse Tools zur Umgehung von Sicherheitsmechanismen des Betriebssystems enthält.

Bewertung:

HW.02	Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	6 - Angrenzend
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	6.25 - Hoch
Gesamt Firma 1:	6.00 - Hoch
Gesamt Firma 2:	7.00 - Hoch

Umsetzung Samsung Galaxy S7: Die populäre Schwachstelle *Broadpwn (CVE-2017-9417)* betraf ebenfalls diverse Android-Geräte (u.a. Samsung Galaxy S7). Mit dem Sicherheitsupdate vom 01.08.2017 wurde diese Lücke von Samsung durch ein Update am 30.08.2017 für dieses Gerät geschlossen.
Im September 2017 tauchten weitere kritische Schwachstellen in der Firmware von Broadcom für WLAN Module auf, welche das Einschleusen und Ausführen von Schadcode ermöglichen (CVE-2017-11120, CVE-2017-11121, CVE-2017-7065)[Prog].
Am 13. September veröffentlichten Sicherheitsforscher der Firma Armis unter dem Titel *BlueBorne* eine Reihe an Schwachstellen in der Implementierung von Bluetooth-Schnittstellen, welche unter Android eine Ausführung von Schadcode mit Systemrechten ermöglicht.[SV17]

Angriffsszenario: Für die Schwachstelle *BlueBorne* existiert ein öffentlich verfügbarer Exploit der die Ausführung von Schadcode mit privilegierten Rechten ermöglicht. Damit ist ein Angreifer in der Lage alle gespeicherten Daten auf dem Smartphone zu korrumpieren. Die Entschlüsselung von Daten, welche mittels des Keystores verschlüsselt sind, ist nach jetzigem Stand nicht möglich.

Bewertung:

HW.02	Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065 (Broadcam), CVE-2017-0781 (BlueBorne)
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	6 - Angrenzend
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	6.25 - Hoch
Gesamt Firma 1:	6.00 - Hoch
Gesamt Firma 2:	7.00 - Hoch

Trusted Exection Zone (Hardwaregetrennter Bereich)

R-HW.03

Anforderung: Das Gerät muss über hardwaregetrennte Module verfügen, welche sicherheitsrelevante Aufgaben übernehmen.

Beschreibung: Das Erstellen eines für die Haupt-CPU nicht direkt erreichbaren und getrennten Bereichs ermöglicht die Auslagerung von sicherheitsrelevanten Aufgaben. Die Kommunikation zwischen Haupt-CPU und dem hardwaregetrennten Bereich wird über eine abgesicherte Schnittstelle ermöglicht.

Allgemeine Hintergrundinformationen: Durch die Auslagerung können Sicherheitsaufgaben des Betriebssystems (z.B. Schlüsselaufbewahrung, Integritätsüberprüfungen) in einen hardwaregetrennten Bereich, erhöht sich der Aufwand eines Angreifers enorm. Neben einer Sicherheitslücke im Betriebssystem muss zusätzlich eine Schwachstelle in der Schnittstelle zum hardwaregetrennten Bereich eruiert werden, welche die Ausführung von Schadcode im hardwaregetrennten Bereich ermöglicht.
Durch die Zunahme des Bezahlens von Einkäufen mit einem Smartphone, bieten die Betriebssysteme Android und iOS mittlerweile diverse Lösungen an, welche ein Bezahlen softwareseitig oder hardwareseitig (NFC) ermöglichen. Zur Absicherung dieser Verfahren, ist die Nutzung hardwaregetrennter Bereiche von Vorteil.

Anmerkung des Autors: Eine konkrete Analyse der Umsetzung dieser Anforderung findet nicht statt. Diese Anforderung wird lediglich unterteilt in die Existenz bzw. Nicht-Existenz eines hardwaregetrennten Bereichs. Sollte ein hardwaregetrennter Bereich für ein ausgewähltes Smartphone existieren, ergeben sich daraus weitere Anforderungen. Sollte kein derartiger Bereich vorhanden sein, so kann dies in Teilen auf Softwareebene abgedeckt werden, welche im Abschnitt Betriebssystem thematisiert wird.

Umsetzung iPhone 7: iOS realisiert mit der *Secure Enclave* einen hardwaregetrennten Bereich. Zusätzlich existiert ein dediziertes Krypto-Modul, welches die Verschlüsselung von Informationen durchführt.
[Incm]

Schwachstelle: Eine Recherche öffentlicher Schwachstellen wird in dieser Anforderung nicht vorgenommen, da ausschließlich eine Bewertung in Umsetzung oder fehlende Umsetzung vorgenommen wird. Die von dieser Anforderung abgeleiteten Anforderungen werden einer Schwachstellenrecherche unterzogen.

Bewertung:

HW.03	Das Gerät muss über hardwaregetrennte Module verfügen, welche sicherheitsrelevante Aufgaben übernehmen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Diese Anforderung wird umgesetzt.

Umsetzung Samsung Galaxy S7: Die für dieses Gerät verwendete Architektur des *System on a Chip* stammt von ARM. Dieses sieht in der verwendeten Version eine Unterstützung der Trustzone vor. Die konkrete Hardwareimplementierung von Samsung trägt den Namen *Exynos 8890* und besitzt einen hardwaregetrennten Bereich.
[Eur]

Schwachstelle: Eine Recherche öffentlicher Schwachstellen wird in dieser Anforderung nicht vorgenommen, da ausschließlich eine Bewertung in Umsetzung oder fehlende Umsetzung vorgenommen wird. Die von dieser Anforderung abgeleiteten Anforderungen werden einer Schwachstellenrecherche unterzogen.

Bewertung:

HW.03	**Das Gerät muss über hardwaregetrennte Module verfügen, welche sicherheitsrelevante Aufgaben übernehmen.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung wird umgesetzt.

R-HW.03.1

Anforderung: Das Gerät muss Hardwareunterstützung zur Verschlüsselung von Daten ermöglichen.

Beschreibung: Die Verschlüsselung von Daten und die Verwaltung der Hardwareschlüssel soll in einer eigenen hardwaregetrennten Umgebung durchgeführt werden. Die Übertragung der zu verschlüsselnden Daten von der Hauptplatine auf die ausgelagerte Verschlüsselungsengine wird durch eine geschützte Schnittstelle realisiert.

Allgemeine Hintergrundinformationen: Die Separation der Verschlüsselung von Daten in einen hardwaregetrennten Bereich hat neben einem Performancegewinn den Vorteil, dass der Aufwand zur Extraktion der Hardwareschlüssel deutlich erhöht wird, da die Verschlüsselung von Informationen in einem separaten, für das Betriebssystem nicht erreichbaren, Prozesskontext ausgeführt wird und zum Beispiel Pufferüberläufe nicht ausgenutzt werden können, um die Verschlüsselung zu korrumpieren.

Umsetzung iPhone 7: Dieses Smartphone verfügt über eine dedizierte AES 256 Crypto-Engine, welche die Verschlüsselung von Benutzerdaten übernimmt. Die Hardwareschlüssel werden bei der Gerätefertigung in Silizium eingebrannt und sind somit unveränderlich. Die Verwaltung der Schlüssel findet in der *Secure Enclave* statt, welche ebenfalls über eine dedizierte CPU verfügt und von der Hauptplatine abgekoppelt interagiert.

Schwachstelle: Stand 2017 gilt AES 256 als sicheres Verfahren zur Verschlüsselung von Informationen[4][ST]. Im August 2017 wurde der Firmwarekey der Secure Enclave für das iPhone 5s publik. Damit ist es möglich die Firmware der Secure Enclave für dieses Modell zu entschlüsseln. Ein Zugriff auf die in der Secure Enclave gespeicherten Daten ist hierdurch nicht möglich, es besteht jedoch die Möglichkeit, dass durch den entschlüsselten Firmware-Code Schwachstellen in der Implementierung entdeckt werden, welche aktuelle Modelle ebenfalls betreffen könnten[Gie].

Bewertung:

HW.03.1	Das Gerät muss Hardwareunterstützung zur Verschlüsselung von Daten ermöglichen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[4] Der Standard AES 256 ist in den USA für Dokumente der höchsten Geheimhaltungsstufe zugelassen.[ST]

Umsetzung Samsung Galaxy S7: Zur Erhöhung der Sicherheit und Effizienz der Verschlüsselung von Informationen, implementieren manche Hersteller von Android-Geräten im *System on a Chip (SoC)* dedizierte Crypto-Engines (z.B. Snapdragon 821). In diesem Gerät ist ein SoC des Modells Exynos 8890 verbaut, welcher keine dedizierte Verschlüsselung unterstützt.

Bewertung:

HW.03.1	Das Gerät muss Hardwareunterstützung zur Verschlüsselung von Daten ermöglichen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung ist aktuell nicht umgesetzt.

R-HW.03.2

Anforderung: Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.

Beschreibung: Schlüssel, welche zur Wahrung der Vertraulichkeit und Integrität von Informationen verwendet werden, erfordern einen zusätzlichen Schutz vor Missbrauch. Um diese erhöhten Anforderungen umzusetzen, hat sich das Konzept durchgesetzt, sensible Daten in einem für den Hauptprozessor nicht erreichbaren Bereich aufzubewahren und zu verarbeiten.

Allgemeine Hintergrundinformationen: Die Aufbewahrung geheimer Schlüssel (Keys) in hardwaregetrennten Bereichen, erhöht den Aufwand eines Angreifers die Vertraulichkeit dieser Schlüssel zu gefährden. Sollten die Schlüssel auf der Hauptplatine verwaltet werden, besteht die Möglichkeit durch einen Pufferüberlauf oder eine Rechteausweitung die Vertraulichkeit und Integrität dieser Informationen zu verletzen. Bei einem separaten Co-Prozessor ist neben einer Schwachstelle in der Schnittstelle zwischen Haupt- und Co-Prozessor zusätzlich eine Schwachstelle in der Applikation des Co-Prozessors erforderlich.

Umsetzung iPhone 7: Das vorliegende Gerät verwaltet alle Schlüssel in der Keychain, welche in der *Secure Enclave* ausgelagert ist. Die Schlüssel selbst verlassen dabei nie den geschützten Bereich. Zur Nutzung der Keychain müssen alle Daten, welche sicher aufbewahrt werden sollen, in der Keychain gespeichert werden. Bei Erstellung eines Eintrages wird festgelegt, welche Applikationen auf diese Informationen Zugriff erhalten und welche Sicherheitsklasse (siehe Grundlagenteil 2.6) verwendet werden soll.

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle für die Implementierung der Keychain Schnittstelle festgestellt werden.

Bewertung:

HW.03.2	Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet in der *Trustzone* die Nutzung eines Keystore's an. Jeder Keystoreeintrag besteht aus einem öffentlichen und privaten Schlüssel. Der öffentliche Schlüssel wird, wie die verschlüsselte Information, im Applikationsspeicher vorgehalten, während der private Schlüssel die Trustzone nicht verlässt. Um eine verschlüsselte Information zu entschlüsseln, wird diese an den Keystore übergeben, welche die Daten in der Trustzone mittels des privaten Schlüssels entschlüsselt und anschließend der Anwendung zur Verfügung stellt. Als zusätzliches Feature kann festgelegt werden, dass vor Entschlüsselung der Zugriffsschutz des Gerätes (PIN, Passwort, Fingerabdruck) durch den Nutzer eingegeben werden muss.

Schwachstelle: Im September 2017 wurde ein Angriff gegen die ARM-Architektur publik, in der Forscher Schlüssel aus der sicheren Trustzone extrahieren konnten[5][TS17].

Angriffsszenario: Durch ein gezieltes Ausnutzen der Stromsparfunktion der ARM-Architektur durch Übertakten der CPU, besteht die Möglichkeit über die Haupt-CPU Befehle in der Trustzone auszuführen. Damit können potenziell alle in der Trustzone gespeicherten Informationen abgegriffen werden, welche nicht durch den Geräteschutz (PIN, Passwort, Fingerabdruck) gesichert sind.

[5] https://www.heise.de/security/meldung/ClkScrew-Geheime-Schluessel-mit-der-Brechstange-aus-der-ARM-Trustzone-auslesen-3841424.html

Bewertung:

HW.03.2	Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle in ARM Architektur
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	1 - Theoretisch
Bekanntheit:	4 - Verborgen
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	1 - Physikalisch
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.50 - Mittel
Gesamt Firma 1:	5.13 - Mittel
Gesamt Firma 2:	6.13 - Hoch

R-HW.03.3

Anforderung: Kostenintensive Vorgänge oder sensible Daten müssen in hardwaregetrenntem Bereich verarbeitet werden.

Beschreibung: Vorgänge, welche Kosten produzieren können, wie beispielsweise *Mobile Payment* oder Anwendungen, welche besonders kritische Daten (z.B. Gesundheitsdaten) speichern, müssen in einem hardwaregetrennten Bereich verarbeitet werden.

Allgemeine Hintergrundinformationen: Die zunehmende Nutzung von mobilen Bezahldiensten und die Erfassung von Gesundheitsdaten durch externe Geräte oder den Nutzern erfordern eine sichere Speicherung der Daten auf einem Smartphone. Hierzu eignet sich die Verwahrung dieser Daten in einem hardwaregetrennten Bereich. Dies kann durch die bereits existierenden Mechanismen ermöglicht werden oder durch zusätzliche Hardwaremodule.

Umsetzung iPhone 7: iOS bietet nativ durch das Betriebssystem die Möglichkeit mittels *Apple Pay* Zahlungen mobil durchzuführen. Die Abwicklung und Speicherung erfolgt in der *Secure Enclave* und *Secure Element*[6]. Zur Erfassung und Speicherung von Gesundheitsdaten bietet Apple mit der Anwendung *HealthKit* die Möglichkeit selbst Daten (z.B. Vorerkrankungen, Dauermedikation) einzugeben oder durch unterstützte Geräte über eine Bluetooth-Schnittstelle übertragene Gesundheitsdaten (Herzfrequenz, Blutdruck, ...) zu erfassen. Die Daten werden in der Keychain gespeichert und mit der höchsten Datensicherheitsklasse versehen. Damit ist ein Zugriff nur nach vorheriger Eingabe des Geräteschutzes (PIN, Password, Touch ID) möglich.
[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

HW.03.3	Kostenintensive Vorgänge oder sensible Daten müssen in hardwaregetrenntem Bereich verarbeitet werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[6] Secure Element ist ein zertifizierter Hardware-Chip, welcher zur Speicherung von verschlüsselten Geräteaccountnummern verwendet wird. Die Geräteaccountnummer wird verschlüsselt von einer Bank zur Verfügung gestellt.[Incm]

Umsetzung Samsung Galaxy S7: Android bietet mit *Android Pay* eine mobile Zahlungsmöglichkeit an, sollte das Gerät über eine NFC-Schnittstelle verfügen. *Android Pay* wird über eine separate Applikation im PlayStore zur Verfügung gestellt. Es konnte trotz intensiver Recherche keine öffentliche Dokumentation der Sicherheit dieser Anwendung ermittelt werden. Seit Android 6.0 ist eine Unterstützung zur Authentifizierung einer Zahlung mittels der Fingerabdruck API vorhanden. [Sin]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. Es besteht jedoch theoretisch die Möglichkeit durch eine allgemeine Schwachstelle, welche das Ausführen von Schadcode mit privilegierten Rechten ermöglicht, die Vertraulichkeit der Daten der App *Android Pay* unter Umständen zu gefährden.

Bewertung:

HW.03.3	**Das Gerät muss über hardwaregetrennte Module verfügen, welche sicherheitsrelevante Aufgaben übernehmen.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-HW.04

Anforderung: Vor dem Systemstart muss eine hardwareunterstützte Integritätsüberprüfung durchgeführt werden.

Beschreibung: Die Überprüfung der Integrität aller für den Systemstart erforderlichen Komponenten stellt sicher, dass keine Manipulationen am Betriebssystem vorgenommen wurden. Hierzu hat sich das Konzept der *„Chain of Trust“* etabliert, welches beginnend mit dem Bootloader Schritt für Schritt die Integrität der als nächstes geladenen Komponenten überprüft. Sollte eine Signaturüberprüfung fehlschlagen, muss ein Prozess existieren, der den Nutzer darüber informiert oder geeignete Maßnahmen zur Wiederherstellung der Systemintegrität vorsieht.

Allgemeine Hintergrundinformationen: Die Manipulation von Systemkomponenten (Bootloader, Kernel, Betriebssystem, Baseband) mit Schadsoftware ermöglicht einem Angreifer sämtliche Sicherheitsmechanismen des Betriebssystems und darüber liegender Schichten zu deaktivieren und damit auf alle Informationen und Schnittstellen des Gerätes zuzugreifen. Umgangssprachlich werden solche Geräte als „Zombie-Smartphones“ bezeichnet.

Umsetzung iPhone 7: iOS bezeichnet den Prozess der Integritätsüberprüfung aller Systemkomponenten als *sicheren Startvorgang*. Bei der Geräteanfertigung wird in den unveränderlichen Boot-ROM Code das von Apple zur Signierung der Komponenten verwendete Root-Zertifikat festgelegt. Der Hauptprozessor lädt bei Einschalten des Gerätes dieses Zertifikat initial und überprüft damit die Integrität des Bootloaders. Anschließend werden in einer *„Chain of Trust“* alle weiteren hardwarenahen Komponenten (Kernel, Kernel-Erweiterungen, Baseband Firmware) verifiziert. Sollte eine Integritätsverletzung festgestellt werden, schlägt der Bootprozess fehl und das Gerät wird in den Wartungsmodus versetzt, und es erfolgt eine Wiederherstellung der Werkseinstellungen durch iTunes. Das Laden des Betriebssystems wird ebenfalls einer Verifikation hinsichtlich einer Manipulation unterzogen. Zusätzlich sorgt ein weiterer Mechanismus dafür, dass ein Downgrade auf eine frühere Betriebssystemversion, welche Schwachstellen enthalten kann, nicht möglich ist. [Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

HW.04	Vor dem Systemstart muss eine hardwareunterstützte Integritätsüberprüfung durchgeführt werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet seit Version 4.4 die Möglichkeit während des Boot-Prozesses Änderungen am Kernel zu detektieren. Zur Erkennung einer Manipulation werden blockweise Hashwerte berechnet, auf dessen Werte anschließend erneut Hash-Werte berechnet werden, bis es einen Wurzel-Hash-Wert ergibt. Dieser Wurzel-Hash wird durch das in der boot-Partition integrierte Hersteller-Zertifikat signiert. Damit ist gewährleistet, dass bei Manipulation eines Blocks die Verifikation fehlschlägt. Das in der Boot-Partition enthaltene Zertifikat wird mit einem unveränderlichen OEM Key der Hardware signiert. [IAq]

Schwachstelle: Es besteht theoretisch die Möglichkeit über eine Debugging-Schnittstelle (z.B. JTAG) Daten auf dem Flash-Speicher zu manipulieren. [Has]

Angriffsszenario: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

HW.04	**Vor dem Systemstart muss eine hardwareunterstützte Integritätsüberprüfung durchgeführt werden.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

B.2 Betriebssystem

Dieser Abschnitt thematisiert alle Anforderungen, welche das Betriebssystem betreffen, und ist unterteilt in die Bereiche *Allgemein*, *Systemsicherheit*, *Applikationssicherheit* und *Datensicherheit*.

Allgemein

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

Beschreibung: Der Zugriff auf sensible Informationen (z.B. Session-Token, geheime Dokumente, ...) darf bei Verlust des Gerätes in einem angemessenen zeitlichen Aufwand nicht möglich sein.

Allgemeine Hintergrundinformationen: Dem Schutz von auf einem mobilen Gerät gespeicherten Daten wird allgemein eine hohe Bedeutung beigemessen. Das Abrufen von E-Mails über ein gespeichertes Profil, die Nutzung von Messenger-Anwendungen, das Einrichten von VPN Zugängen oder dem Generieren und Empfangen von Einmalpasswörtern stellen eine Gefahr dar, sollten Unbefugte Zugriff auf diese Informationen erhalten.

Umsetzung iPhone 7: In iOS Version 10 werden grundsätzlich alle Informationen auf der Benutzerpartition verschlüsselt (dedizierte AES 256 Crypto Engine, dateibasierte Verschlüsselung). Die Daten des Betriebssystem liegen unverschlüsselt auf dem Speicher vor, werden jedoch bei jedem Bootvorgang in einer „Chain of Trust“ auf deren Integrität überprüft. Der Prozess der Integritätsüberprüfung wurde bereits im Grundlagenteil unter dem Punkt *sicherer Bootvorgang* erläutert. Um eine Extraktion der Speicherhardware vorzubeugen, werden alle Daten zusätzlich mit einem Hardwareschlüssel geschützt, der bei der Geräteherstellung in den Anwendungsspeicher eingebrannt, unveränderlich und nicht auslesbar ist.

Der Nutzer wird beim Setup des Gerätes aufgefordert, einen Geräteschutz festzulegen (PIN, Passwort, Fingerabdruck). Um Brute Force Angriffe gegen den Geräte-PIN zu erschweren, erhöht sich exponentiell je Fehlversuch die Zeitspanne, ehe eine erneute Eingabe möglich ist. Zusätzlich kann der Nutzer festlegen, dass nach zehn Fehlversuchen der Dateisystemschlüssel (siehe Abbildung 2.6 im Grundlagenteil) gelöscht wird und ein Entschlüsseln der Daten unmöglich wird.[Incm]

Schwachstelle: Im August 2017 wurde eine Schwachstelle während des Prozess zur Gerätewiederherstellung gefunden, welche eine automatisierte Eingabe des PINs über den Lightning-Anschluss ermöglicht und keine zeitliche Begrenzung bei Fehlversuchen enthält. Betroffen sind das iPhone 7 und iPhone 7 Plus in der Version 10.3.3.[Schc]

Angriffsszenario: Ein derartiges Gerätes wird derzeit für rund 500 US Dollar verkauft. Damit soll ein Entschlüsseln eines vierstelligen PIN Codes innerhalb weniger Tage möglich sein. Durch eine entschlüsseltes Gerät hat ein Angreifer Vollzugriff auf alle lokal gespeicherten Informationen.

Bewertung:

OS.01	Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Effizienter Brute-Force Angriff gegen Zugriffssperre durch Schwachstelle in Lightning-Anschluss.
Schwachstelle	
Entdeckung:	7 - Leicht
Ausnutzung:	3 - Kompliziert
Bekanntheit:	5 -
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	1 - Physikalisch
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	5.38 - Mittel
Gesamt Firma 1:	5.56 - Mittel
Gesamt Firma 2:	6.56 - Hoch

Umsetzung Samsung Galaxy S7: Android bietet die Möglichkeit alle Benutzerdaten zu verschlüsseln. Seit Android 5.0 wird eine „Full-disk encryption“ angeboten, welche in Version 7.0 durch eine dateibasierte Verschlüsselung ersetzt wurde. Die Nutzung der Verschlüsselung ist optional, es existieren jedoch Geräte, welche die Verschlüsselung standardmäßig aktiviert haben (z.B. Pixel Phone). Um sich vor Brute-Force Angriffen zu schützen, wird das Zeitintervall zur erneuten Eingabe nach jedem Fehlversuch erhöht. [IAe; IAf]

Schwachstelle: Es besteht die Möglichkeit über JTAG einen Dump des Flashspeichers zu Erstellen. Die Werksteinstellungen des vorliegenden Gerätes sehen keine Verschlüsselung vor. Durch einen Dump des Flashspeichers ist ein direktes Auslesen der Daten möglich. Alternativ können Inhalte der Partition *sdcard* über den USB Anschluss oder durch Aktivierung der Android Debug Bridge extrahiert werden.

Angriffsszenario: Ein Gerät, welches über keine aktivierte Dateiverschlüsselung verfügt, kann durch Dumpen des Flashspeichers einen Zugriff auf die gespeicherten Daten ermöglichen. Daten, welche mittels des Keystores verschlüsselt wurden, stehen nach einem Dump nur verschlüsselt zur Verfügung und ein Angriff gegen die vom Keystore verwendeten Algorithmen ist derzeit praktisch nicht durchführbar.

Bewertung:

OS.01	Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Zugriff auf unverschlüsselte Daten durch Dumpen des Flashspeichers.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	1 - Theoretisch
Bekanntheit:	6 - Offensichtlich
Transparenz:	1 - Sourcecode einsehbar und Dokumentation
Erkennung:	1 - Aktive Erkennung
Angriffsvektor:	1 - Physikalisch
Umfang:	3 - Lokal
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	2.88 - Niedrig
Gesamt Firma 1:	4.31 - Mittel
Gesamt Firma 2:	5.31 - Mittel

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

Beschreibung: Die Verfügbarkeit eines Browsers auf einem Smartphone eröffnet einem Angreifer vielfältige Angriffsmöglichkeiten. Im Gegensatz zu klassischen mobilen Anwendungen kann der Nutzer direkt eine URL aufrufen, wodurch sich beispielsweise lokale Cross Site Scripting Attacken durchführen lassen.

Anmerkung des Autors: Diese Anforderung betrachtet ausschließlich die von Hersteller / Betriebssystem ausgelieferten Standardbrowser.

Allgemeine Hintergrundinformationen: Sowohl Mobile- als auch Desktop-Browser besitzen mittlerweile Schnittstellen um mit Hardwaremodulen kommunizieren zu können.

Umsetzung iPhone 7: iOS liefert mit dem Betriebssystem seinen eigenen Browser *Safari* aus. Dieser unterstützt Javascript und verwendet Apples JavaScript JIT Compiler zur Ausführung. Unter Safari gespeicherte Passwörter werden in der Keychain aufbewahrt. Die Anwendung ist an das Betriebssystem gebunden und kann somit nicht über den AppStore aktualisiert werden.[Incm]
Apple erlaubt keine alternativen Browser-Engines. Alle im AppStore als *„Browser“* angebotenen Anwendungen nutzen als Basis die iOS *WKWebView* Schnittstelle.[Sok] Welche Module von Safari in einem privilegierten Kontext ausgeführt werden, ist nicht öffentlich dokumentiert.

Schwachstelle: In 2017 erschienen multiple Schwachstellen in der WebKit API[7], welche die Ausführung von Schadcode (CVE-2017-7042) oder eine Rechteausweitung (CVE-2017-2446) erlaubten. Es existierten Exploits auf dem Schwarzmarkt, welche mittlerweile frei zugänglich sind[8]. Betroffen war iOS 10.2 bis 10.3.2. Diese Schwachstellen wurden in iOS 10.3.3 behoben[Corb]. Aktuell existiert jedoch in iOS 10.3.3 eine Schwachstelle, welche Address Bar Spoofing[9] in Safari ermöglicht (CVE-2017-7085). Die gravierendste Schwachstelle (CVE-2017-7087) liegt im *Webkit Framework*, welche das Ausführen von Schadcode nach Aufruf einer bösartigen Webseite ermöglicht. Die *Webkit* Engine wird unter anderem von *Safari* genutzt wird.[Incc]

Angriffsszenario: Für die Schwachstelle im *WebKit Framework* existieren aktuelle keine Exploits sowie öffentliche Informationen über Details der Schwachstelle. Ein Angreifer ist jedoch in der Lage, über eine manipulierte Webseite Schadcode auszuführen. Inwiefern dieser Code mit privilegierten Rechten ausgeführt werden kann, konnte nicht eruiert werden.

[7] Schnittstelle wird u.a. von Safari verwendet.
[8] https://www.exploit-db.com/exploits/42364/
[9] Address Bar Spoofing: Anzeige der legitimen Addresse, obwohl eine falsche Seite aufgerufen wurde.[Incab]

Bewertung:

OS.02	Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	3 - Kompliziert
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	7 - Logging ohne Mitteilung
Angriffsvektor:	4 - Lokal
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	3.50 - Mittel
Gesamt Firma 1:	4.63 - Mittel
Gesamt Firma 2:	5.63 - Mittel

Umsetzung Samsung Galaxy S7: Samsungs vorinstallierter Browser *Samsung Internet* basiert auf der Chromium-Engine. Die Anwendung wird aus Betriebssystemsicht nicht privilegiert mit den Rechten einer gewöhnlichen Benutzeranwendung ausgeführt. Aktualisierungen werden über den AppStore von Android (Google PlayStore) bereitgestellt.

Schwachstelle: Samsung veröffentlicht auf seiner Mobile Security Seite ausschließlich Informationen über Betriebssystemupdates. Offizielle Informationen zu aktuellen Schwachstellen in Samsungs Standardanwendungen werden nicht veröffentlicht. Damit kann eine Bewertung nur auf Grundlage öffentlicher Informationen und bekannten Schwachstellen in der Chromium-Engine vorgenommen werden.
In den Veröffentlichungshinweisen zu *Samsung Internet 5.0* wurde als Engine Chromium in Version 51.2704 benannt. In der aktuellen Betaversion (6.2) wird Chromium 56 verwendet.[Coml]
In Chromium 61 (aktuellste Veröffentlichung) wurden 2 hohe Schwachstellen behoben (CVE-2017-5121, CVE-2017-5122).[Incaa]

Angriffsszenario: Ein Angreifer ist durch Ausnutzen dieser Schwachstelle in der Lage, Schadcode in einer Sandbox auszuführen, nachdem der Nutzer eine manipulierte Webseite aufgerufen hat. Die Engine isoliert jeden Tab einer Webseite in eine eigene Sandbox, welche keinen direkten Zugriff auf den Speicher oder andere Ressourcen hat.

Bewertung:

OS.02	Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Speicherzugriffsverletzung CVE-2017-5121, CVE-2017-5122
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	3 - Kompliziert
Bekanntheit:	4 - Verborgen
Transparenz:	1 - Sourcecode einsehbar und Dokumentation
Erkennung:	7 - Logging ohne Mitteilung
Angriffsvektor:	4 - Lokal
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	3.50 - Mittel
Gesamt Firma 1:	4.63 - Mittel
Gesamt Firma 2:	5.63 - Mittel

Systemsicherheit

R-OS.03

Anforderung: Die Updates des Betriebssystems müssen authentisch sein.

Beschreibung: Aktualisierungen für das Betriebssystem müssen authentisch bereitgestellt werden. Es muss gewährleistet sein, dass keine Manipulation des Updates auf dem Weg zwischen Server und Gerät erfolgt und dass die Aktualisierung von einer autorisierten Stelle angeboten wird.

Allgemeine Hintergrundinformationen: Die Authentizität von Aktualisierungen des Betriebssystems stellt eine elementare Voraussetzung für die Gerätesicherheit der Daten des Nutzers dar. Ohne derartiger Überprüfungen ist ein Angreifer in der Lage, Updates so zu manipulieren, dass Schadcode auf Betriebssystemebene eingespielt werden kann und damit (theoretisch) alle Sicherheitsmechanismen des Gerätes deaktiviert werden können. Um die Integrität von Aktualisierungen sicherzustellen, hat sich das Konzept der Signatur von Update-Dateien mit einem Zertifikat etabliert. Für die Authentizität wird das zur Signatur verwendete Zertifikat mit der Root-CA des Herstellers unterschrieben, dessen öffentlicher Schlüssel in einem sicheren Bereich auf dem Smartphone aufbewahrt wird (siehe Anforderung HW.04). Zusätzlich muss der Transportweg zwischen Server und Zielgerät über einen sicheren Kanal erfolgen.

Umsetzung iPhone 7: Apple bietet für Aktualisierungen die Möglichkeit, diese über iTunes oder „Over the Air (OTA)“ einzuspielen. Während bei einem iTunes Update das komplette Betriebssystem geladen und installiert wird, werden bei einem Update über OTA nur die von einem Update betroffenen Komponenten installiert. Für das Laden einer Aktualisierung sendet das Gerät eine kryptographische Liste mit Kennzahlen, einem Anti-Replay-Zufallswert und der eindeutigen Gerätekennung (ECID) an den Autorisierungsserver. Dieser prüft die übermittelten Werte, signiert diese und verifiziert, dass nur von Apple aktuell freigegebene Versionen geladen und installiert werden können. Damit wird neben einem Downgrade auf ältere Versionen auch die Installation weniger aktuellerer Versionen unterbunden.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.03	Die Updates des Betriebssystems müssen authentisch sein.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Aktuelle technische Details des Prozess zur Aktualisierung des Android-Betriebssystem sind bisher nicht offiziell veröffentlicht worden. Durch Analysen von Forscher konnte der Prozess weitestgehend nachgestellt werden. Je nach Konfiguration sendet das Gerät in definierten Zeitintervallen oder manuell Anfragen an den jeweiligen Updateserver (die meisten Hersteller verwenden den Updateserver von Android) und erhalten in der Antwort des Servers eine Mitteilung, ob ein Update vorliegt. Sollte eine Aktualisierung vorliegen, wird diese geladen und das Gerät im Rahmen des Aktualisierungsprozesses in den Wiederherstellungsmodus gebootet. Dort wird initial eine Überprüfung der Integrität und Authentizität vorgenommen. Sofern diese erfolgreich ist, werden die betroffenen Komponenten aktualisiert.[Prom]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. In einem kurzen Test wurde versucht die HTTP Kommunikation zwischen Smartphone und OTA-Server mitzuschneiden. Dies war erfolglos, da der Updateprozess mittels Certificate Pinning geschützt ist und tiefergehende Modifikationen am System hätten vorgenommen werden müssen, um diesen Mechanismus zu deaktivieren. Ein manuelles Downgrade auf eine ältere signierte Version, welche unter Umständen Schwachstellen enthalten kann, ist jedoch möglich.

Angriffsszenario: Ein Angreifer kann durch ein Downgrade auf eine ältere Android Version, welche nicht behobene Schwachstellen enthält, die Gerätesicherheit beeinträchtigen. Das Downgrade eines Gerätes setzt den physikalischen Zugriff voraus. Anschließend kann eine frühere (signierte) Version über *adb sideload* installiert werden.

Bewertung:

OS.03	Die Updates des Betriebssystems müssen authentisch sein.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Manuelles Downgrade auf frühere Version
Schwachstelle	
Entdeckung:	7 - Leicht
Ausnutzung:	5 - Leicht
Bekanntheit:	9 - Öffentlich
Transparenz:	1 - Sourcecode einsehbar und Dokumentation
Erkennung:	7 - Logging ohne Mitteilung
Angriffsvektor:	1 - Physikalisch
Umfang:	7 - Übergreifend
Nutzerinteraktion:	7 - Nicht erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	5.50 - Mittel
Gesamt Firma 1:	5.63 - Mittel
Gesamt Firma 2:	6.63 - Hoch

R-OS.04

Anforderung: Der Bootvorgang und das Laden aller für den Betrieb erforderlichen Komponenten müssen auf deren Integrität überprüft werden.

Beschreibung: Beim Start eines Betriebssystems müssen alle erforderlichen Komponenten auf deren Integrität überprüft werden, um eine Manipulation von Systemdateien zu erkennen.

Anmerkung des Autors: Diese Anforderung wurde geschaffen, sollte ein Gerät nicht die Anforderung HW.04 erfüllen. Nachdem beide vorliegenden Geräte über eine hardwaregestützte Integritätsüberpüfung verfügen, ist diese Anforderung obsolet.

R-OS.05

Anforderung: Das Betriebssystem soll auf unterschiedlichen Hardware Plattformen verwendbar sein.

Beschreibung: Die Portabilität eines Betriebssystems in einem heterogenen Hardwareumfeld bietet aus Sicherheitsüberlegungen den Mehrwert, dass Hardwarehersteller ausgewählt werden können, deren Hardwaredesign bereits Sicherheitsfeatures enthält und eine rasche Aktualisierung der Softwarekomponenten bei Bekanntwerden einer Schwachstelle zusichern.

Allgemeine Hintergrundinformationen: Die Abhängigkeit von einem Hardwareherstellers oder Anbieter von Smartphones kann für Gruppen mit einem hohen Sicherheitsbedarf von Nachteil sein, da keine Wettbewerbssituation vorhanden ist und eine Möglichkeit des Hardwarewechsels nicht besteht. Dies hat beispielsweise für Sicherheitsbehörden und Nachrichtendienste den Nachteil, dass erforderliche Modifikationen an der Hardware nicht möglich sind und eine Verwendung derartiger Anbieter nicht möglich ist.

Umsetzung iPhone 7: Apple's Betriebssystem iOS ist derzeit nur für von Apple produzierten iPhones verfügbar. Damit ist ein Wechsel der Hardware unter Beibehaltung des Betriebssystems nicht möglich.

Schwachstelle: Angriffsvektoren im klassischen Sinne sind bei dieser Anforderung nicht möglich.

Bewertung: Diese nicht-funktionale Anforderung muss gesondert für jede Anwendergruppe betrachtet werden, da sie ein k.o-Kriterium hinsichtlich der Nutzung darstellen kann.

Umsetzung Samsung Galaxy S7: Android ist ein Betriebssystem, dessen Source-Code öffentlich zur Verfügung gestellt wird. Damit ist eine Portierung auf eigene Hardwareanforderungen möglich.

Schwachstelle: Angriffsvektoren im klassischen Sinne sind bei dieser Anforderung nicht möglich.

Bewertung: Das Betriebssystem Android erfüllt diese Anforderung, da es eine Portierung auf individuelle Hardware zulässt.

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

Beschreibung: Privilegierte Applikationen werden in der Regel vom Betriebssystem oder Hersteller des Gerätes bei Auslieferung des Gerätes vorinstalliert. Sie übernehmen Aufgaben, welche nicht mit den Einschränkungen einer Drittanbieter-Anwendung, durchgeführt werden können (z.B. Geräte-Backup, AppStore). Der Schutz dieser Anwendungen ist von besonderer Bedeutung, da Schwachstellen in Systemanwendungen die Systemsicherheit gefährden können.

Umsetzung iPhone 7: Der Zugriff auf privilegierte Anwendungen in iOS wird bereits auf Kernel-Ebene unterbunden, da Nutzeranwendungen und System-Applikationen nicht unter dem gleichen Nutzer *mobile* ausgeführt werden. Zusätzlich werden System-Applikationen in der Betriebssystempartition verwaltet, welche im Betriebssystem als nur lesend eingebunden wird.

Schwachstelle: Es existiert eine Schwachstelle im *CoreAudio Framework*, welche das Auslesen von Informationen aus einem geschützten Bereich erlaubt (CVE-2017-0381). Die Schwachstelle wird von Apple bestätigt, eine öffentliche Dokumentation existiert jedoch nicht.[Incc]

Angriffsszenario: Ein Angreifer kann durch das Versenden einer entsprechend manipulierten Datei, Daten aus einem geschützten Bereich auslesen. Inwiefern dieser geschützte Bereich sich innerhalb des Audio Frameworks befindet oder Daten außerhalb eines Audio-Prozesses ausgelesen werden können, konnte nicht eruiert werden.

Bewertung:

OS.06	Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Schwachstelle im Audio Framework beim Abspielen von OPUS Dateien.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	1 - Theoretisch
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	4 - Lokal
Umfang:	7 - Übergreifend
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.00 - Mittel
Gesamt Firma 1:	4.88 - Mittel
Gesamt Firma 2:	5.88 - Mittel

Umsetzung Samsung Galaxy S7: Android empfiehlt Entwicklern die Verwendung der System UID zu vermeiden, da Anwendungen, welche als Systemnutzer ausgeführt werden, bei Schwachstellen innerhalb dieser Applikation die Systemsicherheit gefährden können. Sollte das Ausführen einer Anwendung mit Systemrechten erforderlich sein, sollen keine IPC Mechanismen (Service, Broadcast Empfänger, ContentProvider) für Nutzeranwendungen exportiert werden[10].[Prol]

Schwachstelle: Die von Samsung bereitgestellte Systemanwendung *SVoice* ermöglicht einem Angreifer das Ausführen von Schadcode mit privilegierten Rechten. Daneben existiert noch eine weitere Schwachstelle in von Samsung vorinstallierten Anwendungen (SVE-2017-9357), die jedoch ohne Systemrechte ausgeführt wird.
[Comi]

Angriffsszenario: Ein Angreifer kann durch die Sprachverarbeitung der Anwendung *SVoice* Audio Signale versenden, welche für das menschliche Gehör nicht wahrnehmbar sind, und damit das Ausführen von Schadcode erreichen.

[10] Anmerkung des Autors: Die Einschränkung wird durch Setzen des *protectionLevel* auf „*signature*“ oder „*signatureOrSystem*“ erreicht.

Bewertung:

OS.06	Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung *SVoice* mit Systemrechten.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	1 - Theoretisch
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	4 - Lokal
Umfang:	7 - Übergreifend
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.00 - Mittel
Gesamt Firma 1:	4.88 - Mittel
Gesamt Firma 2:	5.88 - Mittel

Applikationssicherheit

R-OS.07

Anforderung: Das Betriebssystem muss Anwendungen in sicherem Umfeld ausführen und vor Zugriffen von außen schützen.

Beschreibung: Das Betriebssystem soll Anwendungen in einem sicherem Umfeld ausführen und vor Zugriffen von außen schützen.

Allgemeine Hintergrundinformationen: Zur Trennung von Applikationen auf Prozessebene hat sich das *„Sandboxing"* etabliert. Das Konzept von *„Sandboxing"* basiert im Wesentlichen darauf, dass Applikationen von anderen isoliert werden und ein direkter Zugriff nicht mehr möglich ist. Damit Applikationen mit Anderen kommunizieren können, werden durch das Betriebssystem Mechanismen (Interprozesskommunikation) angeboten.

Umsetzung iPhone 7: iOS separiert alle Applikationen, welche nicht vom Betriebssystem zur Verfügung gestellt wurden, in einer Sandbox. Jeder Applikation wird ein eigenes Homeverzeichnis zur Verfügung gestellt, auf welches ausschließlich die Anwendung zugreifen kann. Die Applikationen werden unter dem nicht privilegierten Nutzer *mobile* ausgeführt. Um die Auswirkungen eines Buffer-Overflows im Falle einer Schwachstelle beim Sandboxing zu reduzieren, verwendet iOS zusätzlich *Address Space Layout Randomization (ASLR)* bei der Zuweisung des Speicherbereichs.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.07	Das Betriebssystem muss Anwendungen in sicherem Umfeld ausführen und vor Zugriffen von außen schützen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Das *Sandboxing* Konzept von Android findet bereits auf Kernel-Ebene statt, indem jeder Applikation ein eigener UNIX-Benutzer zugewiesen wird. Damit wird sichergestellt, dass Anwendungen untereinander nicht kommunizieren können oder Zugriff auf deren individuellen Daten erhalten. Seit Android 5.0 (Marshmallow) wird für die Speicherreservierung ebenfalls eine Randomisierung (ASLR) verwendet.[IAb; IAo; IAk]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. 2015 wurde durch MWR Info Security eine Möglichkeit des Ausbruchs aus einer Sandbox über eine privilegierte Applikation demonstriert[11]. Die Lücke wurde im August 2015 durch Android geschlossen.

Bewertung:

OS.07	Das Betriebssystem muss Anwendungen in sicherem Umfeld ausführen und vor Zugriffen von außen schützen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[11] https://labs.mwrinfosecurity.com/advisories/sandbox-bypass-through-google-webview/

R-OS.08

Anforderung: Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.

Beschreibung: Für die Kommunikation zwischen Applikationen auf einem Gerät soll das Betriebssystem Mechanismen anbieten, die eine sichere Kommunikation ermöglichen.

Allgemeine Hintergrundinformationen: Durch die Anforderung R-OS.12 wird gewährleistet, dass Applikationen nicht direkt kommunizieren können und ein Zugriff auf die individuellen Daten nicht möglich ist. Um berechtigte Kommunikation (z.B. E-Mailanwendung, welche auf Adressbuch zugreifen möchte) durchzuführen, soll das Betriebssystem auf Anwendungsebene geeignete Mechanismen anbieten.

Umsetzung iPhone 7: Das Betriebssystem iOS bietet zur Kommunikation von Applikationen untereinander die Möglichkeit, Dateien mittels *„Airdrop“* auszutauschen, Applikationen über *URL Schemes (z.B. tel://+49112)* aufzurufen oder über so genannte *App Extensions* individuelle Erweiterungen zu implementieren.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.08	**Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Das Betriebssystem Android bietet die Möglichkeit via *Binder*, *Services*, *Intents* oder *Content Provider* Daten auszutauschen. Die Funktionsweise wurde im Grundlagenteil (Kapitel 2.5) erläutert.

Schwachstelle: Es existieren diverse Schwachstellen (CVE-2017-0756 bis CVE-2017-0765) im *Media Framework* von Android, welche das Ausführen von Schadcode mit privilegierten Rechten ermöglichen[12].[Prog]

Angriffsszenario: Die im *Media Framework* vorhandenen Schwachstellen ermöglichen einem Angreifer, durch manipulierte Audiodateien, Schadcode in einem privilegierten Modus auszuführen. Damit können alle unverschlüsselten Daten auf dem Gerät korrumpiert werden. Es existieren derzeit keine öffentlich verfügbaren Exploits oder technische Dokumentationen dieser Schwachstellen hinsichtlich deren Ausnutzbarkeit.

Bewertung:

OS.08	**Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle im Media Framework
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	3 - Kompliziert
Bekanntheit:	4 - Verborgen
Transparenz:	1 - Sourcecode einsehbar und Dokumentation
Erkennung:	7 - Logging ohne Mitteilung
Angriffsvektor:	4 - Lokal
Umfang:	7 - Übergreifend
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.00 - Mittel
Gesamt Firma 1:	4.88 - Mittel
Gesamt Firma 2:	5.88 - Mittel

[12] Diese Schwachstellen wurden im Sicherheitsupdate vom 01.09.2017 behoben.[Prog]

R-OS.09

Anforderung: Eine Anwendung darf nur sicher mit berechtigten externen Schnittstellen kommunizieren.

Beschreibung: Die Nutzung und Verwendung externer Schnittstellen (z.B. Kamera, GPS Modul, NFC Modul) erhöhen den Komfort für den Nutzer bei der Erledigung von Aufgaben, sowohl im kommerziellen als auch im privaten Umfeld. Ein Missbrauch dieser Schnittstellen kann gravierende Auswirkungen auf die Vertraulichkeit und Privatsphäre von Nutzern mit sich bringen.

Allgemeine Hintergrundinformationen: Zur Autorisierung der Nutzung von Schnittstellen eignet sich die Implementierung einer Access Control List (ACL). Damit kann gewährleistet werden, dass nur berechtigte Anwendungen mit ausgewählten Schnittstellen kommunizieren dürfen.

Umsetzung iPhone 7: Unter iOS wird die Access Control List (ACL) unter dem Namen Capabilities geführt. Capabilities enthalten Entitlements, welche ein Privileg für den Zugriff auf eine Schnittstelle (z.B. Bluetooth), Funktionalität (z.B. App Groups) oder Modul (z.B. Apple Pay) beinhaltet. Manche Entitlements erlauben zur Laufzeit eine feinere Spezifikation vorzunehmen. So kann beispielsweise beim Zugriff auf GPS festgelegt werden, ob dieser komplett verweigert, auf die aktive Nutzung der App begrenzt oder im Hintergrund genutzt werden darf.[Incd; Ince]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.09	Eine Anwendung darf nur sicher mit berechtigten externen Schnittstellen kommunizieren.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bezeichnet sein Sicherheitsmodell zum Schutz des Zugriffs auf externe Schnittstellen als *Android Permission Model*. Die Zuteilung eines derartigen Privilegs wird im Manifest einer Applikation festgelegt. Bevor ein Benutzer die Applikation installiert, werden ihm alle Rechte dieser Anwendung angezeigt, dessen Zustimmung Voraussetzung zur Installation ist. Nach der Installation besteht seit Android 6.0 die Möglichkeit zur Laufzeit die Berechtigungen anzupassen. Schnittstellen, welche Kosten produzieren können (z.B. Telefonie-Funktion), muss vor einer erstmaligen Nutzung explizit zugestimmt werden. Damit soll eine unbemerkte Nutzung kostenintensiver Schnittstellen durch eine Angreifer-Anwendung vermieden werden.[IAI; Proj]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.09	**Eine Anwendung darf nur sicher mit berechtigten externen Schnittstellen kommunizieren.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-OS.10

Anforderung: Eine Anwendung darf nur möglichst feingranulare Berechtigungen für den Zugriff auf andere Anwendungen erhalten.

Beschreibung: Die in Anforderung R-OS.08 beschriebene Interprozesskommunikation soll um eine Autorisierungskomponente erweitert werden. Damit soll vermieden werden, dass ein Angreifer unkontrolliert IPC Mechanismen ausnutzt, um Daten einer Applikation zu korrumpieren.

Allgemeine Hintergrundinformationen: Dem Schutz von IPC Schnittstellen wird im Allgemeinen eine hohe Priorität zugewiesen, da diese Schnittstellen eine Möglichkeit darstellen, ohne Benutzerinteraktion Daten an eine Anwendung zu übermitteln oder auszulesen.

Umsetzung iPhone 7: Um Daten an eine Anwendung über IPC Mechanismen zu transportieren, besteht die Möglichkeit eine Anwendung über einen URL Scheme Handler und entsprechenden Parametern aufzurufen. Für den Austausch von Dokumenten steht AirDrop zur Verfügung. Für komplexere Kommunikation zwischen Anwendungen müssen individuelle App Extensions implementiert werden. Auf Betriebssystemebene werden diese Erweiterungen in einem eigenen Container und Sandbox ausgeführt. Damit wird ein direkter Zugriff auf den Speicher und die Daten der Anwendungen verhindert[13].[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. In iOS 8.4.1 bestand eine kritische Schwachstelle inklusive eines öffentlichen Exploits, welcher mittels AirDrop die Installation von Malware über das Netzwerk zuließ. Diese Schwachstelle wurde in iOS 9 behoben[Bec].
Es besteht die Möglichkeit, dass bei der Implementierung von App Extensions und bei der Verarbeitung von Dateien, welche via AirDrop empfangen wurden, Schwachstellen existieren. Das Ermitteln und Bewerten derartiger Schwachstellen ist Teil eines Penetrationstests.

Bewertung:

OS.10	Eine Anwendung darf nur möglichst feingranulare Berechtigungen für den Zugriff auf andere Anwendungen erhalten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[13] Anmerkung des Autors: Die Sicherheit des Informationsfluss zwischen zwei Anwendungen kann nur individuell durch einen Penetrationstest verifiziert werden.

Umsetzung Samsung Galaxy S7: Die Grundlagen der Interprozesskommunikation wurden bereits im Grundlagenteil (siehe Kapitel 2.5) erläutert. Im Folgenden werden angebotene Mechanismen zum Schutz dieser Mechanismen erläutert.

Für Applikationen, welche einen eigenen *ContentProvider* verwenden, besteht die Möglichkeit den Zugriff auf diese Daten einzuschränken. Die Einschränkung erfolgt mittels des *„protection level“*:

- normal: Anwendungen erhalten automatisch Zugriff
- dangerous: Anwendungen erhalten nach vorheriger expliziter Zustimmung Zugriff
- signature: Nur Anwendungen, welche mit dem gleichen Zertifikat signiert wurden, erhalten Zugriff

[Pron]

Ferner besteht die Möglichkeit, neben den vorhandenen Standardberechtigungen eigene Berechtigungen zu implementieren. Android empfiehlt dieses Vorgehen in den Entwicklerhinweisen jedoch nicht. Durch eigene Berechtigungen kann beispielsweise festgelegt werden, dass beim Aufruf von kostspieligen Schnittstellen (Telefonie / SMS) keine explizite Zustimmung des Nutzers erforderlich ist. Die Sicherheit dieser Berechtigung ist ebenfalls abhängig vom *„protection level“*, welcher die gleichen Attribute wie ContentProvider verwendet.

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. Es besteht die Möglichkeit, dass bei der Implementierung von *ContentProvider* und eigenen Berechtigungen, Schwachstellen existieren. Die Detektion und Bewertung derartiger Schwachstellen ist Teil eines Penetrationstests.

Bewertung:

OS.10	**Eine Anwendung darf nur möglichst feingranulare Berechtigungen für den Zugriff auf andere Anwendungen erhalten.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-OS.11

Anforderung: Das Betriebssystem muss eine sichere Netzwerkkommunikation für Anwendungen anbieten.

Beschreibung: Die Übertragung von Informationen zwischen Smartphone und Server (Backend) sollen die Vertraulichkeit und Authentizität der Informationen zwischen Empfänger und Sender gewährleisten.

Allgemeine Hintergrundinformationen: Um die Vertraulichkeit von Informationen während des Transport sicherzustellen, hat sich die Verschlüsselung der Übertragung bewährt. Das am häufigsten verwendete Verfahren ist TLS (Transport Layer Security, früher Secure Socket Layer). Das Verfahren nutzt ein asymmetrisches Verfahren zum Schlüsselaustausch. Durch die Signierung der öffentlichen Schlüssel mit vertrauenswürdigen Zertifizierungsstellen wird sichergestellt, dass die Identität des Empfängers nicht manipuliert werden kann. Nach erfolgreichem Schlüsselaustausch erfolgt die weitere Kommunikation symmetrisch.

Umsetzung iPhone 7: iOs unterstützt für den Schlüsselaustausch TLS in den Versionen 1.0, 1.1 und 1.2. Für die folgende symmetrische Verschlüsselung kann AES mit einer Schlüssellänge von 128 Bit oder SHA-2 genutzt werden. Unsichere Verfahren wie SSLv3 und RC4 können ab iOS 10 nicht mehr verwendet werden. Die Gegenseite muss Forward Secrecy unterstützen, um einen Verbindungsaufbau zu ermöglichen. Alle oben genannten Anforderungen sind standardmäßig aktiviert, können aber durch den Entwickler explizit deaktiviert werden.[Incm]

Schwachstelle: Das BSI veröffentlichte 2015 ein TLS/SSL Best Practice Dokument mit Empfehlungen zur Konfiguration von TLS Verbindungen. TLS 1.0 wird nicht mehr als sicher angesehen.[Sici]

Bewertung: iOS unterstützt die aktuellen Implementierungen von TLS. Eine erweiterte Risikobewertung dieser Anforderung kann nur durchgeführt werden, wenn die unterstützen TLS-Versionen der Gegenseite (Server) aufgelistet werden. Eine Analyse der TLS-Unterstützung aller verwendeten Server dieses Smartphones erlaubt der zeitlichen Rahmen dieser Arbeit nicht und beinhaltet rechtliche Widrigkeiten bei Durchführung eines SSL-Scan.

Umsetzung Samsung Galaxy S7: Android unterstützt TLS in den Version 1.0, 1.1 und 1.2. SSLv3 wird seit Android 5.1 nicht mehr unterstützt. Die Verwendung von Forward Secredy wird bevorzugt. Unsichere Hash-Algorithmen wie MD5 oder 3DES werden ebenfalls nicht mehr unterstützt.[IAk]
Die erlaubten Zertifizierungsstellen wurden unter Android 7.0 (Nougat) für alle Hersteller konsolidiert. Standardmäßig kommt eine Verbindungen nur noch zustande, wenn das Serverzertifikat von einer der nativ enthaltenen Zertifizierungsstellen signiert wurde. Vom Nutzer händisch importierte oder vom Hersteller mitgelieferte Zertifizierungsstellen wird nur nach expliziter Freigabe im Manifest einer Anwendung vertraut.[IAm]

Schwachstelle: Das BSI veröffentlichte 2015 ein TLS/SSL Best Practice Dokument mit Empfehlungen zur Konfiguration von TLS Verbindungen. TLS 1.0 wird nicht mehr als sicher angesehen.[Sici]

Bewertung: Android unterstützt die aktuellen Implementierungen von TLS. Eine erweiterte Risikobewertung dieser Anforderung kann nur durchgeführt werden, wenn die

unterstützen TLS-Versionen der Gegenseite (Server) aufgelistet werden. Eine Analyse der TLS-Unterstützung aller verwendeten Server dieses Smartphones erlaubt der zeitlichen Rahmen dieser Arbeit nicht.

Datensicherheit

R-OS.12

Anforderung: Das Betriebssystem muss individuellen Speicher einer Anwendung vor externen Zugriff schützen.

Beschreibung: Das Betriebssystem soll Daten einer Anwendung kapseln und den Zugriff darauf vor anderen Applikationen schützen.

Allgemeine Hintergrundinformationen: Dem Schutz individueller Applikationsdaten wird eine hohe Priorität zugesprochen, da andernfalls die Vertraulichkeit und Integrität von sensitiven Nutzerdaten einer Anwendung nicht gewährleistet ist. Ohne die Trennung von Applikationsdaten könnte eine Angreiferanwendung vertrauliche Daten einer anderen Applikation auslesen. Zur Realisierung dieser Anforderung etablierte sich die Verwendung individueller Datenverzeichnisse für jede Applikation, deren Zugriff implizit auf die jeweilige Anwendung beschränkt ist.

Umsetzung iPhone 7: iOS stellt standardmäßig jeder Applikation ein eigenes Homeverzeichnis [14] zur Verfügung, auf das nur die jeweilige Anwendung Zugriff erhält.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.12	Das Betriebssystem muss individuellen Speicher einer Anwendung vor externen Zugriff schützen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android stellt jeder Anwendung ein privates und exklusives Verzeichnis zur Verfügung[15].[IAb]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.12	Das Betriebssystem muss individuellen Speicher einer Anwendung vor externen Zugriff schützen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[14] Beispiel: /private/var/mobile/Containers/Data/Application/com.optima.example/
[15] Beispiel: /data/data/com.optima.example/

R-OS.13

Anforderung: Das Betriebssystem muss eine Schnittstelle anbieten, um sensitive Daten besonders sicher aufzubewahren.

Beschreibung: Das Betriebssystem soll neben der in der Anforderung R-OS.12 geforderten Kapselung, sensitive Daten zusätzlich schützen, um im Falle einer Schwachstelle bei der Separation von Anwendungen die Vertraulichkeit für diese Informationen weiterhin zu gewährleisten. Die in den Betriebssystemen vorhandenen Schnittstellen wurden bereits in der Anforderung R-HW.03.2 beschrieben und werden keiner weiteren Bewertung unterzogen.

Anmerkung des Autors: Diese Anforderung wurde geschaffen, sollte ein Gerät nicht die Anforderung HW.03.2 erfüllen.

R-OS.14

Anforderung: Das Betriebssystem muss alle Benutzerdateien verschlüsselt aufbewahren.

Beschreibung: Die Anforderung dass alle Dateien auf dem Gerät verschlüsselt gespeichert werden sollen, wurde bereits einer Analyse und Bewertung in der Anforderung R-HW.03.1 unterzogen.
Bei einer Bewertung eines anderen Smartphones, welches über keinen hardwaregetrennten Bereich verfügt, muss die konkrete Umsetzung dieser Anforderung näher untersucht werden.

Umsetzung iPhone 7: iOS nutzt zur Verschlüsselung eine dediziertes Krypto-Modul, welches bereits in HW.03.1 einer Bewertung unterzogen wurde.

Umsetzung Samsung Galaxy S7: Die Verschlüsselung von Android auf Betriebssystemebene wurde bereits im Grundlagenteil (Kapitel 2.5) erklärt. Android nutzt seit Version 7.0 eine dateibasierte Verschlüsselung und dem AES-Algorithmus. Die Schlüssel haben eine Länge von 256 Bit und werden im Keystore verwaltet.[IAe]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

OS.14	**Das Betriebssystem muss alle Benutzerdateien verschlüsselt aufbewahren.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

B.3 Management

Dieser Abschnitt thematisiert alle Anforderungen, welche das Management von Geräten betreffen, und ist unterteilt in die Bereiche *Allgemein*, *Laufzeitanalyse*, *Mobile Device Management* und *Transparenz*.

Allgemein

R-MGMT.01

Anforderung: Das Gerät muss Schutzmechanismen anbieten, die einen ungeprüften und unautorisierten Zugriff verhindern.

Beschreibung: Damit Unbefugte im Falle eines Geräteverlustes keinen Zugriff auf Anwendungen oder Daten des Besitzers erhalten, sollen geeignete Schutzmechanismen vorhanden sein.

Allgemeine Hintergrundinformationen: Das Smartphone als ständiger Begleiter ist einer größeren Gefahr des Verlusts oder Diebstahl ausgesetzt als ein herkömmlicher Computer. Der stete Anstieg der Nutzung des Smartphones sowohl im privaten (Online Banking) als auch im geschäftlichen (z.B. Mailzugriff, VPN) Umfeld sorgt für eine Erhöhung der Auswirkungen, sollten Unbefugte Zugriff auf die auf dem Gerät gespeicherten Daten erlangen.

Umsetzung iPhone 7: iOS bietet Nutzern die Möglichkeit bei jeder Displayentsperrung durch Eingabe eines PIN, Passworts oder Fingerabdruck zu überprüfen, ob der rechtmäßige Besitzer dieses Gerät entsperrt. Um Brute-Force Angriffe abzuwehren, erhöht sich inkrementell die Zeit bis zur nächsten Eingabe des Gerätepasswortes nach einer bestimmten Anzahl von Fehlversuchen. Falls aktiviert, werden nach zehn Fehlversuchen alle Benutzerdaten gelöscht. Für den Fall des Geräteverlustes besteht die Möglichkeit bei noch vorhandener Internetverbindung, das Gerät über die Anwendung *Find My iPhone* via GPS zu lokalisieren. Diese Anwendung bietet zusätzlich das Abspielen eines lauten akustischen Tons oder die komplette Sperrung des Gerätes an. Ein Entsperren des Gerätes ist durch die iCloud Zugangsdaten möglich. Die Entsperrung der Benutzerdaten ist ausschließlich durch Kenntnis des PINs / Passwortes / Fingerabdrucks möglich.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.01	Das Gerät muss Schutzmechanismen anbieten, die einen ungeprüften und unautorisierten Zugriff verhindern.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet dem Nutzer die Möglichkeit den Zugriff auf sein Gerät durch Eingabe einer PIN, Passwort oder Fingerabdruck zu schützen. Um Brute-Force Angriffe zu vermeiden, erhöht sich nach einer definierten Anzahl von Fehlversuchen inkrementell das Zeitintervall zur erneuten Eingabe des Gerätepasswortes. Sollte das Geräte verloren gehen, kann durch die seit Android 5.1 bestehende *„Android Factory Reset Protection“* verhindert werden, dass ein Unbefugter durch Zurücksetzen des Gerätes das Smartphone nutzen kann. Diese Option setzt eine Synchronisation mit dem Google Account und das Konfigurieren eines Geräteschutzes (PIN, Passwort, Fingerabdruck) voraus. Das Entsperren des Bootloaders und die Installation eines alternativen Betriebssystems (z.B. LineageOS) wird verhindert, da vorab in den Entwickleroptionen die Entsperrung des Bootloaders aktiviert werden muss.[Incv]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. In der Vergangenheit wurden Schwachstellen veröffentlicht, die ein Umgehen der *Android Factory Reset Protection* ermöglichten. Aktuell lässt sich nur durch ein Downgrade auf Android 5.0 die Sperre entfernen[16]. Nachdem für das vorliegende Gerät keine von Samsung signierten Version von Android 5.0 existiert, ist diese Methode unwirksam.

Bewertung:

MGMT.01	Das Gerät muss Schutzmechanismen anbieten, die einen ungeprüften und unautorisierten Zugriff verhindern.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[16] Das Samsung Galaxy S7 wurde mit Android 6.0 veröffentlicht.

R-MGMT.02

Anforderung: Es muss ein Prozess zur Überprüfung von Anwendungen vor Veröffentlichung in einem App-Store vorhanden sein.

Beschreibung: Der AppStore des Betriebssystems soll Anwendungen vor Veröffentlichung in diesem einer Überprüfung unterziehen, um zu verhindern, dass eine Anwendung mit Schadcode aus offiziellen Quellen installiert werden kann.

Allgemeine Hintergrundinformationen: Die freie Installation aller für die Plattform kompilierten Anwendungen führt zu diversen Nachteilen. Nutzern wird die Unterscheidung zwischen der „echten“ und einer gefälschten Anwendung erschwert. Ein Nutzer kann ferner nicht überprüfen, ob in der Anwendung bösartiger Schadcode enthalten ist. Klassische Virenscanner sorgen nicht für Abhilfe, da der Hersteller derartiger Software erst eine bösartige Anwendung detektieren muss, eine Signatur berechnet und diese dem Virenscanner über ein Update zur Verfügung stellen muss.
Die Einführung eines Prozesses zur Überprüfung von Anwendungen ehe sie durch einen Nutzer installiert werden kann, sorgt für eine vorherige Überprüfung der Anwendung auf Schadcode.

Umsetzung iPhone 7: Apps, die im iOS AppStore veröffentlicht werden sollen, werden einer gründlichen Untersuchung unterzogen. Der Prozess der Überprüfung ist nicht öffentlich dokumentiert. In einem Selbstversuch des Autors dauerte es knapp 3 Wochen vom Upload der Anwendung bis zur ersten Freigabe. Den Log Files des in der Anwendung verwendeten Backends ist anzunehmen, dass neben automatisierten Tests ein manuelles Review erfolgt.

Schwachstelle: In der Vergangenheit erschienen Meldungen über bösartige Anwendungen im iOS AppStore, welche nach Bekanntwerden zügig entfernt wurden[17].

Bewertung: Eine Risikobewertung dieser Anforderung ist nur bei ausreichender Evidenz zu Auswertungen von in Apple's AppStore abrufbaren Anwendungen möglich. Trotz intensiver Recherche konnte hierzu nicht ausreichend Literatur ermittelt werden.

Umsetzung Samsung Galaxy S7: Android überprüft vor Veröffentlichung einer Anwendung im PlayStore den Inhalt auf Schadsoftware. Dieser Prozess läuft automatisiert ab. Eine Dokumentation der Validierung ist nicht öffentlich abrufbar[Prol]. In einem Selbsttest war die Anwendung nach dem Upload wenige Stunden später im PlayStore aufgelistet und installierbar.

Schwachstelle: Die Sicherheitsfirma TrendMicro veröffentlichte im Juni 2017 einen Report, der über 800 Anwendungen im PlayStore aufzählte, welche Malware enthielt.[Incac]

Bewertung: Eine Risikobewertung dieser Anforderung ist nur bei ausreichender Evidenz zu Auswertungen von in Apple's AppStore abrufbaren Anwendungen möglich. Trotz intensiver Recherche konnte hierzu nicht ausreichend Literatur ermittelt werden.

[17] https://www.heise.de/mac-and-i/meldung/Gefaelschte-Shopping-Apps-fluten-Apples-App-Store-3458658.html

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

Beschreibung: Das Betriebssystem soll die Deinstallation von (vorinstallierten) Standardanwendungen ermöglichen.

Allgemeine Hintergrundinformationen: Die Möglichkeit des Entfernens von Standardanwendungen ermöglicht einem Nutzer aus IT-Sicherheitsperspektive die Angriffsfläche zu reduzieren, da nicht benötigte Anwendungen deinstalliert werden können. Ferner besteht die Möglichkeit bei Bekanntwerden einer Schwachstelle, diese Anwendung temporär zu entfernen, um die Gerätesicherheit zu wahren.

Umsetzung iPhone 7: Das Betriebssystem iOS bietet die Möglichkeit vorinstallierte Anwendungen (vom Home-Screen) zu entfernen. Damit ist ein Aufruf dieser Anwendung erst nach „Installation“ aus dem AppStore wieder möglich. Technisch werden Anwendungs- und Konfigurationsdaten gelöscht, die Anwendung selbst verbleibt weiterhin auf dem Betriebssystem. In iOS 10 werden das Betriebssystem und die Standard-Applikationen zusammen signiert. Um technisch eine komplettes Entfernen dieser Applikationen zu ermöglichen, wäre eine Trennung von Betriebssystem und diesen Anwendungen erforderlich, da andernfalls die Signatur des Betriebssystems sich verändern und ein Systemstart nicht mehr erfolgreich wäre.[Inco]

Schwachstelle: In der Vergangenheit existierten für verschiedene Standardanwendungen (auch kritische) Schwachstellen (z.B. Safari CVE-2017-7042). In der vorliegenden Version 10.3.3 existieren diverse kleinere Schwachstellen von Standardanwendung *iBooks (CVE-2017-)*, *Phone (CVE-2017-7139)* und *Messages (CVE-2017-7118)*.

Die gravierendste Schwachstelle (CVE-2017-7087) liegt im *Webkit Framework*, welche das Ausführen von Schadcode nach Aufruf einer bösartigen Webseite ermöglicht. Die *Webkit Engine* wird unter anderem von *Safari* genutzt wird.[Incc]

Angriffsszenario: Für die Schwachstelle im *WebKit Framework* existieren aktuelle keine Exploits sowie öffentliche Informationen über Details der Schwachstelle. Ein Angreifer ist jedoch in der Lage, über eine manipulierte Webseite Schadcode auszuführen. Inwiefern dieser Code mit privilegierten Rechten ausgeführt werden kann, konnte nicht eruiert werden.

Bewertung:

MGMT.03	Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	3 - Kompliziert
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	7 - Logging ohne Mitteilung
Angriffsvektor:	4 - Lokal
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	3.50 - Mittel
Gesamt Firma 1:	4.63 - Mittel
Gesamt Firma 2:	5.63 - Mittel

Umsetzung Samsung Galaxy S7: Android bietet ohne Root Privilegien keine Möglichkeit vorinstallierte Anwendungen zu deinstallieren. Im Selbsttest bestand als einzige Möglichkeit eine Anwendung zu deaktivieren. Damit wird die Applikation und gegebenenfalls durch sie initiierte Hintergrundanwendungen an der Ausführung gehindert. In der Architektur von Android sind Standardanwendungen vom Betriebssystem abgekoppelt und können über den AppStore von Google aktualisiert werden.

Schwachstelle: Die Recherche nach verwundbaren Applikationen beschränkt sich auf die von Samsung entwickelten Standardanwendungen. Bei Auslieferung eines Gerätes sind eine Reihe weiterer Anwendungen vorinstalliert (z.B. Dropbox, Amazon, Instagram). Die vorinstallierte Samsung Applikation *SVoice* weist eine Schwachstelle auf, die es ermöglicht Schadcode auszuführen.[Comi]

Angriffsszenario: Ein Angreifer kann durch die Sprachverarbeitung der Anwendung *SVoice* Audio Signale versenden, welche für das menschliche Gehör nicht wahrnehmbar sind, und damit das Ausführen von Schadcode erreichen.

Bewertung:

MGMT.03	**Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung *SVoice* mit Systemrechten.
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	1 - Theoretisch
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	9 - Unprotokolliert
Angriffsvektor:	4 - Lokal
Umfang:	7 - Übergreifend
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.00 - Mittel
Gesamt Firma 1:	4.88 - Mittel
Gesamt Firma 2:	5.88 - Mittel

R-MGMT.04

Anforderung: Der Gerätenutzer darf Anwendungen ausschließlich aus sicheren Quellen installieren können.

Beschreibung: Um die Authentizität von Anwendungen sicherzustellen, sollen Anwendungen über eine zentrale vertrauenswürdige Stelle (z.B. AppStore) veröffentlicht werden.

Allgemeine Hintergrundinformationen: Das Abrufen und Installieren von Anwendungen über einen *AppStore* verhindert, dass Nutzer durch Aufrufen eines präparierten Links aus beispielsweise einer Mail, nicht direkt die Anwendung aus einer unsicheren Quelle installieren können. Der Aufwand eines Angreifers, manipulierte Anwendungen über den *AppStore* bereitzustellen, erhöht sich, da er die Entdeckung, von in einer Anwendung implementierten Malware, durch die stattfindende Überprüfung des *AppStore's*, verhindern muss. Sollte die bösartige Anwendung eine Fälschung der Original-Applikation sein, muss zusätzlich die Autorisierung als legitimierter Entwickler umgangen werden.

Umsetzung iPhone 7: iOS bietet ausschließlich die Installation von Anwendungen über den mitgelieferten *AppStore* an. Damit wird gewährleistet, dass Anwendungen grundsätzlich vor der Veröffentlichung durch automatisierte und manuelle Reviews überprüft werden. Die Identität des Entwicklers wird bei Registrierung eines Entwicklerzertifikats durch Apple überprüft. Im Selbsttest bei der Firma Optima fand eine Verifikation des Unternehmens durch die eindeutige Steuernummer des Unternehmens und einer telefonischen Überprüfung der Angaben statt.

Schwachstelle: In der Vergangenheit erschienen Meldungen über bösartige Anwendungen im iOS AppStore, welche nach Bekanntwerden zügig entfernt wurden[18].
Das Testen der Schutzmechanismen zur Verifikation des Unternehmens konnte aus rechtlichen Gründen nicht durchgeführt werden.

Bewertung: Eine Risikobewertung dieser Anforderung ist nur bei ausreichender Evidenz zu Auswertungen von in Apple's AppStore abrufbaren Anwendungen möglich. Trotz intensiver Recherche konnte hierzu nicht ausreichend Literatur ermittelt werden.

Umsetzung Samsung Galaxy S7: Android empfiehlt die Installation von Anwendungen über den *Play Store* durchzuführen. Alternativ existiert die Möglichkeit durch Aktivieren der Option *Installation aus unbekannten Quellen*, welche bei Auslieferung standardmäßig deaktiviert ist, Anwendungen durch Öffnen einer kompilierten Anwendung im APK Format auf dem Gerät zu installieren. Anwendungen, welche über letztere Option installiert wurden, sind nicht von Google überprüft worden und stellen für einen Angreifer die einfachste Option dar, bösartige Anwendungen zu verbreiten.[Proa]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden. Ein Angreifer könnte versuchen Malware über den *Play Store* zu verbreiten, indem die Malware so verändert wird, dass sie bei der Überprüfung durch Google nicht erkannt wird. Die Sicherheitsfirma TrendMicro veröffentlichte im Juni 2017 einen Report, der über 800 Anwendungen im PlayStore aufzählte, welche Malware enthielt.[Incac] Alternativ könnte mittels Social Engineering der Benutzer dazu genötigt werden, die Installation von Anwendungen durch unbekannte Quellen zu aktivieren und im Anschluss eine modifizierte Applikation über einen Mailanhang oder Link zu installieren.

[18] https://www.heise.de/mac-and-i/meldung/Gefaelschte-Shopping-Apps-fluten-Apples-App-Store-3458658.html

Bewertung: Eine Risikobewertung dieser Anforderung ist nur bei ausreichender Evidenz zu Auswertungen von auf Android Geräten installierter Malware möglich. Trotz intensiver Recherche konnte derzeit nicht ausreichend Literatur ermittelt werden. Es besteht jedoch die Möglichkeit, dass durch die Einführung von *Google Play Protecthttps://www.android.com/play-protect/* künftig Statistiken zu Anzahl und Umfang von infizierten Anwendungen veröffentlicht werden.

R-MGMT.05

Anforderung: Das Gerät muss nativ eine sichere Mailkommunikation anbieten.

Beschreibung: Für den Versand von Mails mit sensiblen Inhalt ist eine Verschlüsselung der Nachrichten zwingend erforderlich. Aufgrund der hohen Verbreitung von Mails vor allem im geschäftlichen Umfeld, soll durch das Betriebssystem nativ eine Unterstützung existieren.

Allgemeine Hintergrundinformationen: Für die Verschlüsselung des Nachrichteninhalts einer E-Mail haben sich zwei Verfahren etabliert:

Pretty Good Privacy (PGP) ist ein asymmetrisches Verschlüsselungsverfahren, welches von Phil Zimmermann entwickelt wurde. Das Verfahren bietet die Möglichkeit Nachrichten zu signieren (Gewährleisten der Integrität) und zu verschlüsseln (Gewährleisten der Vertraulichkeit).[TJ11]

Secure / Multipurpose Internet Mail Extensions (S/MIME) ist ein Standard zur Verschlüsselung von E-Mail Nachrichten. Das Verfahren verwendet eine hybride Verschlüsselung, welche den Nachrichteninhalt mittels eines zufälligen symmetrischen Schlüssel verschlüsselt. Diese Schlüssel werden anschließend durch das asymmetrische Verfahren verschlüsselt.[LS07]

Umsetzung iPhone 7: Die Standardanwendung von iOS für den Versand elektronischer Nachrichten unterstützt nativ die Signierung und Verschlüsselung von Mails mit S/MIME.[Incm]

Schwachstelle: Bei der Konfiguration eines Exchange Konto existierte eine Schwachstelle, welche einem Angreifer in die Lage versetzte, das für den Mailzugriff erforderliche Passwort zu extrahieren. Der Angriff nutzt eine Fehlkonfiguration im Autodiscover Protokoll von Exchange[NG17]. Aktuell existiert eine Schwachstelle, welche zu einem Absturz der Mail-Anwendung führt, wenn eine manipulierte Bilddatei geöffnet wird (CVE-2017-7097).[Incc]

Bewertung:

MGMT.05	Das Gerät muss nativ eine sichere Mailkommunikation anbieten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Denial of Service durch manipulierten Bilddateianhang (CVE-2017-7097)
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	3 - Kompliziert
Bekanntheit:	1 - Unbekannt
Transparenz:	4 - Dokumentation ohne Hintergrundinformationen
Erkennung:	1 - Aktive Erkennung
Angriffsvektor:	9 - Netzwerk
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	3.38 - Mittel
Gesamt Firma 1:	4.56 - Mittel
Gesamt Firma 2:	5.56 - Mittel

Umsetzung Samsung Galaxy S7: Android unterstützt nativ keine Unterstützung von S/MIME oder PGP. Für Enterprise Kunden von Google wird eine cloudbasierte „*G-Suite*“ zur Verfügung gestellt. Diese ermöglicht die Signierung und Verschlüsselung von Nachrichten ab dem Zeitpunkt des Versands von einem Google-Mailserver[Incz]. Das S/MIME Zertifikat wird nicht auf dem Gerät vorgehalten und es findet keine vollständige Ende-zu-Ende Verschlüsselung statt.

Schwachstelle: Für die Umsetzung unter Android existieren keine Angriffsvektoren, da eine native Implementierung dieser Anforderung fehlt. Im *Google PlayStore* werden zahlreiche Anwendungen angeboten, welche eine Verschlüsselung und Signatur von Mails mit S/MIME und PGP ermöglichen.

Bewertung: Die Bewertung dieser Anforderung kann nur vorgenommen werden, wenn eine konkrete Drittanbieteranwendung vorliegt.

Laufzeitanalyse

R-MGMT.06

Anforderung: Das Gerät muss nach dem Systemstart und im laufenden Betrieb auf Sicherheitsaktualisierungen überprüft werden.

Beschreibung: Eine automatisierte Nachfrage bei den Updateservern über das Vorliegen von Aktualisierungen, erhöht die schnellere Verbreitung von Updates für betroffene Geräte. Dazu soll ein Mechanismus existieren, der zeitnah und regelmäßig nach Aktualisierungen sucht.

Allgemeine Hintergrundinformationen: Nach der subjektiven Einschätzung und Erfahrung des Autors, überprüfen nur sehr wenige Nutzer auf ihrem Smartphone das Vorliegen von (Sicherheits-)Aktualisierungen, was zur Konsequenz hat, das Geräte aktiv genutzt werden, welche für Angriffe anfällig sind, trotz der Verfügbarkeit von Sicherheitsupdates. Durch eine automatische Benachrichtigung der Nutzer über das Vorliegen einer Aktualisierung, soll eine schnellere Verbreitung von Updates erreicht werden.

Umsetzung iPhone 7: Im Selbsttest konnte durch das Intercepten von SSL eine periodische Überprüfung von Aktualisierungen eruiert werden.

```
GET /assets/com_apple_MobileAsset_SoftwareUpdate/
    com_apple_MobileAsset_SoftwareUpdate.xml HTTP/1.1
Host: mesu.apple.com
If-Modified-Since: Tue, 03 Oct 2017 16:58:34 GMT
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: de-de
Connection: close
User-Agent: MobileAsset/1.1
```

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.06	Das Gerät muss nach dem Systemstart und im laufenden Betrieb auf Sicherheitsaktualisierungen überprüft werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Im Selbsttest konnte eine periodische Nachfrage nach (System-) Aktualisierungen unter Android festgestellt werden. Das Intercepten der Anfrage des Gerätes an den Updateserver konnte leider nicht vollzogen werden, da die Verbindung mittels Certificate Pinning abgesichert ist und tiefergehende Eingriffe in das System erforderlich gewesen wären.

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.06	**Das Gerät muss nach dem Systemstart und im laufenden Betrieb auf Sicherheitsaktualisierungen überprüft werden.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-MGMT.07

Anforderung: Manipulationen an installierten Anwendungen müssen erkannt werden.

Beschreibung: Um die Integrität einer Anwendung nach der Installation sicherzustellen, soll vor jedem Aufruf eine Überprüfung stattfinden.

Allgemeine Hintergrundinformationen: Manipulationen an Anwendungen, welche der Nutzer installiert hat und denen er „vertraut“, ermöglichen einem Angreifer die Möglichkeit unentdeckt zu bleiben und reduzieren den Aufwand, Schadcode über den AppStore einzuschleusen.
Um die Manipulation von Dateien und Anwendungen zu erkennen, eignet sich die Signatur von Anwendungen und deren Überprüfung während der Installation und zur Laufzeit (z.B. vor dem Starten einer Anwendung).

Umsetzung iPhone 7: iOS überprüft bei jedem Applikationsstart die Gültigkeit der Signatur. Bei Veröffentlichung einer Anwendung im *AppStore* wird diese durch Apple signiert. Ohne vorhandene oder gültige Signatur kann keine Anwendung ausgeführt werden. Eine Ausnahme stellen Anwendungen dar, welche durch ein Enterprise Zertifikat signiert wurden. Diese können, nachdem das Firmenprofil importiert wurde, auch ohne eine Signatur von Apple gestartet werden. Vor Ausstellung eines Enterprise Zertifikat für Unternehmen, erfolgt eine umfangreiche Überprüfung durch Apple.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.07	Manipulationen an installierten Anwendungen müssen erkannt werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android überprüft die Validität der Signatur einer Applikation vor der Installation. Vor dem Start einer Anwendung existiert kein derartiger Mechanismus. Damit kann eine nachträgliche Manipulation an Anwendungen nicht erkannt werden[IAc]. Im August 2017 stellte Google den Dienst *Play Protect* vor, welcher das Gerät kontinuierlich nach Malware durchsucht und verdächtige Aktivitäten von Anwendungen detektieren soll.[Incu] Eine Dokumentation zur Funktionsweise und Architektur ist nicht öffentlich verfügbar.

Schwachstelle: Eine Bewertung dieser Anforderung ist für dieses Gerät nicht möglich, da das zugrundeliegende Betriebssystem keine Integritätsüberprüfung vor der Ausführung von Anwendungen durchführt.

Bewertung:

MGMT.07	Manipulationen an installierten Anwendungen müssen erkannt werden.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung ist aktuell nicht umgesetzt.

R-MGMT.08

Anforderung: Das Betriebssystem soll eine Möglichkeit bieten, Geräte auf denen ein Root-Zugriff etabliert wurde, zu erkennen.

Beschreibung: Die Erkennung von mobilen Endgeräten in denen der Nutzer über privilegierte Rechte verfügt (z.B. Root), ermöglicht einer Anwendung die Funktionalitäten einzuschränken, da die Vertraulichkeit von auf dem Gerät gespeicherten sensiblen Informationen nicht mehr gewährleistet ist.

Allgemeine Hintergrundinformationen: Die Etablierung eines privilegierten Zugangs auf einem Smartphone ist für das Testen von Anwendungen zum Teil unabdingbar. Im Gegensatz zu dieser legitimen Anforderungen, kann ein Root Zugriff im Produktivbetrieb für einen Standardnutzer im privaten oder geschäftlichen Umfeld die Gesamtsicherheit des Gerätes beeinträchtigen.

Umsetzung iPhone 7: iOS bietet nativ keine Schnittstelle, welche den Gerätestatus überprüft und einen Jailbreak*Als „Jailbreaken“ wird der Prozess des Erlangen von Root-Rechten auf einem iOS-Gerät, durch Ausnutzen einer im Betriebssystem vorhandenen Schwachstelle, genannt.* erkennt. Im Betriebssystem ist keine Möglichkeit implementiert, welche das Freischalten des Root-Nutzers erlaubt. Somit ist eine Rechteausweitung nur durch Schwachstellen im Betriebssystem auf inoffiziellem Wege möglich.

Zur Detektion eines „gejailbreakten'"Gerät werden verschiedene Verfahren angegeben, welche durch jede Anwendung eigenständig implementiert werden muss:

- Überprüfen der Existenz von Dateien (z.B. /Applications/Cydia.app/)
- Schreiben einer Testdatei außerhalb des Sandbox-Bereiches (z.B. /private/jailbreak.txt)
- Erfolgreiches Aufrufen des Protocol Handlers von Cydia[19]
- Ausführen privilegierter Systembefehle
- Überprüfen, ob SSH Dienst vorhanden ist

[Mo17]

Schwachstelle: Durch die fehlende Unterstützung des Betriebssystems, kann die Erkennung eines *„Jailbreaks“* nur auf Anwendungsebene durchgeführt werden. Dazu müssen auf Entwicklungsebene entsprechende Mechanismen implementiert werden. Die Umgehung dieser Mechanismen, um Root-Rechte zu verdecken, können in einem Penetrationstest aufgedeckt werden und unterliegen einer individuellen Bewertung.

[19] Cydia ist ein inoffizieller AppStore, welcher häufig nach Jailbreaks installiert wird, um Anwendungen installieren zu können, welche nicht im offiziellen AppStore verfügbar sind.

Bewertung: Die Bewertung der Sicherheit verwendeter Mechanismen zur Erkennung eines „*Jailbreaks*“ werden im Rahmen eines Penetrationstests durchgeführt.

MGMT.08	**Das Betriebssystem soll eine Möglichkeit bieten, Geräte auf denen ein Root-Zugriff etabliert wurde, zu erkennen.**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Diese Anforderung ist aktuell nicht umgesetzt.

Umsetzung Samsung Galaxy S7: Android bietet mit dem Dienst *SafetyNET* seit Anfang des Jahres Entwicklern die Möglichkeit über eine einheitliche Schnittstelle den Gerätestatus abzurufen. Damit lässt sich aus Entwicklersicht mit einem geringen Aufwand ein Root-Zugriff erkennen.

Schwachstelle: In den vergangenen Wochen erschienen Anwendungen (Magisk , welche die Überprüfungen von *SafetyNET* temporär umgehen konnten um als ein sicheres Gerät kategorisiert zu werden. Durch zusätzliche Überprüfungen von *SafetyNET* wurden die Umgehungsstrategien erkannt und der Gerätestatus als unsicher eingestuft. Zum Zeitpunkt der Recherche (30.09.2017) konnte keine Anwendung recherchiert werden, welche die Überprüfungen von *SafetyNET* umgehen konnte.

Bewertung:

MGMT.08	**Das Betriebssystem soll eine Möglichkeit bieten, Geräte auf denen ein Root-Zugriff etabliert wurde, zu erkennen.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-MGMT.09

Anforderung: Es muss eine vertrauenswürdige Instanz geben, welche nachträglich schadhafte Anwendungen deinstallieren kann.

Beschreibung: Das Betriebssystem soll Mechanismen besitzen, welche die nachträgliche Deinstallation von als bösartig enttarnten Anwendungen auf einem Gerät ermöglichen.

Allgemeine Hintergrundinformationen: In der Vergangenheit wurden unter Android und iOS Anwendungen, teilweise mit mehreren zehntausenden Installationen, gefunden, die Schadcode enthielten und bei den Reviews der AppStores keine Auffälligkeiten erzeugten. Um die ausgehende Gefahr von diesen installierten Anwendungen zeitnah zu eliminieren, soll durch das Betriebssystem eine Schnittstelle existieren, welche eine globale Deinstallation bösartiger Anwendungen durchführen kann.

Umsetzung iPhone 7: Die von Apple veröffentlichten Sicherheitsdokumentation ([Incm]) enthält keine Information, über die Implementierung dieser Anforderung. 2008 wurde ein Artikel eines IT-Forensikers veröffentlicht, der bei einer forensischen Untersuchung eine URL[20] fand, mittels der ein Anwendungsstart verhindert werden konnte.[Kim]

Schwachstelle: Ein Angriff gegen diese Anforderung ist aktuell nicht möglich, da keine Informationen vorliegen, ob ein derartiger Mechanismus in iOS existiert.

Bewertung: Eine Bewertung dieser Anforderung ist aufgrund fehlender Dokumentation nicht möglich.

Umsetzung Samsung Galaxy S7: Der AppStore *Google Play Store* bietet Nutzern die Möglichkeit Anwendungen über eine Weboberfläche unabhängig von ihrem Gerät zu installieren und deinstallieren. Damit ist es grundsätzlich möglich, Anwendungen aus der Ferne zu löschen[Inct]. Inwiefern Google global von dieser Möglichkeit Gebrauch machte, konnte nicht recherchiert werden.

Schwachstelle: Ein Deaktivieren dieser Möglichkeit ist durch Installation eines modifizierten Betriebssystems (z.B. LineageOS), welches standardmäßig nicht die Google Play Services enthält, möglich.
Es konnte aktuell keine Schwachstelle entdeckt werden, welche eine unautorisierte globale Deinstallation von Anwendungen zuließ.

Bewertung:

MGMT.09	Es muss eine vertrauenswürdige Instanz geben, welche nachträglich schadhafte Anwendungen deinstallieren kann.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

[20] https://iphone-services.apple.com/clbl/unauthorizedApps

Mobile Device Management

R-MGMT.10

Anforderung: Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.

Beschreibung: Die Integration mobiler Endgeräte in Firmennetzwerke wird künftig eine immer höhere Priorität bemessen. Einer Umfrage des IFAK zufolge nutzten 2016 1,76 Millionen Arbeitnehmern in Deutschland ein Firmen-Smartphone.[Med]

Allgemeine Hintergrundinformationen: Die zunehmende Nutzung mobiler Geräte im geschäftlichen Umfeld in den vielfältigen Bereichen, erfordern ein zentrales Management der Geräte. Neben proprietären Lösungen von Drittanbietern soll diese Anforderung klären, inwiefern nativ bereits Schnittstellen zur Massenkonfiguration von Smartphones zur Verfügung stehen.

Umsetzung iPhone 7: iOS unterstützt Konfigurationsprofile, die diverse Einstellungen festlegen:

- Coderichtlinien
- Einschränkung von Gerätefunktionen (z.B. Deaktivieren der Kamera)
- WLAN-Einstellungen
- VPN-Einstellungen
- Mail- / Exchange-Konfiguration
- LDAP Verzeichnis
- CalDav-Einstellungen
- Anmeldedaten und Schlüssel
- Erweiterte Mobilfunkeinstellungen

Zum Schutz vor Manipulationen werden die Profile signiert und verschlüsselt. Die Übertragung der Profile kann direkt über den *Apple Configurator* oder über einen Linkaufruf erfolgen.
[Incm]

Schwachstelle: In der Recherche wurde ein von der Firma Checkpoint entdeckter und umstrittener Angriff identifiziert, welcher Geräte die bereits über ein Konfigurationsprofil verwaltet werden, durch Aufruf eines Linkes einem neuen Server zugeordnet werden. Damit ist eine komplette Kontrolle des Gerätes und der Kommunikation möglich[21].[Schb]

[21] https://www.heise.de/security/meldung/Sidestepper-Sicherheitsrisiken-von-Mobile-Device-Management-mit-iOS-3159648.html

Bewertung:

MGMT.10	Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff durch Link-Aufruf
Schwachstelle	
Entdeckung:	3 - Kompliziert
Ausnutzung:	5 - Leicht
Bekanntheit:	6 - Offensichtlich
Transparenz:	6 - keine Information
Erkennung:	3 - Logging mit Mitteilung
Angriffsvektor:	4 - Lokal
Umfang:	7 - Übergreifend
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	4.63 - Mittel
Gesamt Firma 1:	5.19 - Mittel
Gesamt Firma 2:	6.19 - Hoch

Umsetzung Samsung Galaxy S7: Android bietet sei Version 5.0 (Marshmallow) eine Schnittstelle an, um Geräte mit einem *managed profile* zu verwalten. Hinsichtlich der Architektur werden verwaltete Anwendungen und Daten von den restlichen Anwendungen getrennt. Eine direkte Nutzung der Schnittstelle ist über die von Google betriebene kommerzielle Anwendung *G Suite* oder durch Drittanbieter möglich. *G Suite* erlaubt die Konfiguration folgender Einstellungen:

- Zugriffsschutz
- Entfernen der Arbeitsanwendungen und -daten
- Beenden der Synchronisation bei als infiziert kategorisierten Geräte
- Einschränken der Geräteeinstellungen (z.B. WiFi, USB Zugriff, Screenshot, ..)
- Einschränkung des AppStores (Whitelisting von Anwendungen)
- Automatische App-(De)Installation
- Monitoring der Geräte

[Incy]

Schwachstelle: 2016 wurde von Forschern der Firma CheckPoint im *PlayStore* die App *„Gooligan"* entdeckt, welche nach Installation ein Rootkit nachlud und eine Korrumpierung des Gerätes über verschiedene Schwachstellen versuchte. Bei erfolgreicher Infektion wurden konfigurierte E-Mailkonten und Authentifizierungstoken unter anderem von *G Suite* entwendet. Dem Artikel zufolge, wurde diese Anwendung über 1 Million mal installiert und es waren mehrere hundert Nutzer der Enterprise Anwendung *G Suite* betroffen. Die Anwendung wurde umgehend von Google entfernt und betroffene Nutzer informiert.[Tea]

Bewertung:

MGMT.10	Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-MGMT.11

Anforderung: Das Gerät muss die Integration von Sicherheitsrichtlinien ermöglichen.

Beschreibung: Die vorhandene Schnittstelle zur Konfiguration von Sicherheitsrichtlinien wurde für beide Geräte / Betriebssysteme bereits in der Anforderung R-MGMT.10 analysiert und erfordert keine weitere Recherche.
Sollte sich künftig an der Implementierung von Sicherheitseinstellungen grundsätzliches ändern, kann in einer späteren Auswertung eine eigene Bewertung dieser Anforderung durchgeführt werden.

R-MGMT.12

Anforderung: Es muss eine individuelle Möglichkeit geben, nur ausgewählte Anwendungen auf einem Gerät installieren zu können.

Beschreibung: Um insbesondere Geräte im geschäftlichen Umfeld ein Mehr an Sicherheit bieten zu können, soll die Installation auf ausgewählte Anwendungen beschränkt werden.

Allgemeine Hintergrundinformationen: Die AppStores von Android und iOS bieten mittlerweile Millionen von Anwendungen an. Analysen der Vergangenheit und statistische Wahrscheinlichkeiten lassen annehmen, dass unter diesen zahlreichen Anwendungen bösartige enthalten sind. Um die Möglichkeit der Installation von Malware über einen präparierten Link beispielsweise zu verringern, ist die Beschränkung auf ausgewählte Anwendungen (Whitelisten) von Vorteil.

Umsetzung iPhone 7: Administratoren können bei verwalteten Geräten festlegen, ob und welche Anwendungen aus dem *AppStore* installiert werden können.[Incm]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.12	Es muss eine individuelle Möglichkeit geben, nur ausgewählte Anwendungen auf einem Gerät installieren zu können.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet für Entwickler von *Mobile Device Management* Lösungen, eine Schnittstelle für das Whitelisten von Anwendungen des *Play Stores*. Die kommerzielle Google Anwendung *G Suite* ermöglicht über eine Weboberfläche die Installation von Anwendungen auf Zugelassene zu beschränken.[Incw]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.12	Es muss eine individuelle Möglichkeit geben, nur ausgewählte Anwendungen auf einem Gerät installieren zu können.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-MGMT.13

Anforderung: Mobile Device Management Lösungen müssen einen Zugriff aus der Ferne erhalten können.

Beschreibung: Aus Support- und zu Verwaltungszwecken sollen *Mobile Device Management* Lösungen einen Zugriff auf das Gerät aus der Ferne erhalten. Damit könnten beispielsweise globale Probleme zeitnah ohne Interaktion des Nutzers behoben werden.

Allgemeine Hintergrundinformationen: Bei Auftreten von Sicherheitsproblemen oder zum Support von Anwendern soll durch das Betriebssystem eine Möglichkeit existieren, Geräte aus der Ferne zu warten. Insbesondere bei Sicherheitsproblemen ist eine zeitnahe Aktualisierung unabdingbar.

Umsetzung iPhone 7: Bei Aktivierung der Optionen kann jeder Nutzer, der sein Gerät mit einer Apple-ID verknüpft hat, aus der Ferne folgende Aktionen durchführen:

- Fernlöschen, bei Verlust des Gerätes
- Aktivierungssperre, bei Diebstahl des Gerätes zum Schutz vor unberechtigter Nutzung
- Standortabfrage
- Lautes Klingeln des Gerätes

Von einem Administrator verwaltete iOS Geräte unterstützen unter anderem zusätzlich folgende Möglichkeiten[22]:

- (De-)Installation von Anwendungen
- Bildschirmbeobachtung mit *Classroom*
- Einschränkung der via *AirPrint* zulässigen Drucker
- Verändern der APN Einstellungen
- Verwendung eines DNS Proxies
- ..

[Inch]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

[22] Eine vollständige Liste ist unter diesem Link verfügbar: https://developer.apple.com/library/content/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html

Bewertung:

MGMT.13	Mobile Device Management Lösungen müssen einen Zugriff aus der Ferne erhalten können.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet Nutzern über die *PlayStore* Dienste allen registrierten Geräten folgende Remotezugriffe:

- Lokalisation des Gerätes
- Laut stellen des Telefons
- Löschen der Benutzerdaten

Allen Enterprise Nutzern, welche die Lösung *G Suite* installiert und konfiguriert haben, werden zusätzliche Aktionen ermöglicht:

- Zurücksetzen des Geräte-PINs
- Gerät klingeln lassen
- Gerät sperren
- Gerät auf Werkseinstellungen zurücksetzen
- Lokalisation des Gerätes

[Incx]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.13	**Mobile Device Management Lösungen müssen einen Zugriff aus der Ferne erhalten können.**
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

R-MGMT.14

Anforderung: Es muss eine zentrale Möglichkeit geben, Produkt- und Sicherheitsaktualisierungen auszurollen.

Beschreibung: Von einem Administrator verwaltete Geräte sollen zentral mit Produkt- und Sicherheitsaktualisierungen ausgerollt werden können. Damit ist gewährleistet, dass alle (Unternehmens-)Geräte über den selben Schutz an Sicherheit verfügen.

Allgemeine Hintergrundinformationen: Um die Sicherheit einer Vielzahl an Geräten zu homogenisieren, empfiehlt sich die Erstellung von Sicherheitsrichtlinien. Damit werden alle Geräte vor einer produktiven Verwendung bespielt und erhalten das selbe Schutzniveau. Nachdem das Feld der IT Security ein sehr dynamisches ist, müssen Richtlinien permanent überprüft und angepasst werden, sowie Betriebssystem- als auch Anwendungsaktualisierungen eingespielt werden. Dieses soll ebenfalls durch eine zentrale Lösung ermöglicht werden.

Umsetzung iPhone 7: iOS erfüllt durch die Konfigurationsprofile bereits die Anforderung, das gleiche Schutzniveau auf Geräten zu implementieren. Die Aktualisierung dieser Profile kann ebenfalls aus der Ferne durchgeführt werden. Um die Netzwerkauslastung bei der Verteilung von Aktualisierungen für das Betriebssystem oder Anwendungen effizient zu gestalten, bietet der MacOS Server einen Caching Mechanismus, der die Internetbandbreite schont, da ein Update nur einmalig über das Internet abgerufen werden muss und die Verteilung im LAN erfolgt.[Incl]

Schwachstelle: Trotz intensiver Recherche konnte aktuell (September 2017) keine öffentliche Schwachstelle zur Umgehung dieses Verfahrens eruiert werden.

Bewertung:

MGMT.14	Es muss eine zentrale Möglichkeit geben, Produkt- und Sicherheitsaktualisierungen auszurollen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Recherche ergab derzeit keine Möglichkeit.

Umsetzung Samsung Galaxy S7: Android bietet für Entwickler Schnittstellen zum Konfigurieren von Sicherheitsrichtlinien, welche durch die kommerzielle Lösung *G Suite* verwendet wird, um auf einer Vielzahl an Geräten identische Richtlinien zu etablieren[Incw]. Es konnte gegenwärtig kein Mechanismus recherchiert werden, der eine zentrale Bereitstellung von Produkt- und Anwendungsupdates realisiert. *G Suite* ermöglicht Administratoren, die Erfassung der verwendeten App- und Betriebssystemversionen.

Bewertung: Nachdem unter Android derzeit nur eine teilweise Erfüllung dieser Anforderung existiert, kann keine Bewertung vorgenommen werden.

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

Beschreibung: Die Anbindung eines Smartphones an eine Managementlösung erhöht die Anforderungen an die eigene Sicherheit des Produktes enorm, da bei fehlenden Mechanismen die Sicherheit einer Vielzahl an Geräten in Gefahr ist.

Allgemeine Hintergrundinformationen: Diese Anforderung untersucht die vom Hersteller angebotene Managementlösung hinsichtlich der Umsetzung von Industriestandards, die einen Zugriff von Unbefugten verhindern.

Umsetzung iPhone 7: Apple schützt die Apple-ID mit einer Zwei-Faktor-Authentifizierung vor unbefugten Nutzern. Der Versand der Einmalpasswörter kann via SMS, eines automatischen Telefonanrufs oder über ein proprietäres Protokoll auf ausgewählte iOS-Geräte übermittelt werden.[Incm; Incp]
iOS verwendet mutmaßlich Monitoringsysteme, welche Anomalien bei Aktivitäten von Nutzern entdecken, diese sind jedoch nicht öffentlich dokumentiert und können daher keiner Bewertung unterzogen werden.

Schwachstelle: Bei Selektion einer SMS als Zustelloption für das Einmalkennwort, kann über Schwachstellen im Signalisierungssystem von Mobilfunk (SS7) die SMS mitgelesen werden und eine Übernahme des Accounts erfolgen. Dies setzt jedoch zusätzlich die Kenntnis des Apple-ID Passwortes voraus.

Angriffsszenario: Das Abfangen von SMS ist mittlerweile mit nur einem geringen Aufwand möglich. Das weltweite Signalisierungssystem SS7 weist unter anderem die Schwäche auf, dass keine Überprüfung stattfindet, aus welcher Zelle sich ein Mobilfunkgerät eingewählt hat. Damit ist es einem Angreifer möglich, sich aus fast allen Ländern dieser Welt als Einwähler mit einer Mobilfunknummer XY auszugeben. Nachdem SMS Nachrichten auch unter dem neusten Mobilfunkstandard LTE unverschlüsselt verschickt werden, ist ein Mitlesen von SMS Nachrichten möglich.

Bewertung:

MGMT.15	Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff auf Apple-ID Passwort und Mitlesen der SMS über SS7-Schwachstelle(n).
Schwachstelle	
Entdeckung:	1 - Praktisch unmöglich
Ausnutzung:	1 - Theoretisch
Bekanntheit:	6 - Offensichtlich
Transparenz:	3 - Dokumentation mit Hintergrundinformationen
Erkennung:	1 - Aktive Erkennung
Angriffsvektor:	4 - Lokal
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	2.75 - Niedrig
Gesamt Firma 1:	4.25 - Mittel
Gesamt Firma 2:	5.25 - Mittel

Umsetzung Samsung Galaxy S7: Alle Google Dienste können mit einer Zwei-Faktor-Authentifizierung zusätzlich geschützt werden. Die Übermittlung des Einmalpassworts kann via SMS, eines automatischen Telefonanrufs erfolgen oder eine Generierung der Passwörter erfolgt durch Nutzung des *Google Authenticator's*.
[Incr]

Schwachstelle: Bei Selektion einer SMS als Zustelloption für das Einmalkennwort, kann über Schwachstellen im Signalisierungssystem von Mobilfunk (SS7) die SMS mitgelesen werden und eine Übernahme des Accounts erfolgen. Dies setzt jedoch zusätzlich die Kenntnis des Google Account Passwortes voraus.
Die Recherche nach Schwachstellen in der Anwendung *Google Authenticator* ergab keine Befunde, die letzte Schwachstelle mit einem CVE-Bewertung von 1.9 (niedrig) war 2012.[Corc]

Angriffsszenario: Das Abfangen von SMS ist mittlerweile mit nur einem geringen Aufwand möglich. Das weltweite Signalisierungssystem SS7 weist unter anderem die Schwäche auf, dass keine Überprüfung stattfindet, aus welcher Zelle sich ein Mobilfunkgerät eingewählt hat. Damit ist es einem Angreifer möglich, sich aus fast allen Ländern dieser Welt als Einwähler mit einer Mobilfunknummer XY auszugeben. Nachdem SMS Nachrichten auch unter dem neusten Mobilfunkstandard LTE unverschlüsselt verschickt werden, ist ein Mitlesen von SMS Nachrichten möglich.

Bewertung:

MGMT.15	Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Social Engineering Angriff auf Passwort des Google-Accounts und Mitlesen der SMS über SS7-Schwachstelle(n)
Schwachstelle	
Entdeckung:	1 - Praktisch unmöglich
Ausnutzung:	1 - Theoretisch
Bekanntheit:	6 - Offensichtlich
Transparenz:	3 - Dokumentation mit Hintergrundinformationen
Erkennung:	1 - Aktive Erkennung
Angriffsvektor:	4 - Lokal
Umfang:	3 - Lokal
Nutzerinteraktion:	3 - Erforderlich
Eintrittswahrscheinlichkeit	
Angreifer Firma 1:	5.75 - Mittel
Angreifer Firma 2:	7.75 - Hoch
Schwachstelle:	2.75 - Niedrig
Gesamt Firma 1:	4.25 - Mittel
Gesamt Firma 2:	5.25 - Mittel

Transparenz

R-MGMT.16

Anforderung: Der verwendete Code innerhalb eines Gerätes muss offen liegen, um eine transparente Validierung der Sicherheit zu ermöglichen.

Beschreibung: Der Source Code von Betriebssystem, Kernel und Treiber soll offen liegen, um für jedermann die Möglichkeit zu schaffen, konkrete (Sicherheits-)Implementierungen zu überprüfen.

Allgemeine Hintergrundinformationen: Unternehmen oder Behörden, welche Daten verarbeiten, die der höchsten Geheimhaltungsklasse unterliegen, müssen die Möglichkeit erhalten, den verwendenden Code auf konkrete Sicherheitsmechanismen oder Backdoors zu überprüfen. Durch ein Offenlegen oder zur Verfügung stellen des verwendeten Sourcecodes kann dies realisiert werden. Diese Anforderung überprüft den gegenwärtigen Stand der Transparenz von Hersteller und Anbieter mobiler Betriebssysteme.

Umsetzung iPhone 7: Das Betriebssystem iOS ist proprietär und der verwendete Source-Code wird der Öffentlichkeit nicht zur Verfügung gestellt. Damit liegt die Sicherheit dieses Produkts im Vertrauen der erhaltenen Zertifizierungen.
Die vorliegende iOS Version hat folgende Zertifikate erhalten:

- ISO 270001
- Cryptographic Validation FIPS 140-2[23]
- Common Criteria Certification ISO 15408
- Commercial Solutions for Classified (CSfC)

[Incm]

Schwachstelle: Allgemein erhöht sich der Aufwand für Angreifer, eine Schwachstelle in der Implementierung aufzudecken, was jedoch die Existenz von Angriffsvektoren nicht ausschließt.

Bewertung: Eine Bewertung dieser Anforderung ist nicht möglich, da es eine politische Entscheidung des Herstellers ist, seinen Source-Code nicht offenzulegen. Es kann jedoch unter Umständen Anwendungsgruppen geben, die eine Offenlegung des Quelltextes als zwingende Voraussetzung für einen potentiellen Einsatz dieser Produktreihe erwarten.

[23] Die Verschlüsselungsmodule wurden mit der Konformität des US Federal Information Process Standard 140-2 überprüft und validiert.[Incm]

Umsetzung Samsung Galaxy S7: Das Betriebssystem Android wird unter der *Apache Software License, Version 2.0* quelloffen und für jedermann nutzbar zur Verfügung gestellt. Damit kann die Implementierung aller Sicherheitsmechanismen auf Betriebssystemebene transparent überprüft werden. Der verwendete Linux Kernel wird unter der *GNU General Public License* bereitgestellt.[Prok]
Die Kernerlergänzungen der Hardwarehersteller werden nicht von allen Herstellern offen gelegt. Für das vorliegende Modell kann der Source-Code des Kernels abgerufen werden[24]

Bewertung: Eine allgemeine Bewertung dieser Anforderung ist nicht möglich, da konkrete Schwachstellen im Source-Code bereits durch die Anforderungen im Abschnitt Hardware oder Betriebssystem umfasst werden.
Es kann jedoch unter Umständen Anwendungsgruppen geben, die eine Offenlegung des Quelltextes als zwingende Voraussetzung für einen potentiellen Einsatz dieser Produktreihe betrachten.

[24] https://opensource.samsung.com/reception/receptionSub.do?method=sub&sub=F&searchValue=sm-g930

R-MGMT.17

Anforderung: Hersteller und Anbieter von Betriebssystemen sollen Transparenz bei Sicherheitsproblemen zeigen.

Beschreibung: Ein offener Umgang beim Auftauchen von Sicherheitsproblemen erhöht die Glaubwürdigkeit des Anbieters, das Thema Sicherheit ernst zu nehmen und die Daten der Nutzer schützen zu wollen.

Allgemeine Hintergrundinformationen: Das Bereitstellen von Informationen (und Aktualisierungen) über vorhandene Schwachstellen erhöht die Authentizität, dass die Sicherheit des Produktes für den Anbieter von Bedeutung ist. Die Analyse dieser Anforderung zeigt die dazu getroffenen Maßnahmen der Anbieter auf.

Umsetzung iPhone 7:

iOS veröffentlicht auf seiner Produkthomepage bei jeder Aktualisierung die behobenen Schwachstellen unter Nennung (falls vorhanden) der CVE-Nummer. Damit kann eindeutig nachvollzogen werden, welche Schwachstelle behoben wurde[Incf]. Die verwendeten Verfahren und Mechanismen zur Erhöhung der Sicherheit des Gerätes werden im Sicherheitsguide regelmäßig aktualisiert. Die Informationen im Guide stellen auch die Grundlage der Informationen für die Auswertung von iOS-Geräten dar[Incm].

Bewertung: Eine Bewertung dieser Anforderung ist nicht durchführbar. Diese Anforderung liefert einen Überblick, wie viel Transparenz der Hersteller bei Sicherheitsproblemen zeigt.

Umsetzung Samsung Galaxy S7: Android veröffentlicht jeden Monat ein Sicherheitsupdate, welche (neu) aufgetretene Schwachstellen behebt. Die Hardwarehersteller müssen vor Freigabe dieser Aktualisierungen teilweise Modifikationen an Treibern oder Kernel vornehmen, um die Schwachstellen zu beheben. Die Dokumentation der vorhandenen Sicherheitsmechanismen auf den verschiedensten Ebenen eines Android-Gerätes werden umfangreich zur Verfügung gestellt[25]. Diese Informationen richten sich mit Empfehlungen an Hersteller von Smartphones auf Android-Basis.[Proo]

Bewertung: Eine Bewertung dieser Anforderung ist nicht durchführbar. Diese Anforderung liefert einen Überblick, wie viel Transparenz der Hersteller bei Sicherheitsproblemen zeigt.

[25] https://source.android.com/security/

R-MGMT.18

Anforderung: Gerätehersteller und Betriebssystemanbieter müssen (Sicherheits-) Aktualisierungen in einem definierten Zeitraum anbieten.

Beschreibung: Das Bereitstellen von (Sicherheits-)Aktualisierungen durch Hersteller von Smartphones und deren verwendetes Betriebssystem in einem vorher bekannt gegebenen Zeitraum, soll bei Verkaufsstart bekannt und garantiert werden. Damit hat ein Nutzer oder Unternehmen die Möglichkeit, einen Zeitraum für die Nutzung des Gerätes festzulegen.

Allgemeine Hintergrundinformationen:

Die Bekanntgabe der herstellerseitigen Unterstützung des Gerätes mit Softwareaktualisierungen erhöht die Transparenz hinsichtlich des Lebenszyklus eines Gerätes und schafft Vertrauen zu den Nutzern hinsichtlich der Verwendungsdauer des Gerätes.

Umsetzung iPhone 7: Apple als Hardwarehersteller und Anbieter des Betriebssystem iOS liefert keine Garantien für eine Versorgung mit Aktualisierungen über einen garantierten Zeitraum. In der Vergangenheit wurden iOS Geräte über einen langen Zeitraum mit Aktualisierungen versorgt. Das 2013 erschienene iPhone 5 erhielt auch ein Update auf die aktuellste Betriebssystemversion 11.0.[Incf]

Bewertung:

MGMT.18	Gerätehersteller und Betriebssystemanbieter müssen (Sicherheits-) Aktualisierungen in einem definierten Zeitraum anbieten.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Diese Anforderung ist aktuell nicht umgesetzt.

Umsetzung Samsung Galaxy S7: Samsung verspricht für Enterprise Kunden beim vorliegenden Gerät eine Verfügbarkeit der Geräte von 2 Jahren und monatlich garantierte Sicherheitsupdates für einen Zeitraum von drei Jahren ab Erscheinen des Gerätes an.[Comj]

Bewertung:

MGMT.18	Gerätehersteller und Betriebssystemanbieter müssen (Sicherheits-) Aktualisierungen in einem definierten Zeitraum anbieten.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung wird umgesetzt.

R-MGMT.19

Anforderung: Das Gerät, bestehend aus Hardware- und Softwarekomponenten, muss durch legitimierte Stellen zertifiziert werden.

Beschreibung: Diese Anforderung stellt eine Übersicht der vorhandenen Zertifizierungen eines Gerätes auf. Durch Zertifizierungen kann das Einhalten von Sicherheitsstandard verifiziert werden. Die Anbieter oder Organisationen von Zertifizierungen sollen etabliert und vertrauenswürdig sein.

Allgemeine Hintergrundinformationen: Eine Zertifizierung von Produkten ermöglicht die einheitliche Überprüfung von Anforderungen zur Erlangung des jeweiligen Zertifikates. Durch Sicherheitszertifizierungen kann ein Mindestmaß an Sicherheit gewährleistet werden und sorgt für zusätzliches Vertrauen bei den Nutzern.

Umsetzung iPhone 7: iOS validiert die verwendeten kryptographischen Module auf Konformität mit dem *Cryptographic Module Validation Program* des *National Institute of Standards and Technology* der Vereinigten Staaten von Amerika. Das Krypto-Modul wurde in Level 2 des *FIPS 140* Standards validiert. Die ISO 27001 Zertifizierung für ein vorhandenes *Information Security Management System* wurde für die Produkte *Apple School Manager*, *iCloud*, *iMessage*, *FaceTime*, *verwaltete Apple-IDs* und *iTunes* erteilt. Außerdem wurde eine Zertifizierung nach den *Common Criteria's* durchgeführt.[Incm]

Schwachstelle: Schwachstellen im klassischen Sinne können zu dieser Anwendung nicht existieren. Es erfolgt eine Bewertung hinsichtlich der Existenz von Zertifizierung des Betriebssystem und Hardwaremodulen.

Bewertung:

MGMT.19	Das Gerät, bestehend aus Hardware- und Softwarekomponenten, muss durch legitimierte Stellen zertifiziert werden.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Diese Anforderung wird umgesetzt.

Umsetzung Samsung Galaxy S7: Die von Samsung verwendeten kryptographischen Module sind ebenfalls nach *FIPS 140-2* zertifiziert worden. Ebenfalls wurde von der *Common Criteria* die Einhaltung des *MDFPP-Standard (Mobile Device Fundamentals Protection Profile)* überprüft und bescheinigt.
Eine Übersicht aller nationalen Zertifizierungen kann auf der Samsung Seite abgerufen werden[26].[Comk]

Schwachstelle: Schwachstellen im klassischen Sinne können zu dieser Anwendung nicht existieren. Es erfolgt eine Bewertung hinsichtlich der Existenz von Zertifizierung des Betriebssystem und Hardwaremodulen.

Bewertung:

MGMT.19	Das Gerät, bestehend aus Hardware- und Softwarekomponenten, muss durch legitimierte Stellen zertifiziert werden.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung wird umgesetzt.

[26] https://www.samsungknox.com/de/knox-technology/security-certifications

R-MGMT.20

Anforderung: Es muss für Behörden auf nationaler Ebene die Möglichkeit geben, Spezifikationen am Gerät und Betriebssystem individuell vorzunehmen.

Beschreibung: Behörden auf nationaler Ebene sollen die Möglichkeit erhalten, Modifikationen gemäß ihren (Sicherheits-)Anforderungen vornehmen zu können. Behörden, insbesondere mit Sicherheitsaufgaben, besitzen erhöhte Anforderungen an die Sicherheit der auf einem Geräte gespeicherten und verarbeiteten Geräte, welche durch Industriestandards nicht gelöst werden können. Um eine Nutzung mobiler Endgeräte im 21. Jahrhundert auf Behördenebene zu ermöglichen, sollen durch diese Systemmodifikationen möglich sein.

Allgemeine Hintergrundinformationen: Die Sicherheitsanforderungen für beispielsweise Geheimdienste besitzen einen wesentlich höheren Schutzbedarf im Vergleich zu Privatnutzern oder gewöhnlichen Unternehmen. Die Vertraulichkeit der sensitiven Daten kann von höchstem Interesse sein, um die Identität von im geheimen operierenden Personen nicht zu gefährden. Der Schutz dieser Daten kann teilweise durch Standardlösungen nicht gewährleistet werden und erfordert Modifikationen. Daneben unterstehen Hersteller von Smartphones und mobilen Betriebssystemen der nationalen Gesetzgebung des Firmensitzes und es könnte (politischer) Einfluss genommen werden.

Umsetzung iPhone 7: Es konnten keine Informationen über individuelle Modifikationen für iOS Geräte recherchiert werden. Damit besteht keine Möglichkeit Veränderungen auf Systemebene vorzunehmen.

Bewertung:

MGMT.20	Es muss für Behörden auf nationaler Ebene die Möglichkeit geben, Spezifikationen am Gerät und Betriebssystem individuell vorzunehmen.
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Diese Anforderung ist aktuell nicht umgesetzt.

Umsetzung Samsung Galaxy S7: Samsung als Gerätehersteller hat den im vorliegenden Gerät genutzten Quelltext des Kernels veröffentlicht. Das Betriebssystem Android steht als Open Source Lösung quelloffen für jedermann zur Verfügung.[Prok; Comm]

Bewertung:

MGMT.20	Es muss für Behörden auf nationaler Ebene die Möglichkeit geben, Spezifikationen am Gerät und Betriebssystem individuell vorzunehmen.
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Diese Anforderung wird umgesetzt.

C. Benutzergruppe 1 Geschäftsführer

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für die Smartphones iPhone 7 und Samsung Galaxy S7 aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.2 aufgelistet.

C.1 iPhone 7 (iOS 10.3.3)

Dieser Abschnitt bewertet die Auswirkungen der Schwachstellen für das iPhone 7.

C.1.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

HW.02	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	3.33 - Mittel

C.1.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

OS.01	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Effizienter Brute-Force Angriff gegen Zugriffssperre durch Schwachstelle in Lightning-Anschluss.
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	5 - Kleinere Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	6.60 - Hoch
Finanzielle:	1.75 - Niedrig
Gesamt:	4.44 - Mittel

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

OS.02	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.50 - Niedrig
Gesamt:	1.44 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

OS.06	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Schwachstelle im Audio Framework beim Abspielen von OPUS Dateien.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.40 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.67 - Niedrig

C.1.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

Eine Bewertung der automatischen und manuellen Reviews der App Stores ist bei fehlender öffentlich verfügbarer Dokumentation nicht möglich.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

MGMT.03	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.56 - Niedrig

R-MGMT.05

Anforderung: Das Gerät muss nativ eine sichere Mailkommunikation anbieten.

MGMT.05	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Denial of Service durch manipulierten Bilddateianhang (CVE-2017-7097)
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	1.40 - Niedrig
Finanzielle:	0.25 - Niedrig
Gesamt:	0.89 - Niedrig

R-MGMT.10

Anforderung: Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.

MGMT.10	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff durch Link-Aufruf
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	3 - Kleine Auswirkungen auf jährlichen Umsatz
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.80 - Mittel
Finanzielle:	2.25 - Niedrig
Gesamt:	3.67 - Mittel

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

MGMT.15	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff auf Apple-ID Passwort und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	2.22 - Niedrig

C.2 Samsung Galaxy S7 (Android 7.0 - Sicherheitsupdates 01.08.2017)

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für das Smartphone *Samsung Galaxy S7* aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.2 aufgelistet.

C.2.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Bewertung:

HW.02	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065 (Broadcom), CVE-2017-0781 (BlueBorne)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	3.33 - Mittel

R-HW.03.2

Anforderung: Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.

Bewertung:

HW.03.2	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle in ARM Architektur
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	3 - Kleine Auswirkungen auf jährlichen Umsatz
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	5.80 - Mittel
Finanzielle:	2.25 - Niedrig
Gesamt:	4.22 - Mittel

C.2.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

Bewertung:

OS.01	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Zugriff auf unverschlüsselte Daten durch Dumpen des Flashspeichers.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	3 - Kleine schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	3 - Mehrere Betroffene
Gesamtauswirkungen	
Technische:	3.80 - Mittel
Finanzielle:	1.75 - Niedrig
Gesamt:	2.89 - Niedrig

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

Bewertung:

OS.02	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Speicherzugriffsverletzung CVE-2017-5121, CVE-2017-5122.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.50 - Niedrig
Gesamt:	1.44 - Niedrig

R-OS.03

Anforderung: Die Updates des Betriebssystems müssen authentisch sein.

Bewertung:

OS.03	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Manuelles Downgrade auf frühere Version.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	3.60 - Mittel
Finanzielle:	0.50 - Niedrig
Gesamt:	2.22 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

Bewertung:

OS.06	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.50 - Niedrig
Gesamt:	3.00 - Mittel

R-OS.08

Anforderung: Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.

Bewertung:

OS.08	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle im Media Framework.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.50 - Niedrig
Gesamt:	3.00 - Mittel

C.2.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

Bewertung:

MGMT.03	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.50 - Niedrig
Gesamt:	3.00 - Mittel

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

Bewertung:

MGMT.15	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Social Engineering Angriff auf Passwort des Google-Accounts und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	0 -
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	0.50 - Niedrig
Gesamt:	2.11 - Niedrig

D. Benutzergruppe 2 Mitarbeiter

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für die Smartphones iPhone 7 und Samsung Galaxy S7 aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.3 aufgelistet.

D.1 iPhone 7 (iOS 10.3.3)

Dieser Abschnitt bewertet die Auswirkungen für das iOS 10.3.3.

D.1.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

HW.02	**2**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	2.89 - Niedrig

D.1.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

OS.01	**2**
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Effizienter Brute-Force Angriff gegen Zugriffssperre durch Schwachstelle in Lightning-Anschluss.
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	5 - Kleinere Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	6.60 - Hoch
Finanzielle:	0.75 - Niedrig
Gesamt:	4.00 - Mittel

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

OS.02	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.25 - Niedrig
Gesamt:	1.33 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

OS.06	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Schwachstelle im Audio Framework beim Abspielen von OPUS Dateien.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.40 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.67 - Niedrig

D.1.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

Eine Bewertung der automatischen und manuellen Reviews der App Stores ist bei fehlender öffentlich verfügbarer Dokumentation nicht möglich.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

MGMT.03	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.56 - Niedrig

R-MGMT.05

Anforderung: Das Gerät muss nativ eine sichere Mailkommunikation anbieten.

MGMT.05	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Denial of Service durch manipulierten Bilddateianhang (CVE-2017-7097)
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	1.40 - Niedrig
Finanzielle:	0.25 - Niedrig
Gesamt:	0.89 - Niedrig

R-MGMT.10

Anforderung: Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.

MGMT.10	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff durch Link-Aufruf
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.80 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	3.11 - Mittel

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

MGMT.15	2
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff auf Apple-ID Passwort und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	2.22 - Niedrig

D.2 Samsung Galaxy S7 (Android 7.0 - Sicherheitsupdates 01.08.2017)

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für das Smartphone *Samsung Galaxy S7* aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.2 aufgelistet.

D.2.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Bewertung:

HW.02	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065 (Broadcom), CVE-2017-0781 (BlueBorne)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	2.89 - Niedrig

R-HW.03.2

Anforderung: Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.

Bewertung:

HW.03.2	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle in ARM Architektur
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	5.80 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	3.56 - Mittel

D.2.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

Bewertung:

OS.01	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Zugriff auf unverschlüsselte Daten durch Dumpen des Flashspeichers.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	3 - Kleine schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.80 - Mittel
Finanzielle:	0.75 - Niedrig
Gesamt:	2.44 - Niedrig

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

Bewertung:

OS.02	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Speicherzugriffsverletzung CVE-2017-5121, CVE-2017-5122.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.50 - Niedrig
Gesamt:	1.44 - Niedrig

R-OS.03

Anforderung: Die Updates des Betriebssystems müssen authentisch sein.

Bewertung:

OS.03	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Manuelles Downgrade auf frühere Version.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	3.60 - Mittel
Finanzielle:	0.50 - Niedrig
Gesamt:	2.22 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

Bewertung:

OS.06	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	2.78 - Niedrig

R-OS.08

Anforderung: Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.

Bewertung:

OS.08	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle im Media Framework.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	2.78 - Niedrig

D.2.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

Bewertung:

MGMT.03	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	2.78 - Niedrig

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

Bewertung:

MGMT.15	2
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Social Engineering Angriff auf Passwort des Google-Accounts und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	0 -
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	0.50 - Niedrig
Gesamt:	2.11 - Niedrig

E. Benutzergruppe 3 Konzernchef

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für die Smartphones iPhone 7 und Samsung Galaxy S7 aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.4 aufgelistet.

E.1 iPhone 7 (iOS 10.3.3)

Dieser Abschnitt bewertet die Auswirkungen der Schwachstellen für das iPhone 7.

E.1.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

HW.02	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	CVE-2017-11120 (Broadcom Treiber)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	3 -
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	6.00 - Hoch
Gesamt:	5.22 - Mittel

E.1.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

OS.01	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Effizienter Brute-Force Angriff gegen Zugriffssperre durch Schwachstelle in Lightning-Anschluss.
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	5 - Kleinere Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	6 - Verlust von Großkunden
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	6.60 - Hoch
Finanzielle:	6.75 - Hoch
Gesamt:	6.67 - Hoch

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

OS.02	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.50 - Niedrig
Gesamt:	1.44 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

OS.06	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Schwachstelle im Audio Framework beim Abspielen von OPUS Dateien.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.40 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.67 - Niedrig

E.1.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

Eine Bewertung der automatischen und manuellen Reviews der App Stores ist bei fehlender öffentlich verfügbarer Dokumentation nicht möglich.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

MGMT.03	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Ausführen von Schadcode nach Aufruf einer bösartigen Webseite.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.56 - Niedrig

R-MGMT.05

Anforderung: Das Gerät muss nativ eine sichere Mailkommunikation anbieten.

MGMT.05	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Denial of Service durch manipulierten Bilddateianhang (CVE-2017-7097)
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	0 - Kein Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	1.40 - Niedrig
Finanzielle:	0.75 - Niedrig
Gesamt:	1.11 - Niedrig

R-MGMT.10

Anforderung: Das Gerät muss eine Möglichkeit der Integration in ein bestehendes Firmennetzwerk anbieten.

MGMT.10	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff durch Link-Aufruf
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	0 - Kein Ausfall von Diensten
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	4 - Verlust betroffener Kunden
Gesetzesverstoß:	5 - Einzelne Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.80 - Mittel
Finanzielle:	5.75 - Mittel
Gesamt:	5.22 - Mittel

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

MGMT.15	Geschäftsführer
Gerät:	iPhone 7
Betriebssystem:	iOS 10.3.3
Schwachstelle:	Social Engineering Angriff auf Apple-ID Passwort und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	2.33 - Niedrig

E.2 Samsung Galaxy S7 (Android 7.0 - Sicherheitsupdates 01.08.2017)

Dieser Anhang wertet die finanziellen und technischen Auswirkungen der von den im Anhang B ermittelten Schwachstellen für das Smartphone *Samsung Galaxy S7* aus. Die erforderlichen Informationen dieser Nutzergruppe sind in Kapitel 4.4.2 aufgelistet.

E.2.1 Hardware

Dieser Abschnitt bewertet die Auswirkungen der ermittelten Schwachstellen in der verbauten Hardware des Smartphones.

R-HW.02

Anforderung: Drahtlose Schnittstellen müssen Schutz nach Stand der Technik anbieten.

Bewertung:

HW.02	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	CVE-2017-11120, CVE-2017-11121, CVE-2017-7065 (Broadcom), CVE-2017-0781 (BlueBorne)
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	7 - Umfangreiche schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	3 -
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.60 - Mittel
Finanzielle:	6.00 - Hoch
Gesamt:	5.22 - Mittel

R-HW.03.2

Anforderung: Die privaten Schlüssel müssen in hardwaregetrenntem Bereich sicher aufbewahrt werden.

Bewertung:

HW.03.2	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle in ARM Architektur
Technische Auswirkungen	
Vertraulichkeit:	9 - Alle Daten
Integrität:	9 - Komplette (Daten-)Manipulation
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	3 -
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	5.80 - Mittel
Finanzielle:	6.00 - Hoch
Gesamt:	5.89 - Mittel

E.2.2 Betriebssystem

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche vom Betriebssystem ausgehen.

R-OS.01

Anforderung: Die auf dem Gerät gespeicherten Daten müssen trotz Geräteverlust sicher vor unberechtigtem Zugriff sein.

Bewertung:

OS.01	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Zugriff auf unverschlüsselte Daten durch Dumpen des Flashspeichers.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	3 - Kleine schwerwiegende Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	3 - Kleine Auswirkungen auf jährlichen Umsatz
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	2 - Kleine Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	3.80 - Mittel
Finanzielle:	3.25 - Mittel
Gesamt:	3.56 - Mittel

R-OS.02

Anforderung: Der Browser muss einen sicheren Aufruf von Webanwendungen ermöglichen.

Bewertung:

OS.02	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Speicherzugriffsverletzung CVE-2017-5121, CVE-2017-5122.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	1 - Geringfügige Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	0 - Vollständige Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	2.20 - Niedrig
Finanzielle:	0.50 - Niedrig
Gesamt:	1.44 - Niedrig

R-OS.03

Anforderung: Die Updates des Betriebssystems müssen authentisch sein.

Bewertung:

OS.03	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Manuelles Downgrade auf frühere Version.
Technische Auswirkungen	
Vertraulichkeit:	2 - Kaum nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	0 - Kein Betroffener
Gesamtauswirkungen	
Technische:	3.60 - Mittel
Finanzielle:	0.50 - Niedrig
Gesamt:	2.22 - Niedrig

R-OS.06

Anforderung: Das Betriebssystem muss privilegierte Applikationen vor unautorisierten Zugriffen schützen.

Bewertung:

OS.06	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	6 - Verlust von Großkunden
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	6.75 - Hoch
Gesamt:	5.33 - Mittel

R-OS.08

Anforderung: Eine Anwendung darf mit anderen Applikationen nur über einen sicheren Kanal kommunizieren.

Bewertung:

OS.08	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Schwachstelle im Media Framework.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	3 -
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	6.00 - Hoch
Gesamt:	5.00 - Mittel

E.2.3 Management

Dieser Abschnitt bewertet die Auswirkungen aller ermittelten Schwachstellen, welche das Management von Geräten betreffen.

R-MGMT.03

Anforderung: Es muss die Möglichkeit bestehen, Standardanwendungen zu deinstallieren.

Bewertung:

MGMT.03	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Ausführen von Schadcode in der Anwendung SVoice mit Systemrechten.
Technische Auswirkungen	
Vertraulichkeit:	5 - Umfangreiche nicht sensitive Daten
Integrität:	5 - Umfangreiche kleinere Manipulationen
Verfügbarkeit:	1 - Kleine untergeordnete Dienste
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	7 - Signifikante Auswirkungen auf jährlichen Umsatz
Ansehen:	6 - Verlust von Großkunden
Gesetzesverstoß:	7 - Öffentlichkeitswirksame Verstöße
Privatsphäre:	7 - Tausende Betroffene
Gesamtauswirkungen	
Technische:	4.20 - Mittel
Finanzielle:	6.75 - Hoch
Gesamt:	5.33 - Mittel

R-MGMT.15

Anforderung: Die Managementsoftware muss Maßnahmen zur Sicherheit des eigenen Produktes enthalten.

Bewertung:

MGMT.15	Geschäftsführer
Gerät:	Samsung Galaxy S7
Betriebssystem:	Android 7.0 - Sicherheitsupdates 01.08.2017
Schwachstelle:	Social Engineering Angriff auf Passwort des Google-Accounts und Mitlesen der SMS über SS7-Schwachstelle(n).
Technische Auswirkungen	
Vertraulichkeit:	0 - Keine Daten
Integrität:	0 - Keine Manipulation
Verfügbarkeit:	7 - Zahlreiche Hauptdienste unterbrochen
Verantwortliche:	7 - Möglicherweise identifizierbar
Intervenierbarkeit:	3 - Teilweise Datenhoheit
Finanzielle Auswirkungen	
Materieller Schaden:	1 - Geringer als Kosten zur Behebung der Schwachstelle
Ansehen:	1 - Niedriger Ansichtsverlust
Gesetzesverstoß:	0 - Keine Verstöße
Privatsphäre:	2 - Einzelner
Gesamtauswirkungen	
Technische:	3.40 - Mittel
Finanzielle:	1.00 - Niedrig
Gesamt:	2.33 - Niedrig

F. Ausblick zu Verbesserungen in Android 8.0 & iOS 11.0

Gegen Ende der Bearbeitungszeit dieser Masterarbeit wurden fast zeitgleich von Android und Apple neue Betriebssystemversionen veröffentlicht. Dieser Anhang soll einen Einblick in die Verbesserung der Gerätesicherheit in den neuen Versionen ermöglichen.

F.1 Android 8.0

Android 8.0 wurde von Google am 21. August 2017 vorgestellt und trägt den Namen Oreo. Die Aufzählung der Verbesserungen erhält nur jene, welche die Sicherheit verbessern.

- Das unsichere Protokoll SSLv3 wird nicht mehr unterstützt.
- Beim TLS-Verbindungsaufbau wird bei der Protokollaushandlung kein Downgrade auf ältere (und potentiell unsichere) TLS-Versionen zugelassen.
- Alle Anwendungen werden mittels eines Filters (SECCOMP) vor Systemaufrufen geschützt.
- WebView-Objekte werden in einem isolierten Prozess ausgeführt, der von der aufrufenden Anwendung getrennt ist.
- Anwendungen aus unbekannten Quellen muss künftig vor jeder Installation explizit zugestimmt werden.
- Verwaltete Geräte wird automatisch die Installation aus unbekannten Quellen verweigert.

[Prob]

Für den Nutzer sichtbare Verbesserungen ist die feste Etablierung von *Google Play Protect*. Dieser Dienst scannt alle auf dem System installierten Anwendungen nach

bekannten Schadcode-Mustern ab. Diese Überprüfung beinhaltet auch Anwendungen, welche aus unbekannten Quellen installiert wurden. Damit wird sichergestellt, dass boshafte Anwendungen, welche bei den Überprüfungen im *Play Store* nicht erkannt wurden, indem sie zum Beispiel den Schadcode erst nach der Installation nachgeladen haben, fortan einer kontinuierlichen Überprüfung unterzogen werden. Sämtliche Anwendungen, welche nicht über den *Play Store* installiert wurden und damit keinerlei Überprüfung hatten, werden durch *Google Play Protect* überprüft. Das Überblenden von wichtigen System-UI Elementen durch Eigene ist seit Android 8.0 nicht mehr möglich. Damit kann das Angriffsszenario *Cloak & Dagger*[1], welches alle vorherigen Betriebssystemversionen betraf, nicht mehr durchgeführt werden.
[Scha]

In den monatlich erscheinenden Sicherheitsupdates von Android fanden im September zwei Veröffentlichungen statt. Neben einer Behebung von Schwachstellen in Broadcom- und Qualcomm-Treibern wurden diverse Lücken im Kernel und Media Framework behoben.[Prog]

F.2 iOS 11.0

Apple veröffentlichte iOS 11.0 am 19. September 2017. Der iOS Sicherheitsguide ist stand jetzt (30. Oktober 2017) noch nicht aktualisiert. Dieser Guide beinhaltete in der Vergangenheit Informationen über die Architektur und technische Funktionsweise von iOS-Geräten. Durch die fehlende Aktualisierung dieses Dokuments kann keine Vorstellung der Architekturverbesserungen erfolgen. Die recherchierten Informationen zu den Sicherheitsverbesserungen in iOS 11.0 entstammen größtenteils aus Drittquellen.

Seit iOS 11.0 ist bei der erstmaligen Verbindung eines iOS-Gerätes mit einem Computer über das Lightning-Kabel neben dem Entsperren des Gerätes und dem Akzeptieren der Verbindung, zusätzlich die Eingabe des Gerätepasswortes erforderlich. Dies führt zu der Verbesserung, dass das Gerät mittels des Fingerabdrucks zwar entsperrt werden kann, ohne das Passwort, welches im Idealfall nur der Nutzer kennt, jedoch keine Kommunikation mit einem Computer möglich ist. Durch fünfmaliges Drücken des Home Buttons kann ferner die Entsperrung des Gerätes durch den Fingerabdruck oder der Gesichtserkennung deaktiviert werden. Damit ist eine Entsperren des Gerätes ausschließlich mit dem Gerätepasswort möglich.
[Bil] Die Sicherheit des Managements von iOS-Geräten wurde durch eine zwingend erforderliche Zwei-Faktor-Authentifizierung verbessert. Zur Übermittlung des zweiten Faktors wird eine Push-Benachrichtigung versandt, welche als sicherer erachtet wird, als der unverschlüsselte Versand über SMS.[Wen]

Eine Vielzahl an existierenden Schwachstellen des Betriebssystems wurde ebenfalls durch dieses Update behoben. Die komplette Liste kann unter diesem Link[2] abgerufen werden.[Incc]

[1] http://cloak-and-dagger.org/
[2] https://support.apple.com/en-us/HT208112

Literatur

[05] *Konformitätsbewertung - Begriffe und allgemeine Grundlagen*. Norm. März 2005.

[15] *Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2015)*. Norm. Nov. 2015.

[AK16] Oleg Afonin und Vladimir Katalov. *Mobile Forensics - Advanced Investigative Strategies*. Packt Publishing, 2016.

[And] Android. *Anteile der verschiedenen Android-Versionen an allen Geräten mit Android OS weltweit im Zeitraum 26. April bis 02. Mai 2017*. URL: https://de.statista.com/statistik/daten/studie/180113/umfrage/anteil-der-verschiedenen-android-versionen-auf-geraeten-mit-android-os/ (besucht am 17. 05. 2017).

[Ast+12] Vishal Asthana u. a. *Practical Security Stories and Security Tasks for Agile Development Environments*. Techn. Ber. Software Assurance Forum for Excellence in Code, Juli 2012.

[Bec] Leo Becker. *iOS und OS X: Über AirDrop-Lücke lässt sich angeblich Malware einschleusen*. URL: https://www.heise.de/mac-and-i/meldung/iOS-und-OS-X-Ueber-AirDrop-Luecke-laesst-sich-angeblich-Malware-einschleusen-2819867.html (besucht am 05. 10. 2017).

[Ber] Freie Universität Berlin. *Grundwerte*. URL: http://www.fu-berlin.de/sites/it-sicherheit/grundwerte/index.html (besucht am 25. 09. 2017).

[Bil] Denise Bilic. *Security and privacy on the new iOS 11*. URL: https://www.welivesecurity.com/2017/09/25/security-privacy-new-ios-11/ (besucht am 25. 09. 2017).

[Bir] Jim Bird. *Adding Appsec to Agile: Security Stories, Evil User Stories and Abuse(r) Stories*. URL: http://swreflections.blogspot.de/2013/10/adding-appsec-to-agile-security-stories.html (besucht am 10. 09. 2017).

[Bon] Boniversum. *Anteil der Befragten, die mobile Geräte zum Einkauf im Internet verwenden in den Jahren 2011 bis 2016*. URL: https://de.statista.com/statistik/daten/studie/311650/umfrage/nutzung-von-smartphone-und-tablet-zum-mobilen-einkauf/ (besucht am 21. 10. 2017).

[Bra15] Christian Braun. *Computernezte kompakt*. Springer Berlin Heidelberg, 2015.

[Bra17] Christian Braun. *Betriebssysteme kompakt*. Springer Berlin Heidelberg, 2017.

[BSP] Prof. Dr.-Ing. Carsten Bormann, Dr. Karsten Sohr und Niels Pollem. *IT-Sicherheit: Zugriffskontrolle*. Vorlesungsskript im Sommersemester 2005 (Universität Bremen).

[Bud+14] Jens Budszus u. a. *Orientierungshilfe – Cloud Computing*. Techn. Ber. Der Bayerische Landesbeauftragte für den Datenschutz, 2014.

[Bun17] Statistisches Bundesamt. *Bevölkerung in Deutschland voraussichtlich auf 82,8 Millionen gestiegen*. 2017. URL: https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2017/01/PD17_033_12411pdf.pdf?__blob=publicationFile (besucht am 11. 05. 2017).

[Coma] OWASP Community. *About The Open Web Application Security Project*. URL: https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project (besucht am 01. 09. 2017).

[Comb] OWASP Community. *Agile Software Development: Don't Forget EVIL User Stories*. URL: https://www.owasp.org/index.php/Agile_Software_Development:_Don%27t_Forget_EVIL_User_Stories (besucht am 01. 09. 2017).

[Comc] OWASP Community. *Mobile Top 10 2016-Top 10*. URL: https://www.owasp.org/index.php?title=Mobile_Top_10_2016-Top_10&oldid=226317 (besucht am 01. 09. 2017).

[Comd] OWASP Community. *OWASP Mobile Security Testing Guide*. URL: https://www.owasp.org/index.php?title=OWASP_Mobile_Security_Testing_Guide&oldid=231491 (besucht am 01. 09. 2017).

[Come] OWASP Community. *OWASP Project - Project Inventory*. URL: https://www.owasp.org/index.php?title=Category:OWASP_Project&oldid=232765#tab=Project_Inventory (besucht am 01. 09. 2017).

[Comf] OWASP Community. *OWASP Risk Rating Methodology*. URL: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology (besucht am 03. 09. 2017).

[Comg] OWASP Community. *Secure Mobile Development Guidelines Objective*. URL: https://www.owasp.org/index.php?title=OWASP_Mobile_Security_Project&oldid=229187 (besucht am 01. 09. 2017).

[Comh] OWASP Community. *Top 10 Mobile Controls*. URL: https://www.owasp.org/index.php?title=OWASP_Mobile_Security_Project&oldid=229187 (besucht am 01. 09. 2017).

[Comi] Samsung Electronics Company. *Android Security Update Details | Samsung Mobile Security*. URL: https://security.samsungmobile.com/securityUpdate.smsb (besucht am 06. 10. 2017).

[Comj] Samsung Electronics Company. *Flexibel und sicher: Samsung Enterprise Edition*. URL: https://news.samsung.com/de/flexibel-und-sicher-samsung-enterprise-edition (besucht am 12. 10. 2017).

[Comk] Samsung Electronics Company. *Samsung Devices – Now Validated Through Common Criteria and FIPS*. URL: http://www.samsung.com/us/business/security/knox/assets/pdf/hhp-common-criteria-fips-sep-15.pdf (besucht am 13. 10. 2017).

[Coml] Samsung Electronics Company. *Samsung Internet – Release Note*. URL: http://developer.samsung.com/internet/android/releases (besucht am 25. 08. 2017).

[Comm] Samsung Electronics Company. *Samsung Open Source Release Center*. URL: http://opensource.samsung.com/reception/receptionSub.do?method=sub&sub=F&searchValue=sm-g930 (besucht am 13. 10. 2017).

[com] comScore. *Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2016 (in Millionen)*. URL: https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010 (besucht am 11. 05. 2017).

[Cora] Microsoft Corporation. *11. April 2017 – KB4015217 (Betriebssystembuild 14393.1066 und 14393.1083)*. URL: https://support.microsoft.com/de-de/help/4015217/windows-10-update-kb4015217 (besucht am 02. 08. 2017).

[Corb] MITRE Corporation. *Apple Safari: CVE security vulnerabilities, versions and detailed reports*. URL: https://www.cvedetails.com/product/2935/Apple-Safari.html (besucht am 06. 10. 2017).

[Corc] MITRE Corporation. *Google Authenticator: List of security vulnerabilities*. URL: http://www.cvedetails.com/product/25099/Google-Authenticator.html?vendor_id=1224 (besucht am 11. 10. 2017).

[CRP14] Tim Cooijmans, de Joeri Ruiter und Erik Poll. "Analysis of Secure Key Storage Solutions on Android". In: *Radboud University Nijmegen* (2014).

[Das+13] Amitabh "Das u. a. "Secure JTAG Implementation Using Schnorr Protocol". In: *Journal of Electronic Testing* (2013).

[Eik] Ronald Eikenberg. *Stagefright: Android-Smartphones über Kurznachrichten angreifbar*. URL: https://www.heise.de/security/meldung/Stagefright-Android-Smartphones-ueber-Kurznachrichten-angreifbar-2763764.html (besucht am 21. 10. 2017).

[Est] Tom Eston. *Mobile Threat Agend identifiers and types*. URL: https://www.owasp.org/index.php?title=OWASP_Mobile_Security_Project&oldid=229187 (besucht am 01. 09. 2017).

[Eur] Samsung Electronics Europe. *Exynos 8 Octa (8890)*. URL: http://www.samsung.com/semiconductor/minisite/Exynos/Solution/MobileProcessor/Exynos_8_Octa_8890.html (besucht am 29. 09. 2017).

[Eve15] Florian Evers. "Zeitgemäße Linux-IPC". In: *Linux Magazin* (Feb. 2015).

[FBT13] D. Fischer, Markscheffel B. und Seyffarth T. "An overview of threats and security software solutions for smartphones." In: *International Journal for Information Security Research (IJISR)* 3 (2013).

[fin] finanzen.net. *Microsoft beendet Smartphone-Experiment*. URL: http://www.finanzen.net/nachricht/aktien/lumia-verkauf-eingestellt-microsoft-beendet-smartphone-experiment-5256434 (besucht am 02. 08. 2017).

[Fit15] J.R. Fitzer. *Agile Information Security: Using Scrum to Survive in and Secure a Rapidly Changing Environment*. BookBaby, 2015.

[Gie] Hauke Gierow. *Hacker veröffentlicht Secure-Enclave-Key für alte iPhones*. URL: https://www.golem.de/news/apple-iphone-5s-hacker-veroeffentlicht-secure-enclave-key-fuer-alte-iphones-1708-129562.html (besucht am 29. 09. 2017).

[GMA14] Slawomir Grzonkowski, Alejandro Mosquera und Lamine Aouad. "Smartphone Security: An overview of emerging threats." In: *IEEE Consumer Electronics Magazine* 3 (Okt. 2014).

[Has] Job de Hass. *20 ways to past secure boot*. URL: https://conference.hitb.org/hitbseccconf2013kul/materials/D2T3%20-%20Job%20de%20Haas%20-%2020%20Ways%20Past%20Secure%20Boot.pdf (besucht am 01. 10. 2017).

[HE12] Jens Heider und Rachid El Khayari. "Geht Ihr Smarthpone fremd?" In: *Datenschutz und Datensicherheit* 3 (2012).

[Her] Daniel Herbig. *Blackberry KEYone: Android-Smartphone kommt am 16. Mai nach Deutschland*. URL: https://www.heise.de/newsticker/meldung/Blackberry-KEYone-Android-Smartphone-kommt-am-16-Mai-nach-Deutschland-3712391.html (besucht am 03. 08. 2017).

[HM08] Wolfgang Hesse und Heinrich C. Mayr. "Modellierung in der Softwaretechnik: eine Bestandsaufnahme". In: *Informatik-Spektrum* 31.5 (Aug. 2008), S. 377–393.

[Hof] Prof. Dr. Joachim Hof. *Software-Technik für sicherheitskritische Systeme - Grundlagen 02.02*. Vorlesungsskript im Wintersemester 2016/2017 (Technische Hochschule Ingolstadt.

[IAa] Google Inc und Open Handset Alliance. *Android Interfaces and Architecture | Android Open Source Project*. URL: https://source.android.com/devices/ (besucht am 15. 05. 2017).

[IAb] Google Inc und Open Handset Alliance. *Application security | Android Open Source Project*. URL: https://source.android.com/security/verifiedboot/verified-boot (besucht am 16. 05. 2017).

[IAc] Google Inc und Open Handset Alliance. *Application Signing | Android Open Source Project*. URL: https://source.android.com/security/apksigning/ (besucht am 16. 05. 2017).

[IAd] Google Inc und Open Handset Alliance. *Authentication | Android Open Source Project*. URL: https://source.android.com/security/authentication (besucht am 15. 05. 2017).

[IAe] Google Inc und Open Handset Alliance. *Encryption | Android Open Source Project*. URL: https://source.android.com/security/encryption/ (besucht am 15. 05. 2017).

[IAf] Google Inc und Open Handset Alliance. *Full-Disk Encryption | Android Open Source Project*. URL: https://source.android.com/security/encryption/full-disk (besucht am 15. 05. 2017).

[IAg] Google Inc und Open Handset Alliance. *Hardware-backed Keystore | Android Open Source Project*. URL: https://source.android.com/security/keystore/ (besucht am 15. 05. 2017).

[IAh] Google Inc und Open Handset Alliance. *Hardware-backed Keystore | Android Open Source Project*. URL: https://source.android.com/security/keystore/ (besucht am 15. 05. 2017).

[IAi] Google Inc und Open Handset Alliance. *lollipop-release - platform/build - Git at Google*. URL: https://android.googlesource.com/platform/build/+/lollipop-release (besucht am 12. 06. 2017).

[IAj] Google Inc und Open Handset Alliance. *Platform Architecture | Android Developers*. URL: https://developer.android.com/guide/platform/index.html (besucht am 15. 05. 2017).

[IAk] Google Inc und Open Handset Alliance. *Security Enhancement in Android 5.0 | Android Open Source Project*. URL: https://source.android.com/security/enhancements/enhancements50 (besucht am 16. 05. 2017).

[IAl] Google Inc und Open Handset Alliance. *Security Enhancement in Android 6.0 | Android Open Source Project*. URL: https://source.android.com/security/enhancements/enhancements60 (besucht am 16. 05. 2017).

[IAm] Google Inc und Open Handset Alliance. *Security Enhancement in Android 7.0 | Android Open Source Project*. URL: https://source.android.com/security/enhancements/enhancements70 (besucht am 16. 05. 2017).

[IAn] Google Inc und Open Handset Alliance. *Security-Enhanced Linux in Android | Android Open Source Project*. URL: https://source.android.com/security/selinux/ (besucht am 15. 05. 2017).

[IAo] Google Inc und Open Handset Alliance. *System and kernel security | Android Open Source Project*. URL: https://source.android.com/security/overview/kernel-security (besucht am 15. 05. 2017).

[IAp] Google Inc und Open Handset Alliance. *Trusty TEE | Android Open Source Project*. URL: https://source.android.com/security/trusty/ (besucht am 15. 05. 2017).

[IAq] Google Inc und Open Handset Alliance. *Verifying Boot | Android Open Source Project*. URL: https://source.android.com/security/verifiedboot/verified-boot (besucht am 16. 05. 2017).

[Inca] Apple Inc. *About the iOS Technologies*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html (besucht am 17. 05. 2017).

[Incb] Apple Inc. *About the iOS Technologies*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html#//apple_ref/doc/uid/TP40007898-CH9-SW4 (besucht am 17. 05. 2017).

[Incc] Apple Inc. *About the security content of iOS 11*. URL: https://support.apple.com/en-us/HT208112 (besucht am 05. 10. 2017).

[Incd] Apple Inc. *Adding Capabilities*. URL: https://developer.apple.com/library/content / documentation / IDEs / Conceptual / AppDistributionGuide / AddingCapabilities/AddingCapabilities.html (besucht am 05. 10. 2017).

[Ince] Apple Inc. *App Sandbox entitlement Keys*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Reference/EntitlementKeyReference/Chapters/EnablingAppSandbox.html (besucht am 05. 10. 2017).

[Incf] Apple Inc. *Apple security updates*. URL: https://support.apple.com/en-us/HT201222 (besucht am 11. 10. 2017).

[Incg] Apple Inc. *Cocoa Touch Layer*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html#//apple_ref/doc/uid/TP40007898-CH3-SW1 (besucht am 17. 05. 2017).

[Inch] Apple Inc. *Configuration Profile Reference*. URL: https://developer.apple.com/library/content/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html#//apple_ref/doc/uid/TP40010206-CH1-SW7 (besucht am 10. 10. 2017).

[Inci] Apple Inc. *Core OS Layer*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html#//apple_ref/doc/uid/TP40007898-CH11-SW1 (besucht am 17. 05. 2017).

[Incj] Apple Inc. *Core Services Layer*. URL: https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html#//apple_ref/doc/uid/TP40007898-CH10-SW5 (besucht am 17. 05. 2017).

[Inck] Apple Inc. *Inter App Communication*. URL: https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Inter-AppCommunication/Inter-AppCommunication.html (besucht am 18. 05. 2017).

[Incl] Apple Inc. *iOS Deployment Overview for Business*. URL: https://www.apple.com/business/resources/docs/iOS_Deployment_Overview_Business.pdf (besucht am 10. 10. 2017).

[Incm] Apple Inc. *iOS Security - iOS 10*. URL: https://www.apple.com/business/docs/iOS_Security_Guide.pdf (besucht am 17. 05. 2017).

[Incn] Apple Inc. *iPhone 7*. URL: https://www.apple.com/de/iphone-7/specs/ (besucht am 11. 10. 2017).

[Inco] Apple Inc. *Remove built-in Apple apps from the Home screen on your iOS 10 device or Apple Watch*. URL: https://support.apple.com/de-de/HT204221 (besucht am 06. 10. 2017).

[Incp] Apple Inc. *Two-factor authentication for Apple ID*. URL: https://support.apple.com/en-us/HT204915 (besucht am 11. 10. 2017).

[Incq] Apple Inc. *Validating Input and Interprocess Communication*. URL: https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Articles/ValidatingInput.html#//apple_ref/doc/uid/TP40007246 (besucht am 18. 05. 2017).

[Incr] Google Inc. *Add 2-Step Verification*. URL: https://support.google.com/a/answer/175197?hl=en (besucht am 11. 10. 2017).

[Incs] Google Inc. *Android Geschichte*. URL: https://www.android.com/history (besucht am 12. 06. 2017).

[Inct] Google Inc. *Delete or disable apps on Android*. URL: https://support.google.com/googleplay/answer/2521768?hl=en (besucht am 08. 10. 2017).

[Incu] Google Inc. *Google Play Protect*. URL: https://www.android.com/intl/de_de/play-protect/ (besucht am 08. 10. 2017).

[Incv] Google Inc. *Help prevent others from using your device without permission*. URL: https://support.google.com/nexus/answer/6172890 (besucht am 06. 10. 2017).

[Incw] Google Inc. *Manage apps on mobile devices*. URL: https://support.google.com/a/answer/6328701?hl=en (besucht am 10. 10. 2017).

[Incx] Google Inc. *Manage my Android device*. URL: https://support.google.com/a/users/answer/1235372?hl=en (besucht am 10. 10. 2017).

[Incy] Google Inc. *Manage your organization's mobile devices*. URL: https://support.google.com/a/answer/1734200 (besucht am 09. 10. 2017).

[Incz] Google Inc. *Mehr Sicherheit für Nachrichten mit gehostetem S/MIME*. URL: https://support.google.com/a/answer/7280976?hl=de (besucht am 08. 10. 2017).

[Incaa] Google Inc. *Stable Channel Updates for Chrome OS*. URL: https://chromereleases.googleblog.com/2017/10/stable-channel-updates-for-chrome-os.html (besucht am 25. 08. 2017).

[Incab] Trend Micro Incorporated. *Address bar spoofing*. URL: https://www.trendmicro.com/vinfo/us/security/definition/address-bar-spoofing (besucht am 06. 10. 2017).

[Incac] Trend Micro Incorporated. *Analyzing Xavier: An Information-Stealing Ad Library on Android*. URL: http://blog.trendmicro.com/trendlabs-security-intelligence/analyzing-xavier-information-stealing-ad-library-android/ (besucht am 06. 10. 2017).

[Inf] Vorlesung Modellierung - Modellierungsmethoden der Informatik. *König, Prof. Barbara*. Vorlesungsskript im Wintersemester 2011/2017 (Universität Duisburg-Essen).

[Ins] Business Insider. *Marktanteile der Betriebssysteme am weltweiten Endkundenabsatz von Smartphones im 4. Quartal der Jahre 2015 und 2016*. URL: https://de.statista.com/statistik/daten/studie/76081/umfrage/weltweite-marktanteile-der-betriebssysteme-fuer-smartphones/ (besucht am 12. 05. 2017).

[KA14] Sathya Prakash Kadhirvelan und Söderberg-Rivkin Andrew. "Threat Modelling and Risk Assessment". Masterarbeit. University of Gothenburg, 2014.

[Kao] Prof. Dr. Odej Kao. *Betriebssysteme*. Vorlesungsskript im Sommersemeser 2005 (Universität Paderborn).

[Kas16] Kaspersky. *Mobile Malware Evolution*. Techn. Ber. 2016.

[Kep13] Vitali Keppel. "Klassifikation und Analyse von IT-Sicherheitsmodellen". Masterarbeit. Universität Koblenz, 2013.

[Kim] Arnold Kim. *Apple's Ability to Deactivate Malicious App Store Apps*. URL: https://www.macrumors.com/2008/08/06/apples-ability-to-deactivate-malicious-app-store-apps/ (besucht am 09. 10. 2017).

[Kle09] Stephan Kleuker. *Formale Modelle der Softwareentwicklung*. Vieweg+Teubner Verlag, 2009.

[Lab] Kaspersky Lab. *Top 7 Mobile Security Threats: Smart Phones, Tablets and Mobile Internet Devices – What the Future has in Store*. URL: https://usa.kaspersky.com/resource-center/threats/top-seven-mobile-security-threats-smart-phones-tablets-and-mobile-internet-devices-what-the-future-has-in-store (besucht am 21. 10. 2017).

[Lim] BlackBerry Limited. *BlackBerry Reports 89 Per Cent Year Over Year Growth in GAAP Software and Services Revenue for Q2 Fiscal 2017; Company Announces New Strategic Direction for Mobility Solutions*. URL: https://global.blackberry.com/en/company/newsroom/press?id=2090412 (besucht am 02. 08. 2017).

[LR10] Prof. Dr. Josef Langer und Michael Roland. *Anwendungen und Technik von Near Field Communication (NFC)*. Springer Berlin Heidelberg, 2010.

[LS07] Lijun Liao und Jörg Schwenk. "End-to-End Header Protection in Signed S/MIME". In: *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS: OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part II*. Hrsg. von Robert Meersman und Zahir Tari. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 1646–1658.

[Mac] Macwrench. *Liste aller iOS Versionen - Macwrench*. URL: http://www.macwrench.de/wiki/Vorlage:Liste_aller_iOS_Versionen (besucht am 12. 06. 2017).

[MC03] J.D. Meier und Microsoft Corporation. *Improving Web Application Security: Threats and Countermeasures*. Patterns & practices. Microsoft, 2003.

[McA] McAfee. *Three Mobile Threats Impacting Android Smartphones and Tablets*. URL: https://www.mcafee.com/de/security-awareness/articles/three-mobile-threats-impacting-android.aspx (besucht am 21. 10. 2017).

[McA17] McAfee. *Mobile Threat Report*. Techn. Ber. 2017.

[Med] VuMA (Arbeitsgemeinschaft Verbrauchs- und Medienanalyse). *Anzahl der Personen in Deutschland, die in ihrem Haushalt ein Mobiltelefon besitzen, nach hauptsächlicher Nutzung von Privat- oder Firmen-Handy/-Smartphone, von 2013 bis 2016 (in Millionen)*. URL: https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010 (besucht am 08. 10. 2017).

[Mo17] Bernhard Mueller und owasp. *OWASP Mobile Security Testing Guide - Early Access*. leanpub, 2017.

[Mor] Dr. rer. nat. Frank Morherr. *Modellieren im Mathematik-Unterricht*. Vortrag Universität Giessen.

[Mou15] Hassan Mourad. "The Fall of SS7 – How Can the Critical Security Controls Help?" In: *SANS Institute* (2015).

[Mue17] Bernhard Mueller. *Mobile AppSec Verification*. Open Web Application Security Project (OWASP). Sep. 2017.

[NG17] Ilya Nesterov und Maxim Goncharov. "All your emails belong to us: exploiting vulnerable email clients via domain name collision". In: *Blackhat Asia 2017* (März 2017).

[NSA13] Information Assurance Directorate (NSA). "Security Requirements for Mobile Operating Systems". In: (Jan. 2013).

[Osc03] Per Oscarson. "Information Security Fundamentals". In: *Security Education and Critical Infrastructures: IFIP TC11 / WG11.8 Third Annual World Conference on Information Security Education (WISE3) June 26–28, 2003, Monterey, California, USA*. Hrsg. von Cynthia Irvine und Helen Armstrong. Boston, MA: Springer US, 2003, S. 95–107.

[Pay] PayPal. *Volumen der mobilen Zahlungen via PayPal in den Jahren 2008 bis 2016 (in Millionen US-Dollar)*. URL: https://de.statista.com/statistik/daten/studie/218465/umfrage/volumen-der-mobilen-zahlungen-via-paypal/ (besucht am 21. 10. 2017).

[Pin] Ralph Pini. *Secure, Protected, and Connected: BlackBerry 10 Achieves Government-Grade Security Certification*. URL: http://blogs.blackberry.com/2016/12/secure-protected-and-connected-blackberry-10-achieves-government-grade-security-certification/ (besucht am 02. 08. 2017).

[PP03] C.P. Pfleeger und S.L. Pfleeger. *Security in Computing*. Prentice Hall professional technical reference. Prentice Hall PTR, 2003.

[Pre] Sebastian Preetz. *Grundlagen der Modellierung*.

[Proa] Android Open Source Project. *Alternative distribution options | Android Developers*. URL: https://developer.android.com/distribute/marketing-tools/alternative-distribution.html (besucht am 07. 10. 2017).

[Prob] Android Open Source Project. *Android 8.0 Behavior Changes | Android Developers*. URL: https://developer.android.com/about/versions/oreo/android-8.0-changes.html#security-all (besucht am 28. 09. 2017).

[Proc] Android Open Source Project. *Android Security Bulletin - April 2017*. URL: https://source.android.com/security/bulletin/2017-04-01 (besucht am 29. 09. 2017).

[Prod] Android Open Source Project. *Android Security Bulletin - July 2017*. URL: https://source.android.com/security/bulletin/2017-07-01 (besucht am 29. 09. 2017).

[Proe] Android Open Source Project. *Android Security Bulletin - June 2017*. URL: https://source.android.com/security/bulletin/2017-06-01 (besucht am 29. 09. 2017).

[Prof] Android Open Source Project. *Android Security Bulletin - May 2017*. URL: https://source.android.com/security/bulletin/2017-05-01 (besucht am 29. 09. 2017).

[Prog] Android Open Source Project. *Android Security Bulletin - September 2017*. URL: https://source.android.com/security/bulletin/2017-09-01 (besucht am 29. 09. 2017).

[Proh] Android Open Source Project. *Android Security Bulletin - September 2017*. URL: https://source.android.com/security/bulletin/2017-08-01 (besucht am 29. 09. 2017).

[Proi] Android Open Source Project. *Android Security Bulletin - September 2017*. URL: https://source.android.com/security/bulletin/2017-10-01 (besucht am 29. 09. 2017).

[Proj] Android Open Source Project. *Application Security*. URL: https://source.android.com/security/overview/app-security (besucht am 05. 10. 2017).

[Prok] Android Open Source Project. *Content License*. URL: https://source.android.com/source/licenses (besucht am 11. 10. 2017).

[Prol] Android Open Source Project. *Implementing Security | Android Open Source Project*. URL: https://source.android.com/security/overview/implement (besucht am 06. 10. 2017).

[Prom] Android Open Source Project. *OTA Updates*. URL: https://source.android.com/devices/tech/ota/ (besucht am 29. 09. 2017).

[Pron] Android Open Source Project. *permission | Android Developers*. URL: https://developer.android.com/guide/topics/manifest/permission-element.html#plevel (besucht am 05. 10. 2017).

[Proo] Android Open Source Project. *Security*. URL: https://source.android.com/security/ (besucht am 11. 10. 2017).

[PYJ14] Jong Hyuk Park, Ki Jung Yi und Young-Sik Jeong. "An enhanced smartphone security model based on information security management system (ISMS)". In: *Electronic Commerce Research* 14.3 (Nov. 2014), S. 321–348.

[Ric] Haines Richard. *Multi-Level Security and Multi-Category Security*. URL: https://selinuxproject.org/w/?title=NB_MLS&oldid=1792 (besucht am 18. 09. 2017).

[Rüd] Prof. Dr. Grimm Rüdiger. *IT-Sicherheitsmodelle*. Arbeitsberichte aus dem Fachbereich Informatik Nr. 3/2008.

[Sau15] Martin Sauter. *Grundkurs Mobile Kommunikationssysteme*. 6. Aufl. Springer Fachmedien Wiesbaden, 2015.

[Scha] Fabian Scherschel. *Android Oreo: Das sind die Sicherheits-Neuerungen bei Android 8.0*. URL: https://www.heise.de/security/meldung/Android-Oreo-Das-sind-die-Sicherheits-Neuerungen-bei-Android-8-0-3812341.html (besucht am 03. 09. 2017).

[Schb] Fabian Scherschel. *Sidestepper: Sicherheitsrisiken von Mobile Device Management mit iOS*. URL: https://www.heise.de/security/meldung/Sidestepper-Sicherheitsrisiken-von-Mobile-Device-Management-mit-iOS-3159648.html (besucht am 09. 10. 2017).

[Schc] Ben Schwan. *Aktuelle iOS-Version anfällig für PIN-Code-Knacker*. URL: https://www.heise.de/mac-and-i/meldung/Aktuelle-iOS-Version-anfaellig-fuer-PIN-Code-Knacker-3808240.html (besucht am 21. 08. 2017).

[Sch+16] Gabriel Schulz u. a. *Das Standard-Datenschutzmodell*. Techn. Ber. AK Technik der Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder, 2016.

[Sch13] G. Schmitt. *Architectura et Machina: Computer Aided Architectural Design und Virtuelle Architektur*. Vieweg+Teubner Verlag, 2013.

[Sch14] Tobias Schüttler. *NAVSTAR GPS*. Springer Berlin Heidelberg, 2014.

[Sica] Bundesamt für Sicherheit in der Informationstechnik. *Anwendungsbeispiel für eine kleine Institution*. Techn. Ber. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Hilfsmittel/Profile/it-grundschutz_profil_klein.pdf?__blob=publicationFile (besucht am 26. 10. 2017).

[Sicb] Bundesamt für Sicherheit in der Informationstechnik. *Glossar und Begriffsdefinitionen*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html (besucht am 20. 10. 2017).

[Sicc] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz: 4.2 Schutzbedarfskategorien*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzAbout/ITGrundschutzSchulung/WebkursITGrundschutz/Schutzbedarfsfeststellung/Schutzbedarfskategorien/schutzbedarfskategorien_node.html (besucht am 22. 05. 2017).

[Sicd] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz: Kapitel 4: Schutzbedarfsfestellung*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzSchulung/WebkursITGrundschutz/Schutzbedarfsfeststellu schutzbedarfsfeststellung_node.html (besucht am 22. 05. 2017).

[Sice] Bundesamt für Sicherheit in der Informationstechnik. *IT-GrundschutzL: 4.2 Schutzbedarfskategorien*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzSchulung/WebkursITGrundschutz/Risikoanalyse/Vorgehensweisen/vorgehensweisen_node.html (besucht am 21. 09. 2017).

[Sicf] Bundesamt für Sicherheit in der Informationstechnik. *Leitfaden Informationssicherheit*. IT-Grundschutz kompakt.

[Sicg] Bundesamt für Sicherheit in der Informationstechnik. *Sichere Nutzung von WLAN (ISi-WLAN)*. BSI-Leitlinie zur Internet-Sicherheit (ISi-L).

[Sich] Bundesamt für Sicherheit in der Informationstechnik. *Sicheres mobiles Arbeiten*. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/Sicheres-Mobiles-Arbeiten.pdf (besucht am 03. 10. 2017).

[Sici] Bundesamt für Sicherheit in der Informationstechnik. *TLS/SSL Best Practice*. BSI-CS 012.

[Sic11] Bundesamt für Sicherheit in der Informationstechnik. *Basis-Sicherheitscheck*. IT-Grundschutz kompakt. Apr. 2011.

[Sin] Manish Singh. *Google Play Gets Fingerprint Payment Authentication for Android 6.0 Marshmallow*. URL: http://gadgets.ndtv.com/apps/news/google-play-gets-fingerprint-payment-authentication-for-android-60-marshmallow-756275 (besucht am 01. 10. 2017).

[Sju15] Insa Sjurts. *Gabler Lexikon Medienwirtschaft*. Gabler Verlag, 2015.

[Sne16] Bruce Snell. *Mobile Threat Report*. Techn. Ber. McAfee, 2016.

[Sok] Daniel Sokolov. *Google I/O: Chrome und Drive für iOS, Docs jetzt auch offline*. URL: https://www.heise.de/newsticker/meldung/Google-I-O-Chrome-und-Drive-fuer-iOS-Docs-jetzt-auch-offline-1628564.html (besucht am 06. 10. 2017).

[Spr17] Dr.-Ing. Michael Spreitzenbarth. *MObile Hacking*. dpunkt.verlag, 2017.

[ST] National Institute of Standards und Technology. *Cryptographic Standards and Guidelines*. URL: https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development (besucht am 29. 09. 2017).

[Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag Wien New York, 1973.

[Ste+17] Marc Stevens u. a. *The first collision for full SHA-1*. Techn. Ber. CWI Amsterdam, Feb. 2017.

[Sut16] Elliott Suthers. *Wi-Fi Alliance® introduces low power, long range Wi-Fi HaLow™*. 2016. URL: http://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-low-power-long-range-wi-fi-halow (besucht am 10. 05. 2017).

[SV17] Ben Seri und Gregory Vishnepolsky. *The dangers of Bluetooth implementations: Unveiling zero day vulnerabilities and security flaws in modern Bluetooth stacks*. Techn. Ber. Armis Labs, 2017. URL: http://go.armis.com/hubfs/BlueBorne%20Technical%20White%20Paper.pdf.

[Tea] Checkpoint Research Team. *More Than 1 Million Google Accounts Breached by Gooligan*. URL: https://blog.checkpoint.com/2016/11/30/1-million-google-accounts-breached-gooligan/ (besucht am 10. 10. 2017).

[TH] Prof. Dr. Jean-Paul Thommen und Prof. Dr. Gustav Horn. *Stichwort: Modell*. URL: http://wirtschaftslexikon.gabler.de/Archiv/495/modell-v11.html (besucht am 09. 05. 2017).

[Tho02] Prof. Dr. Maraco Thomas. "Informatische Modellbildung - Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht". dissertation. Universität Potsdam, 2002.

[TJ11] "PGP". In: *Encyclopedia of Cryptography and Security*. Hrsg. von Henk C. A. van Tilborg und Sushil Jajodia. Boston, MA: Springer US, 2011, S. 927–927.

[Tol] Sami Tolvanen. *Strictly Enforced Verified Boot with Error Correction | Android Developers Blog*. URL: https://android-developers.googleblog.com/2016/07/strictly-enforced-verified-boot-with.html (besucht am 16. 05. 2017).

[TS17] Simha Tang Adrian Sethumadhavan und Salvatore Stolfo. *CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management*. Techn. Ber. Columbia University, 2017.

[UM15] T. UcedaVelez und M.M. Morana. *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Wiley, 2015.

[USB] All USB. *USB History*. URL: http://www.allusb.com/usb-history (besucht am 11. 05. 2017).

[Wal] Vicki Walker. *Security and Productivity is Key for Small Businesses, Too. DTEK50 and BlackBerry Software are the Solution*. URL: http://blogs.blackberry.com/2016/08/security-and-productivity-is-key-for-small-businesses-too-dtek50-and-blackberry-software-are-the-solution/ (besucht am 02. 08. 2017).

[Wel] Ryan Welton. *Remote Code Execution as System User on Samsung Phones*. URL: 2017-10-21 (besucht am).

[Wen] Martin Wendel. *iOS 11 and macOS High Sierra: Wechsel auf Zwei-Faktor-Authentifizierung notwendig*. URL: https://www.apfeltalk.de/magazin/news/ios-11-macos-high-sierra-wechsel-auf-zwei-faktor-authentifizierung-notwendig/ (besucht am 30. 09. 2017).

[Wer01] Prof. Dr. Benno Werlen. *Sozialwissenschaftliche Modelle*. 2001. URL: http://www.spektrum.de/lexikon/geographie/sozialwissenschaftliche-modelle/7405&_druck=1 (besucht am 09. 05. 2017).

[Wil] Andreas Wilkens. *Gerichtliche Anordnung zum iPhone-Entsperren: Apple-Chef Tim Cook widersetzt sich*. URL: https://www.heise.de/newsticker/meldung/Gerichtliche-Anordnung-zum-iPhone-Entsperren-Apple-Chef-Tim-Cook-widersetzt-sich-3107769.html (besucht am 30. 10. 2017).

[WM16] M.E. Whitman und H.J. Mattord. *Management of Information Security*. Cengage Learning, 2016.

[Zer] Project Zero. *Broadcom: OOB write when handling 802.11k Neighbor Report Response*. URL: https://bugs.chromium.org/p/project-zero/issues/detail?id=1289#c3 (besucht am 30. 09. 2017).

[Zie17] Peter-Michael Ziegler. "Ihr Geld wird smart". In: *c't - Magazin für Computertechnik* 07 (2017), S. 120–126.

[Živ17] Dušan Živadinović. *Kleine Umwälzung - Wie Bluetooth 5 Reichweite und Datenrate erhöht*. 2017. URL: http://heise.de/-3575330 (besucht am 10. 05. 2017).