h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Hochschule Darmstadt

– Fachbereich Informatik –

Leitfaden zur forensischen Untersuchung von Android-Smartphones

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt von

Denise Muth
Matrikel-Nr. 714018

Referent: Prof. Dr. Harald Baier
Korreferent: Prof. Dr. Ralf Hahn

Ausgabedatum: 14. Februar 2013
Abgabedatum: 14. August 2013

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Soweit ich auf fremde Materialien, Texte oder Gedankengänge zurückgegriffen habe, enthalten meine Ausführungen vollständige und eindeutige Verweise auf die Urheber und Quellen. Alle weiteren Inhalte der vorgelegten Arbeit stammen von mir im urheberrechtlichen Sinn, soweit keine Verweise und Zitate erfolgen. Mir ist bekannt, dass ein Täuschungsversuch vorliegt, wenn die vorstehende Erklärung sich als unrichtig erweist.

Darmstadt, den 14. August 2013

Abstrakt

Smartphones bieten durch eine leistungsstarke Hardware, hohe Speicherkapazitäten, ein Betriebssystem mit offenen Schnittstellen und eine permanente Datenverbindung vielseitige Einsatzmöglichkeiten. Diese führen zu einer wachsenden Bedeutung der Geräte im Bereich der IT-Forensik. Vor allem Geräte basierend auf dem Betriebssystem „Android“ weisen hierbei gemäß ihrer herausragenden Stellung am Markt eine hohe Verbreitung auf. Im Zuge eines Ermittlungsverfahrens können die auf dem System gespeicherten Daten als potentielle Spuren zur Aufklärung einer Straftat genutzt werden.

Die Methoden zur forensischen Untersuchung weichen aufgrund der besonderen technischen Gegebenheiten von Smartphones wesentlich von denen der klassischen IT-Forensik ab. So erfordert beispielsweise die Sicherung der Speichermedien (Datensammlung) ein Umdenken, da eine einfache Entnahme des fest verdrahteten Flash-Speichers, vergleichbar zum Ausbau einer Festplatte aus einem Computer, nicht möglich ist. Bisherigen Ausarbeitungen auf dem Gebiet der forensischen Untersuchung von Android-Smartphones fehlt entweder der betriebssystemspezifische Fokus oder eine prägnante und an anerkannte Prozessmodelle angelehnte Systematik zur Durchführung. Daraus resultiert im ungünstigsten Fall eine falsche Handhabung der Geräte, die zum Verlust von essentiellen Beweismitteln führt.

Zielsetzung der Masterarbeit ist daher, eine ordnungsgemäße forensische Untersuchung von Android-Smartphones zu ermöglichen. Der Schwerpunkt wird hierbei auf die Datensammlung gelegt, da diese die essentielle Basis für die weiterführenden Untersuchungen bildet. Durch den praktischen Bezug soll ferner ein Nachschlagewerk zur Lösung gegenwärtiger Problemstellungen, besonders hinsichtlich der Datensammlung, geschaffen werden.

Das Ergebnis ist ein Leitfaden, der sich den mit Smartphones einhergehenden Veränderungen in der IT-Forensik widmet und eine systematische forensische Untersuchung der auf Android basierenden Geräte ermöglicht. Hierzu wurde unter Berücksichtigung von anerkannten Ansätzen ein Prozessmodell entwickelt, welches die essentielle Basis des Leitfadens darstellt. Darauf aufbauend erfolgte die Erarbeitung der aus Android resultierenden spezifischen forensischen Aspekte. Ein praktisches Beispielszenario demonstriert schließlich die Anwendbarkeit der Methoden aus der Datensammlung.

Abstract

Smartphones offer versatile applications due to powerful hardware, high storage capacities, an operating system with open interfaces and a permanent data connection. Consequently, smartphones have emerged as increasingly important devices in the field of computer forensics. Especially devices based on the operating system “Android” are widespread and accordingly hold an outstanding market position. In a criminal investigation, the data stored on the system can be used as potential evidence to investigate a crime.

Due to the special technical features of smartphones, the forensic investigation methods essentially differ from those of classical IT forensics. For instance, the approach to securing storage media (data collection) needs rethinking. A simple removal of the hard-wired flash memory, comparable to the removal of a hard drive from a computer, is not possible. Previous studies in the field of forensic investigation of Android smartphones either ignore the operating system-specific focus or do not follow a systematic approach based on approved process models. This might result in improper handling of the devices, leading to loss of essential evidence.

The objective of this master thesis is to provide a proper forensic investigation of Android smartphones. The focus is set on data collection since it forms the essential basis for all further steps of investigation. Through the practical approach, a reference work should be compiled to solve current problems – especially with respect to data collection.

The result is a guideline dedicated to smartphone related changes in IT forensics, allowing a systematic forensic investigation of Android-based smartphones. It builds on a fundamental process model that has been established while taking into account approved approaches and specific forensic aspects relating to Android. In conclusion, a practical scenario successfully demonstrates the applicability of the data collection methods.

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung dieser Masterarbeit unterstützt haben.

Mein ausdrücklicher Dank gilt meinen Kollegen bei der Controlware GmbH: Sebastian Racz, der mir als Betreuer dieser Masterarbeit stets mit Rat und Tat zur Seite stand. Nik Krüger und Tobias Stegmann, die als Korrekturleser mit ihrem Blick fürs Detail, wertvollen Anregungen sowie konstruktiver Kritik die Ausarbeitung bereichert haben.

Auch möchte ich Prof. Dr. Harald Baier und Prof. Dr. Ralf Hahn danken, die als betreuende Referenten fortwährend als Ansprechpartner zur Klärung theoretischer sowie praktischer Fragen zur Verfügung standen. Durch essentielle Ratschläge im Rahmen gemeinsamer Diskussionen und den notwendigen Freiraum, die Inhalte nach meinen Vorstellungen zu gestalten, haben Sie zur Entstehung dieser Masterarbeit maßgeblich beigetragen.

Ferner geht mein Dank an zwei sehr bedeutende Menschen: Martina Schorr-Löbig, die sich mit geschultem Auge meiner Ausarbeitung in einem ihr fachfremden Thema annahm, um mir wertvolle Korrekturvorschläge für den textlichen Feinschliff bereitzustellen. Mark Schorr, mein langjähriger Lebensgefährte, der mir während meines Studiums nicht nur Liebe und Rückhalt schenkte, sondern sehr viel Geduld aufbrachte, wenn ich mich in den Tiefen der Arbeit verlor.

Nicht zuletzt bin ich meinem treuen und wichtigsten Freund, Nils Rogmann, zutiefst für seine unermüdliche Unterstützung zu jeder Tages- und Nachtzeit dankbar. Er half mir, in der Masterarbeit mit konstruktiven sowie humorvollen Gesprächen den Weg durch das Labyrinth meiner Gedanken zu finden und knifflige technische Probleme zu lösen. Darüber hinaus verlieh mir seine fürsorgliche Art, gepaart mit seiner unverbesserlichen guten Laune, fortwährend essentielle Kraft und das entscheidende Selbstvertrauen.

Abschließend gilt mein besonderer Dank meiner Mutter Claudia Frühwein, der ich diese Masterarbeit zum Abschluss meines Studiums widmen möchte. Sie war in den vergangenen Jahren mein elementarer Ruhepol und zugleich eine wertvolle Beraterin. Ihre andauernde Liebe und Fürsorge in allen Lebenslagen ermöglichten es mir, dieses Studium erfolgreich zu meistern und dabei das Wichtigste – das Leben – nicht aus den Augen zu verlieren.

Inhaltsverzeichnis

Abkürzungsverzeichnis

ADB	Android Debug Bridge
ADM	Android Device Monitor
API	Application Programming Interface
APDU	Application Protocol Data Unit
APK	Android Application Package File
BSI	Bundesamt für Sicherheit in der Informationstechnik
CPU	Central Processing Unit
DDMS	Dalvik Debug Monitor Server
DVM	Dalvik Virtual Machine
Ext4	Fourth Extended Filesystem
FAT32	File Allocation Table 32
GB	Gigabyte
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IEEE	Institute of Electrical and Electronics Engineers
IMEI	International Mobile Equipment Identifier
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LiME	Linux Memory Extractor
LTE	Long Term Evolution

MB	Megabyte
MD	Message Digest
NFC	Near Field Communication
NIST	National Institute for Standards and Technology
NTP	Network Time Protocol
PID	Process Identifier
PIN	Personal Identification Number
PUK	Personal Unblocking Key
RAM	Random Access Memory
ROM	Read Only Memory
Rootfs	Root Filesystem
SCSI	Small Computer System Interface
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SQL	Structured Query Language
SSL	Secure Sockets Layer
TAP	Test Access Port
Tmpfs	Temporary Filesystem
UMTS	Universal Mobile Telecommunications System
URI	Uniform Ressource Identifier
USB	Universal Serial Bus

WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
YAFFS	Yet Another Flash File System

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Smartphones sind aus dem privaten und beruflichen Umfeld nicht mehr wegzudenken. Entsprechend der Hochrechnungen des Marktforschungsinstituts Gartner wurden im vierten Quartal 2012 weltweit 472 Millionen Mobiltelefone verkauft, wovon allein 207 Millionen Geräte Smartphones waren [Gar13]. Im Vergleich zu traditionellen Mobiltelefonen verfügen Smartphones über eine deutlich leistungsstärkere Hardware und höhere Speicherkapazitäten. Ein vollwertiges Betriebssystem mit offenen Schnittstellen ermöglicht die Erweiterung des Funktionsumfangs um zusätzliche Applikationen. Diese erlauben neben der Verwaltung von Kontakten, Terminen und E-Mails auch die Bearbeitung von Dokumenten oder die Navigation nach ausgewählten Routen. Mobile Datenverbindungen gewährleisten hierbei die permanente Synchronisation sämtlicher Daten. Diese vielseitigen Einsatzmöglichkeiten der Smartphones führen zu einer wachsenden Bedeutung der Geräte im Bereich der IT-Forensik. Die auf dem System befindlichen Daten dienen als potentielle Spuren, die im Zuge eines Ermittlungsverfahrens zur Aufklärung einer Straftat genutzt werden können. Aufgrund ihrer Stellung am Markt (Anteil von 69,9% im vierten Quartal 2012) sind hierbei vorrangig Geräte basierend auf dem Betriebssystem „Android“ von besonderer Wichtigkeit [Gar13].

1.1 Motivation

Die klassische IT-Forensik berücksichtigt die besonderen technischen Gegebenheiten von Smartphones nicht umfassend. Vor allem die Methoden zur Untersuchung der Geräte weichen wesentlich von den etablierten Ansätzen ab. Beispielhaft ist hier der Leitfaden „IT-Forensik“ des Bundesamtes für Sicherheit in der Informationstechnik (BSI) anzuführen [Bun11]. Bereits die Beschlagnahmung und Überführung der Geräte in ein forensisches Labor erfordern ein Umdenken. Während dies bei Computern in der Regel im ausgeschalteten Zustand erfolgt, werden Smartphones aufgrund ihrer Beschaffenheit sowie Portabilität bevorzugt im laufenden Betrieb transportiert. Auf diese Weise soll vor allem einem Datenverlust, der mit einer Abschaltung des Geräts einhergeht, entgegengewirkt werden. Bestehende Funkverbindungen bergen jedoch die Gefahr einer Veränderung von existierenden Daten wie Standortdaten und im ungünstigsten Fall einer Löschung des gesamten Systems durch einen „Remote Wipe“. Eine Abschirmung der Geräte ist daher zur Gewährleistung der Integrität wichtig.

Die anschließende Datensammlung, bei der Sicherungen der auf dem Smartphone gespeicherten Daten angefertigt werden, stellt eine weitere Herausforderung dar. Zur Erstellung eines vollständigen physikalischen 1:1-Abbilds analog zu einer ausgebauten Festplatte ist das Herauslöten des im Smartphone fest verdrahteten Flash-Speichers erforderlich. Dies führt nicht nur zur unwiderruflichen Zerstörung des Geräts, sondern gegebenenfalls, durch die beim Löten entstehenden hohen Temperaturen, zur Vernichtung des Speichers. Alternativ besteht die Möglichkeit, die bereitgestellten Betriebssystemschnittstellen von Android zur Datensammlung zu verwenden. Unter Umständen können hierbei jedoch aufgrund von herstellerspezifischen Schutzmaßnahmen nicht alle ausschlaggebenden Daten ohne die Manipulation des Geräts und somit Verletzung der Integrität bezogen werden.

Bisherige Ausarbeitungen zur forensischen Untersuchung von Smartphones – wie vom National Institute for Standards and Technology (NIST) [Nat07] oder von Alexander Geschonneck [Ges11] – versuchen zwar, diese Gegebenheiten zwar aufzugreifen, thematisieren jedoch nicht die spezifischen Problemstellungen von Android. Explizite Publikationen zur Android-Forensik wie von Andrew Hoog, Geschäftsführer des Unternehmens „viaForensics“, bearbeiten wesentliche Teilaspekte, allerdings fehlt hier eine prägnante und an anerkannte Prozessmodelle angelehnte Systematik zur Untersuchung [Hoo11]. Ferner erfordern die Veränderungen in Bezug auf die Architektur und Sicherheit von Android der letzten zwei Jahre eine weiterführende wissenschaftliche Bearbeitung. Eine forensische Untersuchung von auf Android basierenden Smartphones kann demnach bisher nicht angemessen durchgeführt werden. Die Folge ist unter Umständen eine falsche Handhabung, die zum Verlust von essentiellen Beweismitteln führen kann.

1.2 Zielsetzung

Im Zuge dieser Masterarbeit gilt es, sich den mit Smartphones einhergehenden Veränderungen in der IT-Forensik zu widmen. Die Aufgabe besteht darin, auf Grundlage einschlägiger Literatur eine Systematik zur forensischen Untersuchung von Smartphones basierend auf Android zu erarbeiten. Hierbei ist es erforderlich, den aktuellen Stand der Technik auf diesem Gebiet tiefgehend zu erforschen sowie neue, notwendige Methoden im forensischen Umgang mit den Geräten aufzuzeigen. Der wissenschaftliche Schwerpunkt der Arbeit liegt auf der Datensammlung.

Diese bildet die essentielle Basis für sämtliche weiterführenden forensischen Ermittlungen und ist daher vor allem im Hinblick auf die Gewährleistung der Integrität sowie die gerichtliche Verwertbarkeit der digitalen Spuren zu beleuchten. Aufgrund der Wichtigkeit der Datensammlung wird abschließend mit Hilfe eines praxisnahen Beispielszenarios die Anwendbarkeit der darin beschriebenen Ergebnisse demonstriert. Die zentralen Fragestellungen, die in der Masterarbeit beantwortet werden, lauten:

- Wie können Android-Smartphones systematisch einer forensischen Untersuchung unterzogen werden?
- Wie kann eine forensisch korrekte und umfassende Datensammlung erfolgen, welche die Integrität und die gerichtliche Verwertbarkeit der digitalen Spuren gewährleistet?
- Welche Herausforderungen ergeben sich bei der forensischen Untersuchung, im Speziellen bei der Datensammlung, und wie können diese ggf. bewältigt werden?
  - Wie können Daten des Android-Smartphones bezogen werden und welche Auswirkungen, besonders auf die Integrität und gerichtliche Verwertbarkeit, haben notwendige Manipulationen am Gerät zur Folge?
  - Wie können die Daten extrahiert und gesichert werden, wenn das Android-Smartphone durch verschiedene Schutzmaßnahmen, wie z. B. einen Sperrbildschirm oder eine Verschlüsselung, geschützt ist?
- Welche Daten sind auf Android-Smartphones von Interesse und können innerhalb einer Untersuchung ermittelt werden?
- Welche existierenden Werkzeuge sind für den Einsatz zur forensischen Untersuchung, im Speziellen zur Datensammlung, von Android-Smartphones geeignet?

Die Masterarbeit hat zum Ziel, eine systematische und somit ordnungsgemäße forensische Behandlung der aktuellen auf Android basierenden Smartphones, besonders im Zusammenhang mit der Datensammlung, zu ermöglichen. Demnach soll keine abstrakte Auflistung von Untersuchungsmethoden erfolgen, sondern vielmehr ein Leitfaden, vergleichbar mit dem Leitfaden „IT-Forensik“ des BSI, mit praxisnahen Handlungsanweisungen zur Verfügung gestellt werden. Ferner ist durch den praktischen Bezugs ein Nachschlagewerk zur Lösung gegenwärtiger Problemstellungen, vor allem in Bezug auf die Datensammlung, zu schaffen.

1.3 Aufbau der Arbeit

Die Masterarbeit untergliedert sich in insgesamt sieben Kapitel, beginnend mit der **Einleitung**, in der auf die Motivation und Zielsetzung eingegangen wird.

Im Kapitel **Grundlagen** wird in die zum Verständnis der Masterarbeit relevanten Themengebiete eingeführt. Hierbei erfolgen die Darstellung der forensisch relevanten Speichermedien eines Smartphones, die tiefgehende Beleuchtung von Android und die Beschreibung essentieller Aspekte hinsichtlich der IT-Forensik.

Für den Aufbau eines Leitfadens zur forensischen Untersuchung von Android-Smartphones werden im Kapitel **Prozessmodell** in der Fachwelt anerkannte Modelle auf dem Gebiet der IT-Forensik analysiert. Die Stärken und Schwächen der Ansätze bilden schließlich die Grundlage zur Etablierung eines auf Smartphones abgestimmten Prozessmodells, das den essentiellen Rahmen für die tiefgehenden Betrachtungen bezüglich Android bereitstellt.

Das allgemein auf Smartphones bezogene Prozessmodell wird im Kapitel **Android-Prozess** ausführlich um die Gegebenheiten von Android ergänzt. Hierbei erfolgt die Analyse des aktuellen Stands der Technik auf dem Gebiet und daraus abgeleitet die Beschreibung von Methoden, die einen systematischen forensischen Umgang mit den spezifischen Smartphones ermöglichen.

Die Datensammlung, d. h. die Anfertigung von Sicherungen der auf dem Smartphone gespeicherten Daten, bildet den Grundstein der forensischen Untersuchung. Deshalb wird innerhalb dieser Masterarbeit ein besonderes Augenmerk auf den Bereich gelegt und eine vertiefende Betrachtung essentieller Bestandteile im Kapitel **Datensammlung** durchgeführt.

Im Kapitel **Anwendbarkeit** wird aufgrund der Wichtigkeit der Datensammlung abschließend die Anwendung der zuvor erarbeiteten und damit in Verbindung stehenden Handlungsanweisungen demonstriert. Dies beinhaltet die Erstellung eines Beispielszenarios anhand dessen die beispielhafte Sicherung der auf einem Smartphone gespeicherten Daten erfolgt.

Abgeschlossen wird die Masterarbeit durch das Kapitel **Resümee**, in dem nochmals die Motivation und Zielsetzung aufgegriffen, das erreichte Ergebnis überprüft und ein Ausblick gegeben wird. Ein persönliches Fazit hinsichtlich der forensischen Untersuchung von Smartphones basierend auf Android rundet das zu Grunde liegende Thema schließlich ab.

2 Grundlagen

Die Grundlagen geben eine Einführung in die für diese Masterarbeit relevanten Themengebiete, welche für die forensische Untersuchung von Smartphones basierend auf Android benötigt werden. Einleitend erfolgt hierzu im Kapitel „Smartphones“ eine Beschreibung der aus forensischer Sicht zentralen Speichermedien der Geräte. Daran knüpft im Kapitel „Android“ eine tiefgehende Beleuchtung des in dieser Ausarbeitung fokussierten Betriebssystems an. Abschließend werden im Kapitel „IT-Forensik“ die elementaren Anforderungen, gängigen Praktiken und essentielle Aspekte hinsichtlich des Einsatzes von Werkzeugen im forensischen Umfeld betrachtet. Insgesamt sollen mit der Einführung fundamentale Fragen zur Architektur von Android-Smartphones und zur IT-Forensik geklärt werden.

2.1 Smartphones

Smartphones verfügen im Vergleich zu traditionellen Mobiltelefonen neben einem Betriebssystem mit offenen Schnittstellen, die eine Erweiterung des Funktionsumfangs erlauben, über eine deutlich leistungsstärkere Hardware. Im Bereich der IT-Forensik sind vor allem die elektronischen Speichermedien der Hardware relevant, da diese die für die forensische Untersuchung entscheidenden Daten enthalten. Darunter fallen der „Random Access Memory“ (RAM), der „Read Only Memory“ (ROM), der „Flash-Speicher“ und der Speicher der „Subscriber Identity Module“-Karte (SIM-Karte) eines Smartphones. Die in diesen Medien vorliegenden Daten dienen als potentielle Spuren, die innerhalb eines Ermittlungsverfahrens zur Aufklärung einer Straftat genutzt werden können. Um eine umfassende forensische Untersuchung und vor allem Datensammlung durchführen zu können, ist ein Verständnis der wesentlichen Eigenschaften und Strukturen der Speichermedien essentiell.

2.1.1 RAM-Speicher

Der Random Access Memory dient als Arbeitsspeicher im Smartphone, den das Betriebssystem und Applikationen zur Verarbeitung ihrer Daten verwenden. Android arbeitet, wie auch die Betriebssysteme von klassischen Computern, mit einer virtuellen Speicherverwaltung (vgl. [Bec11] S. 122). Demnach wird der physikalische Speicher, der in heutigen Geräten bereits 4 Gigabyte umfassen kann, nicht direkt allokiert, sondern über „Seitentabellen“ auf virtuelle Speicheradressen abgebildet.

Auf diese Weise kann ein virtuell zusammenhängender Speicherbereich bereitgestellt werden, der physikalisch über viele Seiten verteilt ist. Anders als in traditionellen Systemen, die eine virtuelle Speicherverwaltung verwenden, lagert Android keine Daten aus dem RAM in einer Datei auf den Flash-Speicher aus. Aufgrund der flüchtigen Eigenschaft des RAMs und der fehlenden Auslagerungsdatei gehen sämtliche zwischengespeicherten Daten nach Abschaltung der Stromzufuhr verloren. Ferner bewirkt die Interaktion mit dem Smartphone eine Veränderung des Speicherinhalts. Die Daten können wertvolle Rückschlüsse auf zuletzt ausgeführte Aktivitäten geben und relevante Informationen, wie z. B. Benutzernamen und Passwörter oder Schlüssel zur Entschlüsselung, enthalten (vgl. [Hoo11] S. 133). Um einen solchen Verlust zu vermeiden, ist der RAM, sofern das Smartphone im eingeschalteten Zustand vorgefunden wird, zu Beginn einer Datensammlung zu sichern.

2.1.2 ROM-Speicher

Der persistente Speicher Read Only Memory enthält ausschließlich den Code zum Initiieren des Bootvorgangs. Der Bootvorgang eines auf Android basierenden Smartphones untergliedert sich in insgesamt sieben Phasen (siehe Abbildung 1). Diese spielen im Hinblick auf den Einsatz forensischer Untersuchungsmethoden und die Datensammlung eine maßgebliche Rolle und werden daher nachstehend näher beleuchtet. Hierbei erfolgt jedoch nur die Darstellung eines vereinfachten Überblicks. Für eine tiefgehende technische Betrachtung sei auf die in diesem Kapitel verwendete Literatur verwiesen [XDI09].

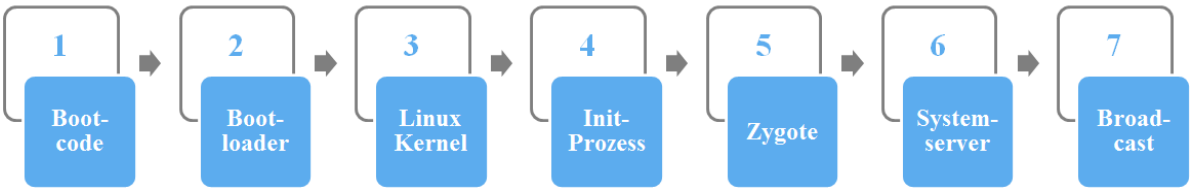

Abbildung 1: Bootvorgang eines Android-Smartphones

(1) Der Bootvorgang beginnt mit der Ausführung des im ROM befindlichen **Bootcodes**, der sich der Lokalisierung des Bootmediums widmet. Typischerweise wird als Bootmedium der interne Flash-Speicher genutzt. Kann dieser detektiert werden, lädt der ROM-Code den primären Bootloader von dem Flash in den internen RAM. Hierbei handelt es sich um einen RAM-Baustein, der vergleichbar zu einem Cache ein Bestandteil der CPU ist.

(2) Der **primäre Bootloader** verfügt über einen kleinen Programmteil, der ausschließlich für die Bereitstellung des externen RAMs zuständig ist. Mit Hilfe des externen RAMs kann der Hauptteil des Bootloaders, nachfolgend als „sekundärer Bootloader“ bezeichnet, geladen werden. Der **sekundäre Bootloader** bietet die Möglichkeit, einen alternativen Bootmodus auszuwählen. Standardmäßig verfügen Android-Smartphones über einen Wiederherstellungsmodus (engl. Recovery Mode). Der Recovery Mode dient zur Wartung des Geräts und ermöglicht unter anderem das Einspielen von Aktualisierungen oder die Wiederherstellung der Werkseinstellungen. Im Falle eines standardmäßigen Bootvorgangs wird, neben der Einrichtung der Dateisysteme, der auf Linux basierende Kernel des Smartphones in den externen RAM geladen.

(3) Infolgedessen übernimmt der **Linux Kernel**, welcher für die Verwaltung der Systemressourcen verantwortlich ist (siehe Kapitel 2.2.1 „Architektur“), die Kontrolle über das Smartphone. Das Gerät wird in diesem Rahmen für den laufenden Betrieb vorbereitet. Eine essentielle Aufgabe, die damit einhergeht, ist der Start des Prozesses „Init“.

(4) Der **Init-Prozess** bildet den Ursprung aller weiteren Prozesse auf dem System und liest das im Wurzelverzeichnis liegende Shell-Skript `init.rc` ein. Dadurch werden eine Vielzahl von Systemeinstellungen vorgenommen und elementare Prozesse des Betriebssystems etabliert. Erwähnenswert ist an dieser Stelle der Prozess „Zygote“.

(5) Bei **Zygote** handelt es sich um einen Prozess, der im Kern die virtuelle Maschine mit dem Namen „Dalvik Virtual Machine“, kurz DVM, initialisiert. Die DVM dient als Umgebung zur Ausführung einer Applikation unter Android (siehe Kapitel 2.2.1 „Architektur“). Für jede aufgerufene Applikation wird eine Kopie von Zygote und somit der DVM erzeugt.

(6) Der **Systemserver** stellt die erste Applikation dar, welche auf dem zuvor beschriebenen Weg gestartet wird. Sie initialisiert die grundlegenden Systemdienste (z. B. für die Hintergrundbeleuchtung, Bluetooth oder die im Gerät verbauten Sensoren) und stellt damit die wesentlichen Kernfunktionalitäten des Smartphones zur Verfügung.

(7) Zur Beendigung des Bootvorgangs wird abschließend die Nachricht `ACTION_BOOT_COMPLETED` per **Broadcast** an das System ausgesendet. Dadurch wird die Betriebsbereitschaft des Smartphones signalisiert.

2.1.3 Flash-Speicher

Das Smartphone nutzt zur dauerhaften Datenspeicherung, d. h. auch nach Abschaltung der Stromzufuhr, Flash-Speicher mit einer Kapazität von bis zu 64 Gigabyte. Fokussiert werden hierbei Modelle, die zur Ansteuerung der Speicherzellen einen integrierten Controller mit einem „Flash Translation Layer“ bereitstellen [WGP+08]. Die zusätzliche Schicht emuliert oberhalb der Flash-Hardware ein „Block Device“. Dadurch können die Speicherzellen des Flash-Speichers analog zu einer Festplatte über logische Blockadressen angesprochen werden. Dies erlaubt den Gebrauch von Dateisystemen auf dem Flash-Speicher, die sich im Bereich der Computer etabliert haben und eine solche Form der Adressierung voraussetzen.

Aufgrund der persistenten Eigenschaft und der hohen Speicherkapazitäten stellt der Flash-Speicher die wichtigste Komponente innerhalb einer forensischen Untersuchung im IT-Umfeld dar. Generell verfügen Smartphones über einen fest auf der Leiterplatine des Geräts verdrahteten Speicher. Darauf befinden sich, analog zu einer Festplatte, Systemdaten (z. B. der Bootloader und das Betriebssystem), Applikationsdaten (z. B. Installationsdateien) sowie beliebige Benutzerdaten (z. B. Bilder, Videos und Dokumente). Einige Smartphones erlauben ferner – vornehmlich zur Speicherung von Benutzerdaten – die Ergänzung eines entfernbaren Flash-Speichers, der sich unter dem Akku oder seitlich im Gehäuse befindet.

Partitionierung

Der interne Flash-Speicher unter Android untergliedert sich im Gegensatz zum externen Flash-Speicher, der typischerweise aus einer Partition besteht, in mehrere Partitionen. Die Kenntnis der Partitionierung ist für eine gezielte Datensammlung unerlässlich. Die genaue Unterteilung ist primär vom Hersteller des Smartphones abhängig. Generell sind auf Android basierenden Geräten die in Abbildung 2 dargestellten Partitionen mit den nachfolgend aufgeführten Dateisystemen (blau) und Einhängepunkten im System (grün) vorzufinden (vgl. [Hoo11] S. 132-134 und [VZC11] S. 16-17):

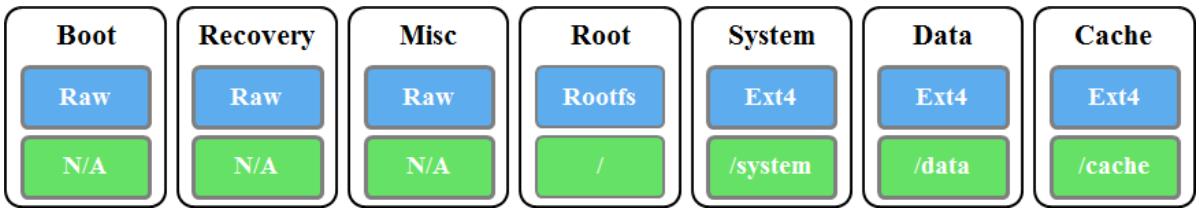

Abbildung 2: Typische Partitionierung des internen Flash-Speichers unter Android

Die Partition **Boot** ist für den herkömmlichen Bootvorgang des Smartphones entscheidend und enthält unter anderem den hierfür benötigten Kernel. Ohne ihre Existenz kann kein Systemstart erfolgen. Die Daten für den alternativen Bootvorgang im Recovery Mode werden unter Android in **Recovery** bereitgestellt. Beide aufgeführten Partitionen abstrahieren die Daten nicht über ein Dateisystem, sondern erlauben einen direkten Zugriff darauf. **Misc** umfasst verschiedene Hardware-Einstellungen, die während des Bootvorgangs geladen werden und für den Betrieb des Smartphones zwingend erforderlich sind. Der Zugriff auf diese Daten erfolgt ebenfalls direkt, ohne die Verwendung eines Dateisystems. Den Ausgangspunkt aller weiteren Partitionen stellt **Root** dar. Ferner werden darüber die für den Systemstart notwendigen Initialisierungsskripte, wie z. B. `init.rc`, bereitgestellt. Das Betriebssystem, mit Ausnahme des Kernels, befindet sich in **System**. Durch das Löschen der Partition wird lediglich das Betriebssystem entfernt. Ein Start des Smartphones im Recovery Mode und somit das Installieren eines neuen Betriebssystems ist weiterhin möglich. Die Daten der auf dem Smartphone installierten Applikationen werden unter **Data** abgelegt. Darunter fallen neben Installationsdateien auch benutzerspezifische Einstellungen sowie Inhalte der Applikationen (z. B. angelegte Kontakte und Termine). Für Benutzerdaten, wie Bilder, Videos und Dokumente, existiert für gewöhnlich eine gesonderte Partition auf dem internen Speicher, deren Einhängepunkt vom Hersteller des Smartphones abhängt. **Cache** wird schließlich vom Betriebssystem und manchen Applikationen zur Zwischenspeicherung beliebiger Daten im laufenden Betrieb verwendet.

Dateisysteme

Für die verschiedenen Partitionen des internen Flash-Speichers kommt eine Vielzahl von Dateisystemen zum Einsatz. Den Ursprung bildet hierbei das „Root Filesystem“ (Rootfs) der Partition **Root**, welches im Wurzelverzeichnis eingehängt wird (vgl. [Hoo11] S. 132-133). Die darunter liegenden Dateisysteme, welche aus forensischer Sicht relevante Daten enthalten, basieren seit Android 2.3 auf dem in Linux-Umgebungen verbreiteten „Fourth Extended Filesystem“ (Ext4). Dazu gehören neben **System** vor allem die Partitionen **Data** und **Cache**. In den vorherigen Betriebssystemversionen diente das sogenannte „Yet Another Flash File System“ (YAFFS) als Standard. Aufgrund der fehlenden Unterstützung von mehreren Prozessorkernen und Multithreading wurde allerdings ein Dateisystemwechsel vollzogen [Hei10].

Auf den externen Flash-Speichern hingegen wird aus Gründen der Kompatibilität zu den meisten auf dem Markt erhältlichen Betriebssystemen für gewöhnlich „File Allocation Table 32“ (FAT32) eingesetzt (vgl. [Ges11] S. 280). Ermöglicht wurde die Verwendung dieser traditionellen Dateisysteme vornehmlich durch die Implementierung des Flash Translation Layers. Daneben ist noch das „Temporary Filesystem“ (Tmpfs) anzuführen. Das Dateisystem nutzt zur Speicherung der Daten den virtuellen Speicher des RAMs (vgl. [Hoo11] S. 137). Demnach gehen sämtliche darin vorliegenden Daten nach Abschaltung der Stromzufuhr verloren. In einer forensischen Untersuchung bietet sich Tmpfs z. B. zur Ablage von Werkzeugen, die keine Installation erfordern, oder zur Zwischenspeicherung kleinerer Sicherungen an. Auf diese Weise können unnötige Veränderungen an den Inhalten des Flash-Speichers vermieden werden. Typischerweise wird das Dateisystem von dem Verzeichnis `/dev` genutzt (vgl. [Hoo11] S. 280).

2.1.4 SIM-Karte

Beim Subscriber Identity Module handelt es sich um eine Chipkarte, die für gewöhnlich in eine Vorrichtung unter dem Akku oder seitlich am Gehäuse des Smartphones eingesteckt wird. Die SIM-Karte verfügt über die notwendigen Informationen zur Identifikation und Authentifizierung eines Benutzers im Mobilfunknetz (vgl. [Nat07] S. 11). Nach einer erfolgreichen Authentifizierung ist der Benutzer beispielsweise in der Lage, Telefonate zu führen, Kurzmitteilungen über den Nachrichtendienst „Short Message Service“ zu versenden und das mobile Internet zu nutzen. Zum Schutz vor einer unberechtigten Nutzung der Funktionalitäten ist die SIM-Karte mit einer veränderbaren vier- bis achtstelligen „Personal Identification Number“ (PIN) geschützt. Diese wird, sofern der Schutz aktiviert ist, beim Einschalten des Smartphones abgefragt. Nach einer dreimaligen Falscheingabe kann ein Zugriff auf die SIM-Karte nur noch durch Verwendung eines „Personal Unblocking Keys“ (PUK) erfolgen, der beim Mobilfunkanbieter hinterlegt ist. Eine permanente Sperrung tritt schließlich nach insgesamt zehn fehlerhaften Eingabeversuchen des PUKs ein.

Dateisystem

Eine aus forensischer Sicht entscheidende Komponente der SIM-Karte ist der intern verbaute persistente Speicher mit einer Größe von 16 bis 512 Kilobyte. Das Dateisystem des Speichers ist als hierarchische Baumstruktur realisiert.

Der Baum setzt sich, wie in Abbildung 3 dargestellt, aus einem `Master File` sowie einer beliebigen Anzahl von `Dedicated Files` und `Elementary Files` zusammen (vgl. [Nat07] S. 43-44):

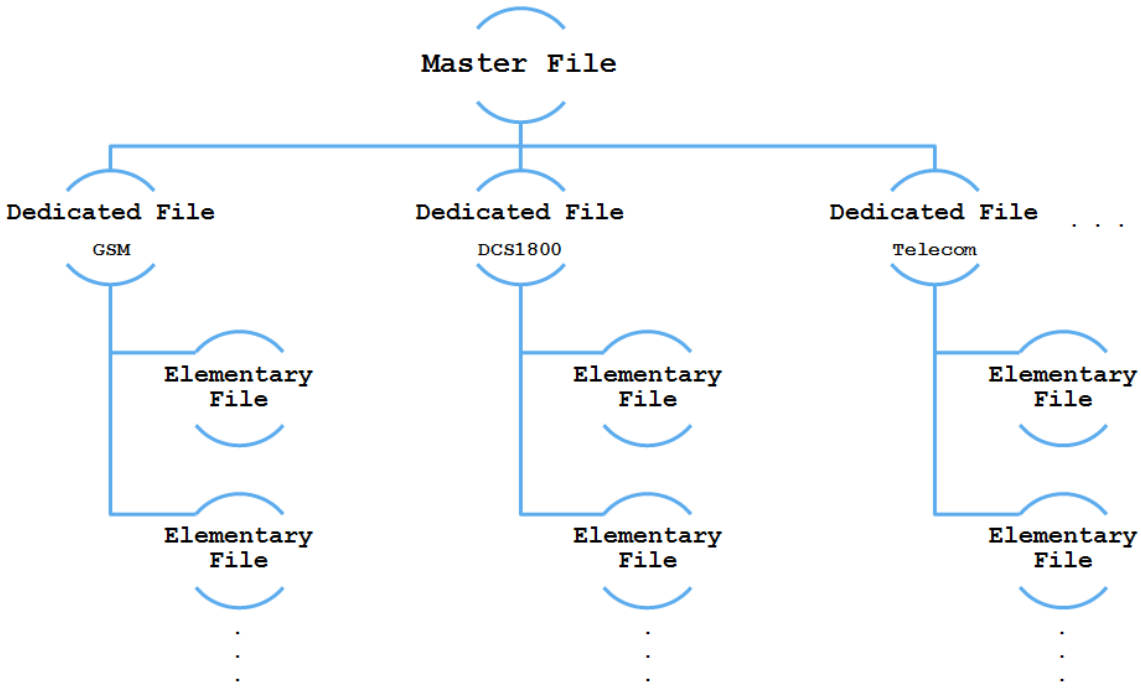

Abbildung 3: Dateisystem des persistenten Speichers der SIM-Karte

Das `Master File` stellt, analog zum Wurzelverzeichnis von Linux, den Ursprung des hierarchischen Baums und folglich des Dateisystems dar. Gemäß dieser Definition existiert nur ein `Master File`, das ein oder mehrere `Dedicated Files` und darunter einzelne `Elementary Files` umfasst. `Dedicated Files` sind hierbei vergleichbar mit Verzeichnissen und weisen in ihrer Struktur lediglich einen Header auf (vgl. [SG07] S. 182). Der Header enthält Informationen über das zu Grunde liegende Verzeichnis im Dateisystem (z. B. den verfügbaren Speicherplatz, die Anzahl der `Elementary Files` und die gegebenen Zugriffsrechte darauf). `Elementary Files` sind einzelne Dateien, die sowohl einen Header als auch einen Body beinhalten (vgl. [SG07] S. 182).

Im Body befinden sich die für eine forensische Untersuchung relevanten Daten. Dabei handelt es sich, wie beispielhaft in Abbildung 3 aufgeführt, unter `Dedicated Files GSM` sowie `DSC1800` um netzwerkbezogene Daten für verschiedene Frequenzbänder und unter `Dedicated Files Telecom` um servicebezogene Daten (vgl. [Nat07] S. 44).

Aus den zugehörigen `Elementary Files` können z. B. gespeicherte Kontakte, gesendete, empfangene sowie entworfene Kurzmitteilungen, vorhandene Anruflisten und Standortdaten gewonnen werden.

2.2 Android

Android stellt mit einem Marktanteil von 69,9% im vierten Quartal 2012 die weltweit meist verbreitete Plattform für mobile Endgeräte dar (siehe Abbildung 4). Hierzu zählen neben den in dieser Ausarbeitung fokussierten Smartphones auch Geräte wie Spielekonsolen, Tablets, Netbooks, Digitalkameras oder E-Book-Reader.

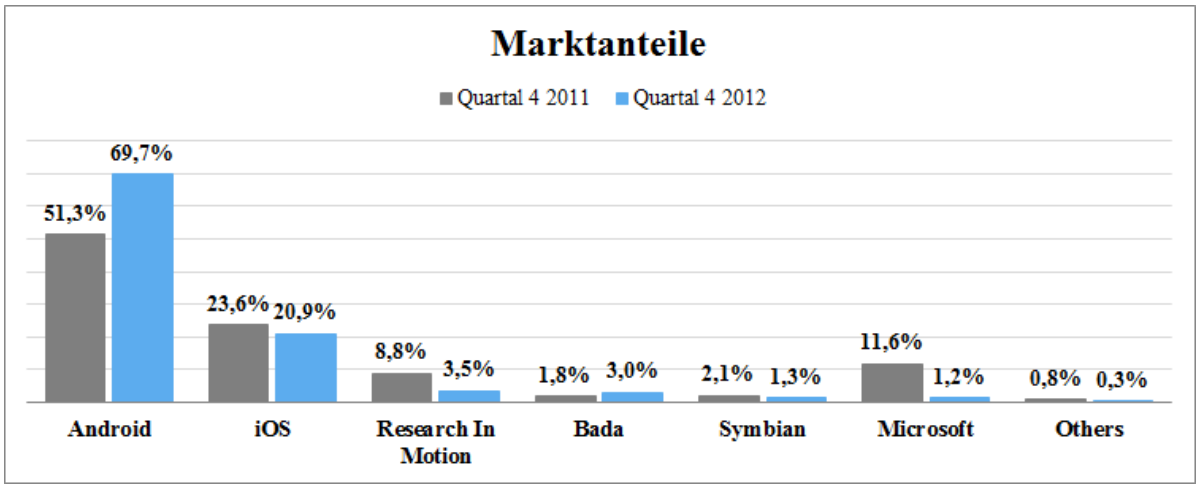

Abbildung 4: Marktanteile der Betriebssysteme [Gar13]

Ursprünglich wurde das Betriebssystem von der Android Inc., ein im Jahr 2003 maßgeblich durch den Mitgründer Andrew E. Rubin etabliertes Unternehmen, eingeführt [Blo13]. Google kaufte schließlich im Jahr 2005 Android und setzte die Entwicklung in seiner eigens initiierten Open Handset Alliance fort [Blo05]. Die Open Handset Alliance ist ein Konsortium bestehend aus insgesamt 84 Hardware- und Software-Unternehmen zur Entwicklung eines offenen Standards für mobile Endgeräte [Ope13].

Seit der Übernahme durch Google wird Android als Open Source-Software unter der freien Software Lizenz der Apache Software Foundation bereitgestellt. Diese Offenheit fördert maßgeblich die Bildung einer großen Community, die erheblich zur Weiterentwicklung und zum Erfolg des Betriebssystems beiträgt.

Aktuell ist Android in der Version 4.2 mit dem Namen „Jelly Bean“ verfügbar. Die seit Version 2.3 „Gingerbread“ durchgeführten Aktualisierungen haben unter anderem zu architektonischen und sicherheitstechnischen Veränderungen geführt.

2.2.1 Architektur

Die grundlegende Architektur des Betriebssystems untergliedert sich in die „Applikationsschicht, den „Applikationsrahmen“, verschiedene „Bibliotheken“, die „Android Laufzeitumgebung“ und den „Linux Kernel“ (siehe Abbildung 5). Die aufgeführten Ebenen sind für den laufenden Betrieb des Smartphones erforderlich und werden nachfolgend beschrieben.

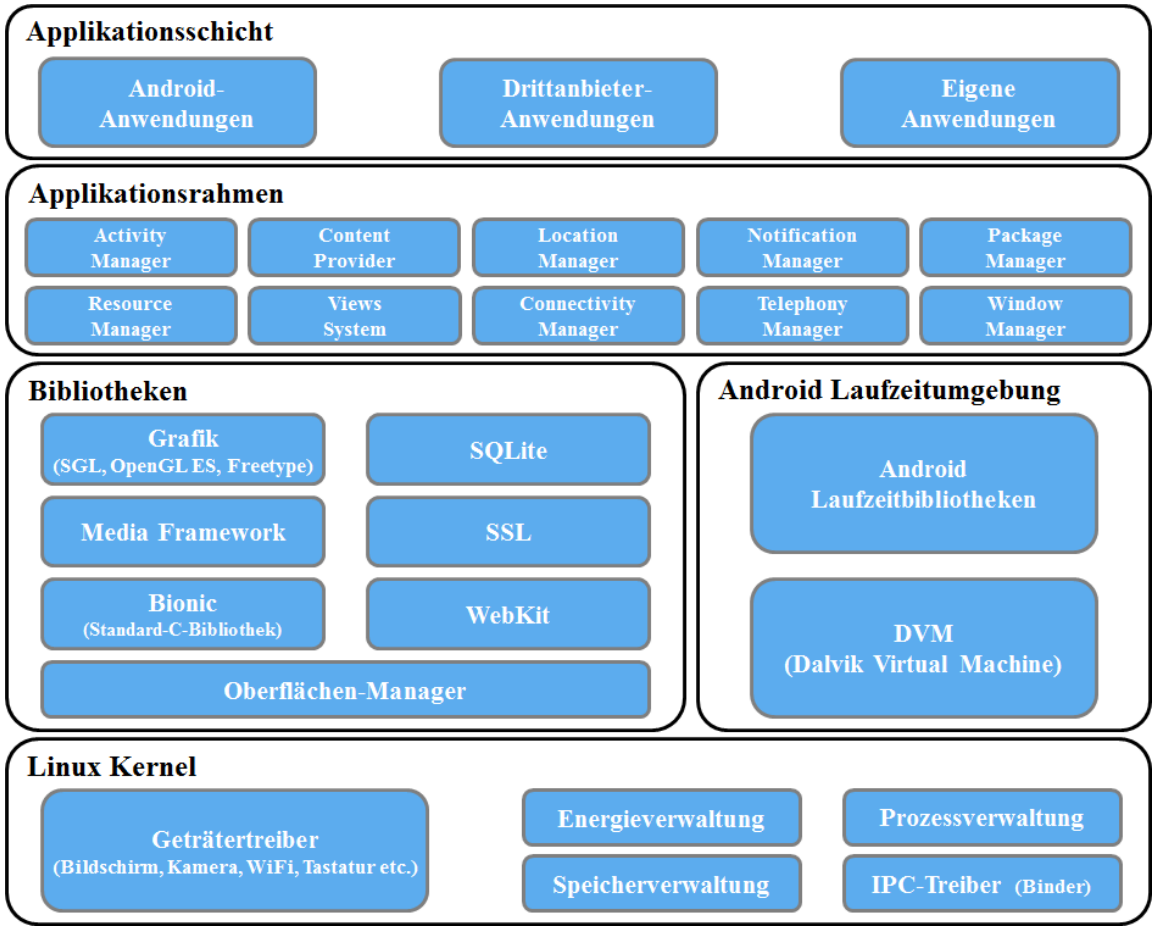

Abbildung 5: Betriebssystemarchitektur von Android (vgl. [BP10] S. 19)

Linux Kernel

Den Kern des Betriebssystems bildet ein angepasster Linux Kernel, welcher in Jelly Bean in der Version 3.4 implementiert ist [Lar13].

Dieser dient als Abstraktionsschicht zwischen der Hardware und den darüber liegenden Ebenen. Demzufolge stellt der Kernel die notwendigen Gerätetreiber zur Verfügung und ist unter anderem für die Energie-, Speicher- und Prozessverwaltung zuständig (vgl. [BP10] S. 19). Die durchgeführten Anpassungen sollen den Kernel für die Eigenheiten mobiler Hardware optimieren. Infolgedessen wurde beispielsweise der Energieverbrauch reduziert, indem die Hintergrundbeleuchtung von Tastatur und Bildschirm sowohl manuell als auch automatisiert ein- und ausgeschaltet werden kann.

Android Laufzeitumgebung

Die Laufzeitumgebung von Android besteht im Wesentlichen aus verschiedenen Laufzeitbibliotheken, welche die Kernfunktionalitäten von Java bereitstellen, und der bereits in Kapitel 2.1.2 angeführten Dalvik Virtual Machine. Die DVM wurde von Google entworfen, um eine effiziente und sichere Umgebung zur Ausführung von mobilen Applikationen zu schaffen (vgl. [Hoo11] S. 87).

In der Regel werden Applikationen unter Android in Java mit dem Java Software Development Kit geschrieben und kompiliert. Durch Einsatz des Cross-Compilers `dx`, der Bestandteil des Android Software Development Kits ist, kann aus dem kompilierten Java-Bytecode der spezielle Dalvik-Bytecode erzeugt werden (vgl. [BP10] S. 21). Dieser ermöglicht aufgrund seines vergleichsweise geringeren Speicherbedarfs eine effiziente Speicherung und ist für die Ausführung innerhalb der DVM optimiert (vgl. [BP10] S. 21 und [Hoo11] S. 87). Die einzelnen Dalvik Bytecode-Dateien können zu einer vollendeten Applikation in einem Installationspaket mit der Endung `.apk` zusammengeführt werden. Darin sind ferner ein „Android-Manifest“ mit essentiellen Informationen über die Applikation sowie weitere Ressourcen wie Bilder und Sounds enthalten (vgl. [Bec11] S. 123).

Zur Gewährleistung der Sicherheit startet jede Applikation einen eigenen Betriebssystemprozess (vgl. [BP10] S. 23). Die gestarteten Betriebssystemprozesse stellen Kopien des im Bootvorgang vorgestellten Zygote-Prozesses dar, in denen jeweils eine eigene Instanz der DVM läuft. Die DVM ist hierbei so optimiert, dass mehrere Instanzen gleichzeitig auf einem Android-Smartphone betrieben werden können (vgl. [BP10] S. 23). Durch diese Technik verfügt jede Applikation über ihren eigenen gesonderten Speicherbereich.

Ferner trägt dies zur Systemstabilität bei, da ein abgebrochener Betriebssystemprozess nur die Beendigung einer Applikation zur Folge hat.

Bibliotheken

Die Kernfunktionalitäten von Android sind in C/C++-Bibliotheken enthalten, welche vom Applikationsrahmen und letztlich der Applikationsschicht genutzt werden. Auch hier wurde auf eine Optimierung des Energie- und Speicherverbrauchs geachtet. Demnach kommt als C-Bibliothek eine vom Berkeley Software Distribution-Standard abgeleitete Variante mit dem Namen „Bionic“ zum Einsatz. Diese enthält einige spezifische Erweiterungen für Android und ist kleiner als der für Linux entwickelte GNU-Standard (vgl. [Bec11] S. 122-123). Weiterhin existieren Bibliotheken wie „WebKit“ zum Rendern von Webseiten, „SSL“ für sichere Kommunikationsverbindungen oder „Media Framework“ zur Darstellung von Multimediaformaten (vgl. [BP10] S. 23 und [Hoo11] S. 86). Ebenso erwähnenswert sind die „SQLite“-Bibliotheken und das darüber bereitgestellte quelloffene sowie relationale Datenbanksystem. Dieses wird von Applikationen zur strukturierten Speicherung beliebiger Daten genutzt und stellt folglich einen besonders wichtigen Bereich im Zuge einer forensischen Untersuchung dar (vgl. [Hoo11] S. 86).

Applikationsrahmen

Der in Java entwickelte Applikationsrahmen bildet die Basis für sämtliche Applikationen. Mit Hilfe dieser Schicht werden die zu Grunde liegende Hardware und das Betriebssystem des Smartphones in Form von beliebigen Manager-Klassen abstrahiert (vgl. [BP10] S. 24). Die Manager-Klassen erlauben schließlich den Applikationen über bereitgestellte „Application Programming Interfaces“, kurz APIs, den Zugriff auf die Funktionalitäten dieser Ressourcen.

Applikationsschicht

Auf der obersten Ebene der Architektur befinden sich von Google mitgelieferte, von Drittherstellern angebotene oder selbstentwickelte Applikationen (vgl. [BP10] S. 20). Durch die verbreitete Community sind seit April 2013 bereits mehr als 850.000 Applikationen im Umlauf, die eine umfassende Erweiterung des Funktionsumfangs eines Smartphones erlauben [Sta13]. Eine wesentliche Bezugsquelle stellt hierbei das Portal „Google Play“ dar.

Google Play ist in Form einer Applikation auf von Google lizensierten Android-Smartphones und im Internet unter *https://play.google.com/* verfügbar. Für die Nutzung ist die Einrichtung eines Benutzerkontos bei Google erforderlich.

Eine Applikation kann sowohl auf dem internen als auch auf dem externen Flash-Speicher installiert werden. Hierbei variiert lediglich der Speicherort der `.apk`-Dateien [And13a]. Auf dem internen Speicher befinden sich die `.apk`-Dateien von bereits durch das Betriebssystem bereitgestellten Applikationen unter `/system/app` und von nachträglich installierten Applikationen unter `/data/app`.

Die durch die Applikation gespeicherten Daten können in einer forensischen Untersuchung von besonderer Bedeutung sein. Üblicherweise verfügt jede Applikation zur Datenablage über ein dediziertes Verzeichnis, benannt nach dem Namen des zugehörigen Packages, unter `/data/data` im Dateisystem des internen Flash-Speichers (vgl. [BP10] S. 259 und [Hoo11] S. 96). Zur Strukturierung existieren darin weitere Unterverzeichnisse, die gleiche Dateitypen zusammenfassen. Demzufolge befinden sich beispielsweise Datenbanken in `./databases`, Bibliotheken in `./lib`, Cachedateien in `./caches`, beliebige Dateien in `./files` und Applikationseinstellungen in `./shared_prefs` (siehe Abbildung 6):

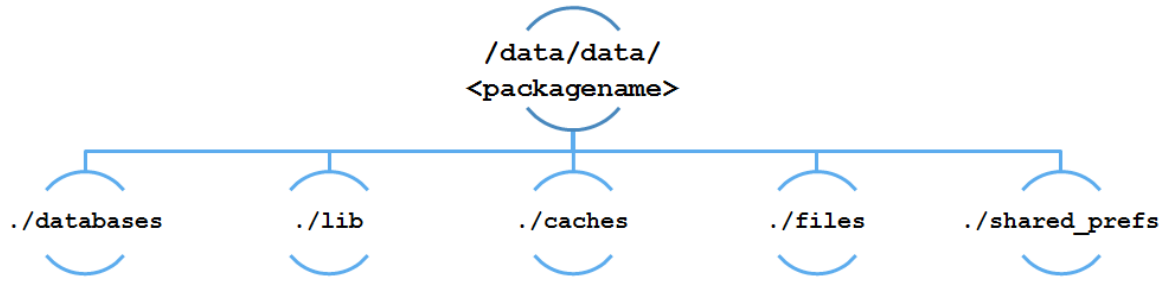

Abbildung 6: Verzeichnisstruktur einer Applikation

Alternativ können Applikationen zur Entlastung des internen Speichermediums die von ihnen erzeugten Daten auch direkt auf einen verfügbaren externen Flash-Speicher sichern (vgl. [Hoo11] S. 96). Ein Beispiel hierfür wären Bilder oder Videos, die mit dem Smartphone aufgenommen wurden.

2.2.2 Sicherheit

Android implementiert verschiedene Schutzmaßnahmen, die während einer forensischen Untersuchung nicht zu vernachlässigen sind. Sie dienen unter anderem dem Schutz der Daten vor unberechtigten Zugriffen und verhindern somit im ungünstigsten Fall die erfolgreiche Durchführung einer Datensammlung. Zu den für diese Ausarbeitung elementaren Maßnahmen gehören das Prinzip der „Sandbox“, die Anwendung von Berechtigungen sowie Verschlüsselungen und die Einrichtung eines Sperrbildschirms.

Sandbox

Innerhalb der Architektur wurde bereits gezeigt, dass Applikationen in einer eigenen Umgebung ausgeführt werden und über einen eigenen Bereich im Dateisystem des internen Flash-Speichers verfügen. Dieser Ansatz wird in der Literatur als „Sandbox“ bezeichnet, d. h. eine dedizierte, von anderen Ressourcen isolierte Laufzeitumgebung für Applikationen (vgl. [BP10] S. 27). Android nutzt zur Umsetzung der Sandbox das Prozessmanagement und das Berechtigungssystem von Linux (vgl. [BP10] S. 27 und [Hoo11] S. 89). Jede Applikation erhält hierbei während der Installation einen gesonderten Benutzer auf Betriebssystemebene, unter welchem sie in einem zugehörigen Prozess anschließend ausgeführt wird. Der Prozess, in dem jeweils eine eigene Instanz der DVM läuft, ist in Linux gegen einen Zugriff von außen geschützt. Das Berechtigungssystem erlaubt ausschließlich dem bei der Installation angelegten Benutzer einen lesenden und schreibenden Zugriff auf die entsprechenden Ressourcen. Ein Problem stellen in diesem Zusammenhang externe Flash-Speicher dar, die aus Kompatibilitätsgründen häufig FAT32 als Dateisystem verwenden und demnach keine Benutzerrechte unterstützen (vgl. [Kle11] S.29). Sämtliche Benutzer und somit Applikationen, die Zugang zum externen Flash-Speicher besitzen, können folglich uneingeschränkt lesend und schreibend auf die darauf befindlichen Daten zugreifen.

Berechtigungen

Um einen Zugriff auf Ressourcen eines anderen Benutzers, d. h. außerhalb der Sandbox, zu erlauben, sind zusätzliche Berechtigungen erforderlich (vgl. [BP10] S. 29 und [Hoo11] S. 89). Diese sind im Android-Manifest unter dem Tag `uses-permission` festgehalten und werden zum Zeitpunkt der Installation angefordert. Die erteilten Rechte, welche ggf. explizit durch den Gerätebesitzer zu bestätigen sind, bleiben schließlich dauerhaft bestehen.

Oftmals gehen Besitzer unachtsam mit diesen Freigaben um, wodurch die Sicherheit der auf dem Gerät befindlichen Daten stark beeinträchtigt wird. Dies ist auch dem Umstand geschuldet, dass keine selektive Bestätigung von einzelnen Rechten möglich ist (vgl. [Kle11] S. 33). Demzufolge müssen entweder alle Freigaben zugelassen oder die Installation der Applikation abgebrochen werden.

Zum Austausch von Daten zwischen Applikationen ist der im Applikationsrahmen enthaltene „Content Provider“ vorgesehen (siehe Abbildung 5 auf Seite 13). Der Content Provider, welcher über einen „Uniform Ressource Identifier“ (URI), vergleichbar zu einer Website, eindeutig adressierbar ist, kann auf Anfrage ausgewählte Daten anderen Applikationen zur Verfügung stellen [And13b]. Die anfragenden Applikationen müssen hierzu über die vom Provider geforderten Berechtigungen, welche ebenfalls im Android-Manifest unter dem Tag `provider` definiert sind, verfügen. Der Content Provider bietet einen entscheidenden Mechanismus, der auch in forensischen Werkzeugen zur Datensammlung eingesetzt wird.

Alternativ zum Content Provider können mehrere Applikationen in einer gemeinsam genutzten Sandbox ausgeführt werden. Hierzu wird den entsprechenden Applikationen im Android-Manifest eine gemeinsame `SharedUserID` vergeben. Die gleiche Kennung erlaubt die Zuweisung des gleichen Benutzers auf Betriebssystemebene und somit des gleichen Prozesses bei Ausführung (vgl. [BP10] S. 31). Im Rahmen des Berechtigungssystems gelten die einzelnen Applikationen als eine Komponente, wodurch folglich ein Zugriff auf die gleichen Ressourcen, d. h. Funktionalitäten und Daten, erfolgen kann. Voraussetzung für diese Methode ist, dass die Applikationen mit dem gleichen Zertifikat signiert sind.

Verschlüsselung

Auf dem externen Flash-Speicher installierte Applikationen lagern lediglich ihre `.apk`-Dateien aus, welche in Folge einer fehlenden Unterstützung von Benutzerrechten frei zugänglich wären. Damit die Applikationen trotzdem vor unberechtigten Zugriffen geschützt sind, werden die `.apk`-Dateien in einem verschlüsselten Container abgelegt (vgl. [And13a] und [Kle11] S. 29-30). Beim Start einer solchen Applikation kann die Datei durch Android in das Dateisystem eingebunden und entschlüsselt werden. Der Besitzer des Smartphones hat ferner die Möglichkeit, die Daten des externen und mit Jelly Bean auch die Data-Partition des internen Flash-Speichers über das Betriebssystem zu verschlüsseln.

Auf diese Weise sind die Daten des Geräts vor unberechtigten Zugriffen geschützt und nur nach Eingabe eines Sicherheitscodes beim Systemstart zugänglich. Der Sicherheitscode wird über die Aktivierung eines Sperrbildschirms in Form einer PIN oder eines Passworts gesetzt und dient zur Ver- und Entschlüsselung des eigentlichen Schlüssels [And13c]. Sämtliche generierten Schlüssel befinden sich auf dem Smartphone, wodurch eine Verwendung der jeweiligen Applikationen und Daten in anderen Geräten unterbunden wird [And13c]. Eine forensische Untersuchung kann durch die Ergreifung einer solchen Schutzmaßnahme maßgeblich beeinträchtigt und im ungünstigsten Fall vollkommen verhindert werden.

Sperrbildschirm

Android bietet die Möglichkeit, in den Einstellungen unter der Rubrik „Sicherheit“ einen Sperrbildschirm einzurichten. Die darauf folgende Aktivierung des Bildschirms erfolgt zum einen nach dem Systemstart und zum anderen manuell oder automatisch nach Ablauf einer vom Besitzer vorgegebenen Zeitspanne. Ein aktivierter Sperrbildschirm erfordert die Eingabe einer PIN, eines Passworts oder eines zu zeichnenden Musters. Das Setzen einer PIN oder eines Passworts ist für eine Verschlüsselung des internen sowie externen Flash-Speichers notwendig. In Jelly Bean ist ferner die Verwendung einer Gesichtserkennung möglich. Ausschließlich nach erfolgreicher Authentifizierung kann eine Interaktion mit dem eingeschalteten Smartphone erfolgen. Die Schutzmaßnahme stellt somit, besonders im Hinblick auf die stets automatische Aktivierung, innerhalb einer forensischen Untersuchung ein enormes Hindernis dar.

2.2.3 Werkzeuge

Android bietet verschiedene Werkzeuge, die nicht nur in der Entwicklung, sondern auch während der forensischen Untersuchung von Smartphones eingesetzt werden. Von Bedeutung sind zum einen das Software Development Kit, kurz SDK, für Android und zum anderen die „Entwickleroptionen“ auf dem Gerät.

Software Development Kit

Das SDK dient zur Entwicklung von Applikationen basierend auf Android. Die darin enthaltenen Werkzeuge eignen sich auch für den forensischen Einsatz. Anzuführen sind hier die „Android Debug Bridge“ (ADB) und der „Android Device Monitor“ (ADM).

Bei der Android Debug Bridge handelt es sich um ein kommandozeilenbasiertes Werkzeug, das eine Kommunikation mit dem Smartphone über eine Workstation ermöglicht [And13d]. Auf dem Smartphone wird hierzu ein Daemon im Hintergrund ausgeführt. Die Workstation besitzt einen Client, welcher durch Eingabe des Befehls `adb` in einer geöffneten Shell gestartet wird, sowie einen Server, welcher die Verbindung zwischen dem Client und dem Daemon verwaltet. Für die Verbindung wird ein gesonderter Benutzer mit eingeschränkten Rechten verwendet, wodurch nicht alle Daten auf dem Gerät eingesehen werden können. Die vollständige Befehlsreferenz der ADB samt inhaltlicher Beschreibung und beispielhafter Anwendung ist unter [And13d] zu finden. Zu den wichtigsten Befehlen gehören:

- `adb devices` zur Auflistung der angeschlossenen Smartphones
- `adb start-server` zum Starten sowie `adb kill-server` zum Beenden des Servers
- `adb shell` zum Öffnen einer Shell auf dem Smartphone
- `adb forward tcp:<port> tcp:<port>` zur Einrichtung einer Portweiterleitung
- `adb push <local> <remote>` zur Übertragung von Daten der forensischen Workstation auf das Smartphone
- `adb pull <remote> [<local>]` zur Übertragung von Daten des Smartphones auf die forensische Workstation

Der Android Device Monitor bietet eine grafische Oberfläche zum Debuggen selbstentwickelter Applikationen [And13e]. Ein wesentlicher Bestandteil des Monitors ist der „Dalvik Debug Monitor Server“ (DDMS), welcher die Android Debug Bridge für die Kommunikation nutzt. DDMS stellt einen „File Explorer“ bereit, der eine komfortable Möglichkeit bietet, auf die Dateisysteme eines Smartphones zuzugreifen und darin abgelegte Daten auf der Workstation zu sichern. Ferner können mit Hilfe von DDMS Logdaten des Betriebssystems und der installierten Applikationen gesichtet werden.

Entwickleroptionen

Das Betriebssystem des Smartphones bietet in den Einstellungen diverse Entwickleroptionen an, die unter anderem zum Debuggen und Überwachen des Geräts benötigt werden. Seit der Version 4.2 sind diese Optionen aus Sicherheitsgründen standardmäßig ausgeblendet.

Zum Anzeigen ist es erforderlich, in den Einstellungen unter der Rubrik „Geräteinformationen“ das Feld „Buildnummer“ sieben Mal zu betätigen [And13f]. Eine in der Forensik wichtige Option ist das „USB-Debugging“. Dieser Modus erlaubt nach Einschaltung einen umfassenden Zugriff auf das Smartphone über eine daran angeschlossene Workstation. Dadurch können beliebige Daten ausgetauscht, Applikationen ohne Benachrichtigung installiert und Protokolldaten ausgelesen werden. Ferner ist die Aktivierung des USB-Debuggings notwendig, um eine Verbindung über die Android Debug Bridge mit dem Smartphone herzustellen und z. B. eine Datensammlung durchzuführen. Ein aktivierter Sperrbildschirm des Geräts, der die Eingabe eines Passworts oder ein zu zeichnendes Musters erfordert, kann die Aktivierung unter Umständen verhindern und damit eine forensische Untersuchung maßgeblich einschränken.

2.3 IT-Forensik

Die IT-Forensik ist ein Teilgebiet der Forensik und widmet sich dem Nachweis sowie der Aufklärung strafbarer Handlungen durch die Sicherung und Bewertung von digitalen Spuren auf IT-Systemen (vgl. [Ges11] S. 2). Diese können schließlich als potentielle Beweismittel in einem Gerichtsverfahren genutzt werden. Die Ermittlung der digitalen Spuren soll entsprechend der eingangs aufgeführten Definition die Beantwortung der nachfolgenden kriminalistischen Fragestellungen ermöglichen (vgl. [Bun11] S. 22):

- Was ist geschehen?
- Wo ist es geschehen?
- Wann ist es geschehen?
- Wie ist es geschehen?
- Wer hat es getan?
- Was kann gegen eine Wiederholung getan werden?

Die letzten beiden Fragestellung sind vor allem im Hinblick auf eine Strafverfolgung oder Sicherheitsbewertung von besonderer Relevanz (vgl. [Bun11] S. 22). Prinzipiell existiert im gesamten Untersuchungsverlauf das Gebot, sowohl digitale Spuren zu ermitteln, die einen Tatbestand belegen, als auch solche, die ihn widerlegen können.

2.3.1 Anforderungen

Im Zuge einer Strafverfolgung stellt die Gewährleistung der gerichtlichen Verwertbarkeit der digitalen Spuren eine wesentliche Aufgabe dar. Der grundlegende Prozess zur forensischen Untersuchung von IT-Systemen unterliegt zur Einhaltung dieses Grundsatzes verschiedenen Anforderungen. Diese sehen im Kern wie folgt aus (vgl. [Bun11] S. 23 und [Ges11] S. 66-67):

- **Akzeptanz:** Die angewandten Untersuchungsschritte und Methoden müssen in der Fachwelt dokumentiert und anerkannt sein. Bei Verwendung neuer Untersuchungsschritte und Methoden muss deren Korrektheit nachgewiesen werden.
- **Glaubwürdigkeit:** Die Glaubwürdigkeit setzt die Robustheit und Funktionalität der angewandten Methoden sowie bei Bedarf den Nachweis dieser Aspekte voraus. Diese Anforderung ist besonders entscheidend bei komplexen Methoden, deren Verfahrens- und Wirkungsweise nur schwer nachzuvollziehen ist.
- **Wiederholbarkeit:** Eine erneute Durchführung der forensischen Untersuchung unter Verwendung der gleichen Ausgangsmaterialien, Untersuchungsschritte und Methoden muss dieselben Ergebnisse erzielen.
- **Integrität:** Im gesamten Untersuchungsverlauf muss sichergestellt werden, dass die erhobenen Daten und somit digitalen Spuren unverändert sind. Demnach dürfen weder bewusst noch unbewusst Manipulationen vorgenommen werden. Darüber hinaus besteht die Pflicht, jederzeit nachweisen zu können, dass die Integrität erhalten bleibt.
- **Authentizität:** Die Authentizität beschreibt die Echtheit der durchgeführten Untersuchungsschritte und erhobenen digitalen Spuren. Diese muss ebenfalls während des gesamten Untersuchungsverlaufs gewährleistet werden.
- **Ursache und Auswirkung:** Die angewandten Untersuchungsschritte und Methoden sollen es ermöglichen, logisch nachvollziehbare Verbindungen zwischen Ereignissen, Spuren und evtl. auch Personen herzustellen.
- **Dokumentation:** Im Rahmen der forensischen Untersuchung müssen alle durchgeführten Untersuchungsschritte, verwendeten Werkzeuge, veränderten und gesicherten Daten sowie gewonnenen Spuren dokumentiert werden. Darunter fällt auch die Anfertigung einer „Kontrollkette“ (engl. Chain of Custody).

> Die Chain of Custody erfordert die lückenlose Dokumentation „des Verbleibs der Beweismittel und deren Einsichtnahme“ (siehe [Bun11] S. 90). Dadurch soll fortwährend nachvollziehbar sein, welche Person zu welchem Zeitpunkt auf die Beweismittel zugegriffen hat und welche Arbeiten daran ausgeführt wurden. Dies soll insgesamt die Anfechtbarkeit der Beweiskraft von erhobenen Beweismitteln, z. B. in einem Gerichtsverfahren, reduzieren.

Die Einhaltung der aufgeführten Anforderungen stellt in einer forensischen Untersuchung von Smartphones eine Herausforderung dar. Die Sicherung des RAMs und des internen Flash-Speichers erfolgt in der Regel im laufenden Betrieb und erfordert unter Umständen die Manipulation des Geräts (vgl. [Hoo11] S. 197-198 und [Nat07] S. 45). Daraus ergeben sich im ungünstigsten Fall Veränderungen an bestehenden Daten und folglich digitalen Spuren, die zu einer Minderung des Beweiswertes führen. Neben der Verletzung der Integrität hat dieser Sachverhalt auch Auswirkungen auf die Wiederholbarkeit. Durch die Modifikation der Ausgangsmaterialien können aufeinanderfolgende Sicherungen der Daten unterschiedliche Ergebnisse erzielen (vgl. [Nat07] S. 46).

Bereits innerhalb der Ausarbeitung „IT-Forensik im Wandel – Die Aufweichung des Paradigmas der Unveränderbarkeit am Beispiel von Smartphones mit dem Windows Phone Betriebssystem“ wurde das Problem hinsichtlich der Integrität der Daten beleuchtet [RB12]. Demzufolge sind bei der forensischen Untersuchung von Smartphones drei Grundregeln zu beachten: Die Veränderungen müssen **(1)** gerechtfertigt, **(2)** so gering wie möglich und **(3)** genauestens protokolliert sein (vgl. [RB12] S. 8). Eine vergleichbare Herangehensweise empfiehlt auch der Experte auf dem Gebiet der IT-Forensik Eoghan Casey in seinem Buch „Digital Evidence and Computer Crime“ (vgl. [Cas11] S. 20). Besonders aus den ersten beiden Grundregeln lässt sich allgemein ableiten, dass die Interaktion mit dem Smartphone auf ein Mindestmaß zu beschränken ist.

2.3.2 Praktiken

Die Sicherung der zu Grunde liegenden Daten bildet die Basis für die weiterführenden Untersuchungsschritte und hat somit eine herausragende Bedeutung für den gesamten Ermittlungsverlauf sowie die gerichtliche Verwertbarkeit der digitalen Spuren.

Zur Durchführung einer korrekten Datensammlung haben sich in der klassischen IT-Forensik, die sich der forensischen Untersuchung von IT-Systemen wie Computern widmet, essentielle Praktiken etabliert. Diese Praktiken sind auch in der Smartphone-Forensik relevant und entsprechend auf das Umfeld zu übertragen.

Reihenfolge

Die Reihenfolge, in der eine Sicherung durchzuführen ist, hängt maßgeblich von der Flüchtigkeit der Daten ab. Gemäß Alexander Geschonneck kann zwischen „flüchtigen", „temporär zugreifbaren" und „fragilen" Daten unterschieden werden (siehe [Ges11] S. 69). Flüchtige Daten befinden sich vornehmlich im RAM und gehen bei Unterbrechung der Stromversorgung verloren. Temporär zugreifbare Daten sind zwar dauerhaft auf einem persistenten Speichermedium verfügbar, jedoch kann nur zu bestimmten Zeitpunkten darauf zugegriffen werden. Ein Beispiel hierfür wäre eine verschlüsselter Container, dessen Daten nur zugänglich sind, wenn dieser unverschlüsselt eingebunden ist. Fragile Daten werden im Smartphone-Umfeld typischerweise auf den bereits eingeführten Flash-Speichern gesichert und verändern ihren Zustand lediglich durch einen Zugriff darauf. Entsprechend dieser Definition sind zur Vorbeugung eines Verlusts zu Beginn die flüchtigen, darauf folgend die temporär zugreifbaren und zuletzt die fragilen Daten zu sichern (vgl. [Bun11] S. 34).

Vollständigkeit

Während der Datensammlung kann eine bitweise vollständige oder bitweise partielle Sicherung durchgeführt werden (vgl. [Nat07] S. 13).

Die Anfertigung einer vollständigen physikalischen 1:1-Abbildung des Speichermediums schließt die Sicherung von nicht allokierten Speicherbereichen mit ein. Diese können sich innerhalb oder außerhalb eines Dateisystems befinden. Beispielsweise entfernt das Dateisystem in der Regel bei der Löschung einer Datei nur den Verweis auf die jeweiligen Daten und markiert die zugehörigen Speicherbereiche als frei verfügbar bzw. nicht allokiert. Infolgedessen bleiben die eigentlichen Daten weiterhin erhalten. Lediglich werden diese nicht mehr im Dateisystem angezeigt. Eine fehlende Allokation impliziert somit nicht, dass die Speicherbereiche zwingend leer sind. Demnach können sich darin Daten von bereits gelöschten oder absichtlich versteckten Dateien befinden, die in einer forensischen Untersuchung von Bedeutung sind.

Eine partielle Sicherung umfasst wiederum physikalische Speicherbereiche (z. B. eine Partition) oder logische Objekte (z. B. Dateien oder Verzeichnisse) eines Speichermediums. Nicht allokierte Speicherbereiche müssen folglich explizit ermittelt und gesichert werden. Aufgrund dieser Einschränkung ist in einer forensischen Untersuchung nach Möglichkeit die vollständige der partiellen Variante vorzuziehen.

Arbeitskopien

Die notwendigen Schritte zur Durchführung der Datensammlung untergliedern sich im Wesentlichen in vier übergeordnete Aufgaben (siehe Abbildung 7):

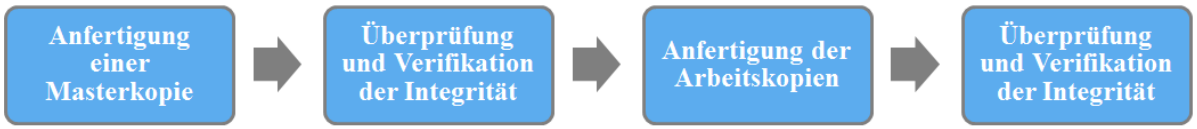

Abbildung 7: Durchführung der Datensammlung

Sämtliche Untersuchungsschritte sind zum Schutz der Originaldaten auf Arbeitskopien durchzuführen (vgl. [Bun11] S. 26 und [Nat07] S. 45). Die Arbeitskopien erlauben nicht nur eine mehrfache Ausführung verschiedener Untersuchungsschritte ohne die Veränderung der Originaldaten, sondern lassen darüber hinaus eine parallele Bearbeitung des Vorfalls durch mehrere Personen zu. Eine Masterkopie dient hierbei als Grundlage für alle weiteren Kopien. Dadurch soll der Kontakt mit dem Original und die Wahrscheinlichkeit einer unbeabsichtigten Manipulation auf ein Minimum reduziert werden. Zur Gewährleistung der Integrität ist ein kryptografischer Hash (z. B. mit Hilfe von MD5 oder SHA-1) sowohl über die Originaldaten als auch die erstellten Kopien zu berechnen (vgl. [Bun11] S. 27 und [Nat07] S. 45). Die Übereinstimmung der Hashwerte belegt, dass keine Veränderungen vorgenommen wurden.

Schreibschutz

Insbesondere während der Sicherung der Daten empfiehlt sich, sofern die Methoden dies erlauben, der Einsatz eines Schreibschutzes. Dadurch soll ein ausschließlich lesender Zugriff auf die Daten und die Einhaltung der Integrität garantiert werden (vgl. [Bun11] S. 26-27 und [Nat07] S. 22). Typischerweise werden für diesen Zweck spezielle „Writeblocker“ eingesetzt. Das BSI rät vor allem zur Verwendung von hardwarebasierten Systemen.

Diese können direkt zwischen forensischer Workstation und dem Speichermedium, auf dem die zu untersuchenden Daten liegen, geschaltet werden (vgl. [Bun11] S. 26). Hardwarebasierte Writeblocker werden von auf dem Gebiet der IT-Forensik etablierten Herstellern wie „WiebeTech“ und „Tableau“ für USB-, SCSI-, Firewire- oder IDE-Schnittstellen entwickelt.

2.3.3 Werkzeuge

Eine forensische Untersuchung erfordert den Einsatz verschiedener hardware- und softwarebasierter Werkzeuge. Diese können sowohl speziell auf die IT-Forensik als auch auf andere Einsatzgebiete (wie z. B. die Softwareentwicklung) ausgerichtet sein. Die Selektion der passenden Werkzeuge für eine Untersuchung wird maßgeblich durch das zu Grunde liegende IT-System und dessen Eigenschaften bestimmt. Vor allem im Zuge der Datensammlung sind forensische Werkzeuge zu präferieren, da diese in der Regel die Einhaltung der Integrität anstreben (vgl. [Nat07] S. 15). Sind diese für einen gegebenen Anwendungsfall nicht verfügbar, so kann auf Alternativen aus anderen Einsatzgebieten zurückgegriffen werden. Darüber hinaus definiert das NIST zusätzlich zu den in Kapitel 2.3.1 genannten Anforderungen an den Untersuchungsprozess verschiedene Kriterien, die bei der Werkzeugauswahl berücksichtigt werden sollten (vgl. [Nat07] S. 41-42):

- **Benutzerbarkeit:** Die Benutzerbarkeit adressiert die Fähigkeit des Werkzeugs, Daten in einer für den Ermittler zweckdienlichen Art und Weise zu präsentieren.
- **Reichhaltigkeit:** Das Werkzeug sollte eine vollständige Übersicht aller bedeutsamen Daten liefern, sodass sowohl belastende als auch entlastende Spuren identifiziert werden können.
- **Zuverlässigkeit:** Die Ausgaben des Werkzeugs müssen hinreichend überprüft worden sein, sodass potentielle Fehler, die eine Manipulation von digitalen Spuren zur Folge hätten, ausgeschlossen werden können.
- **Überprüfbarkeit:** Die durch das Werkzeug generierten Ergebnisse müssen nachvollziehbar sein, sodass deren Richtigkeit überprüft und sichergestellt werden kann.
- **Determinismus:** Die wiederholte Anwendung des Werkzeugs muss unter Einsatz der gleichen Ausgangsmaterialien und Instruktionen dieselben Ergebnisse erzielen. Die durch das Werkzeug generierten Ereignisse sind somit eindeutig bestimmt.

Letztlich beeinflussen auch die angebotenen Funktionalitäten, die Akzeptanz in der Fachwelt und die anfallenden Kosten maßgeblich die Auswahl. Generell sind die bereitgestellten Funktionalitäten eines Werkzeugs, besonders wenn diese für andere als forensische Zwecke entwickelt wurden, in einer Testumgebung zu untersuchen (vgl. [Nat07] S. 15). Auf diese Weise soll die Verfahrens- und Wirkungsweise verstanden und überprüft werden. Das NIST rät darüber hinaus, die Werkzeuge zur Vermeidung möglicher Konflikte in voneinander getrennten Umgebungen zu installieren (vgl. [Nat07] S. 22).

Im Kapitel „Grundlagen“ erfolgte eingangs eine Beschreibung der zentralen Speichermedien eines Smartphones. Hierzu zählen der RAM, der ROM, die Flash-Speicher und der Speicher der SIM-Karte. Die Hardware-Komponenten sind aus forensischer Sicht von besonderer Relevanz, da diese die Daten und folglich potentiellen Spuren enthalten.

Daran knüpfte eine Betrachtung des mit den Speichermedien in Verbindung stehenden und in dieser Ausarbeitung fokussierten Betriebssystems „Android“ an. Die Architektur des Betriebssystems untergliedert sich in die „Applikationsschicht, den „Applikationsrahmen“, verschiedene „Bibliotheken“, die „Android Laufzeitumgebung“ und den „Linux Kernel“. Durch das Prinzip der Sandbox, die Anwendung von Berechtigungen sowie Verschlüsselungen und die Einrichtung eines Sperrbildschirms wird der Schutz der vom Betriebssystem verwalteten Daten sichergestellt. Aus forensischer Sicht sind die Android Debug Bridge und der Android Device Monitor aus dem SDK von Android und die „Entwickleroptionen“ auf dem Gerät wichtige Untersuchungswerkzeuge.

Abgerundet wurden die Grundlagen durch eine Einführung in die IT-Forensik. Hierbei kristallisierte sich heraus, dass die Integrität der Daten eine maßgebliche Anforderung und zugleich Herausforderung im Untersuchungsprozess darstellt. Bei der Durchführung der Datensammlung spielen die Reihenfolge, die Vollständigkeit der Sicherung und die Etablierung eines Schreibschutzes eine bedeutende Rolle. Letztlich sind Werkzeuge zur forensischen Untersuchung mit Bedacht auszuwählen und vor dem praktischen Einsatz in realen Vorfällen in einer Testumgebung zu testen.

3 Prozessmodell

Der Aufbau eines Leitfadens zur forensischen Untersuchung von Smartphones basierend auf Android erfordert zu Beginn die Etablierung eines allgemeinen Prozessmodells. Hierzu werden in der Fachwelt anerkannte Ansätze analysiert. Zur fundierten Vergleichbarkeit gilt es dabei, die jeweiligen Kernaspekte herauszuarbeiten und mit einer einheitlichen Terminologie zu beschreiben. Darauf aufbauend erfolgt die Durchführung einer inhaltlichen Bewertung, in der die Stärken und Schwächen der Vorgehensweisen beleuchtet werden. Auf Basis der Erkenntnisse wird ein auf Smartphones abgestimmtes Modell ermittelt, das den essentiellen Rahmen für die tiefgehenden Betrachtungen bezüglich Android bildet.

3.1 Modellanalyse

Im Bereich der IT-Forensik existieren neben klassischen Prozessmodellen auch bereits allgemein auf Smartphones spezialisierte Ansätze.

Die klassischen Modelle beschreiben das Vorgehen zur Durchführung einer forensischen Untersuchung von traditionellen Computern. Ein anerkannter Repräsentant auf diesem Gebiet ist der Leitfaden „IT-Forensik“ vom Bundesamt für Sicherheit in der Informationstechnik [Bun11]. Zwar sind die darin beschriebenen Methoden, wie in der Einleitung dargestellt, nicht weitreichend genug, um den forensischen Gegebenheiten im Umgang mit Smartphones gerecht zu werden. Allerdings bietet die darin aufgezeigte allgemeine Vorgehensweise einen guten Themeneinstieg und stellt eine wertvolle Grundlage zur Entwicklung eines angepassten Modells dar.

Die spezialisierten Ansätze widmen sich dem Vorgehen zur forensischen Untersuchung von Smartphones. Anzuführen ist hier der etablierte Leitfaden „Guidelines on Cell Phone Forensics“ vom National Institute for Standards and Technology [Nat07] und das in der Fachwelt anerkannte Ablaufschema von Alexander Geschonneck aus dem Buch „Computer-Forensik – Systemeinbrüche erkennen, ermitteln, aufklären“ [Ges11]. Die Ausarbeitungen fokussieren die Besonderheiten von Smartphones in der IT-Forensik, welche unabhängig von der zu Grunde liegenden Betriebssystemarchitektur (z. B. Android) von Bedeutung sind. Durch deren teilweise unterschiedliche Sichtweisen wird eine objektivere und zugleich umfassende Betrachtung der aktuell angewandten Praktiken ermöglicht.

Eine ebenso erwähnenswerte Ausarbeitung stellt das Paper „Smartphone Forensic Investigation Process Model“ [GTA12] dar. Dieses ist im Vergleich zu den vorgenannten Veröffentlichungen zwar weniger anerkannt, jedoch werden auch darin wichtige forensische Aspekte in Bezug auf Smartphones beleuchtet, die in der Bewertung Berücksichtigung finden.

3.1.1 Modell „BSI“

Der Leitfaden „IT-Forensik“ des Bundesamts für Sicherheit in der Informationstechnik wurde im Jahr 2011 veröffentlicht. Mit Hilfe des Leitfadens werden die wesentlichen Grundlagen zur Durchführung einer forensischen Untersuchung von IT-Systemen bereitgestellt und praxisnahe Problemstellungen auf diesem Gebiet beleuchtet. Die darin beschriebene Vorgehensweise richtet sich vorwiegend an Administratoren von IT-Systemen und untergliedert sich in die insgesamt sechs übergeordneten Abschnitte „Strategische Vorbereitung“, „Operationale Vorbereitung“, „Datensammlung“, „Datenuntersuchung“, „Datenanalyse“ und „Dokumentation“ (vgl. [Bun11] S. 60). Jeder Abschnitt gruppiert hierbei einzelne, logisch zusammengehörige Untersuchungsschritte. Mit Hilfe dieser Unterteilung soll eine Strukturierung des Ablaufs vorgenommen werden, der schließlich die Erfassung eines zeitlichen Verlaufs innerhalb einer forensischen Untersuchung zulässt (siehe Abbildung 8):

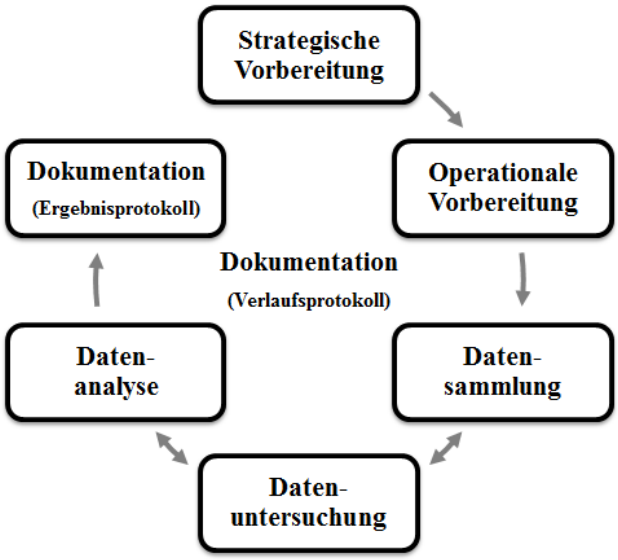

Abbildung 8: Forensisches Prozessmodell des BSI (vgl. [Bun11] S. 61 und S. 86)

Nachfolgend werden gemäß des Leitfadens die Kernaspekte der sechs Abschnitte des Modells, beginnend mit der strategischen Vorbereitung, vorgestellt (vgl. [Bun11] S. 87-91).

Strategische Vorbereitung

Die strategische Vorbereitung beinhaltet sämtliche Untersuchungsschritte, die vor Eintreten eines Vorfalls durch den Administrator der IT-Systeme zu ergreifen sind. Dadurch soll eine nachfolgende forensische Untersuchung unterstützt werden. Übergeordnet wird in diesem Zusammenhang die Einrichtung einer Workstation und des potentiell zu untersuchenden IT-Systems unter forensischen Gesichtspunkten fokussiert. Hinsichtlich des IT-Systems führt das BSI die Sicherstellung einer korrekten Systemzeit und die Bereitstellung eines Protokollierungsdienstes an. Ferner kann vorsorglich für die nachfolgenden Prozessabschnitte ein erweiterbarer Katalog mit klassischen forensischen Werkzeugen angefertigt werden.

Operationale Vorbereitung

Nach Eintritt eines Vorfalls folgt die operationale Vorbereitung, in der die Fragen „Was kann gesichert werden?“ und „Was soll gesichert werden?“ zu beleuchten sind.

Mit einer Bestandsaufnahme gilt es, potentielle Quellen zu identifizieren und zu beschlagnahmen, die für eine Sicherung in Frage kommen. Weiterhin sind hierbei all jene Quellen zu berücksichtigen, die innerhalb der strategischen Vorbereitung eingerichtet wurden.

Der aus dem Vorfall abgeleitete Anfangsverdacht gibt schließlich vor, welche Datenarten der identifizierten Quellen zur Sicherung herangezogen werden sollten. Übergeordnet können dies flüchtige, fragile und temporär zugreifbare Daten sein.

Datensammlung

Von den zuvor ausgewählten Datenarten ist im Abschnitt der Datensammlung eine Sicherung anzufertigen. Im Mittelpunkt steht hierbei die Anfertigung einer vollständigen physikalischen 1:1-Abbildung der persistenten Speichermedien. Alternativ können auch logische Objekte, wie Verzeichnisse oder Dateien eines Mediums, gesichert werden. Anschließend muss die Integrität der erhobenen Daten überprüft werden. Die Wahl der in diesem Abschnitt benötigten Werkzeuge kann anhand eines zu Beginn erstellten Katalogs erfolgen.

Ferner beginnt mit diesem Prozessabschnitt die Pflicht, eine Chain of Custody anzulegen und aufrecht zu erhalten. Diese setzt die lückenlose Dokumentation „des Verbleibs der Beweismittel und deren Einsichtnahme im Rahmen der nachfolgenden Untersuchungsschritte“ voraus (siehe [Bun11] S. 90).

Datenuntersuchung

Die anknüpfende Datenuntersuchung hat zum Ziel, für den Vorfall relevante Spuren aus den Sicherungen zu extrahieren und bei Bedarf in ein anderes Format zu überführen. Eine Formatänderung ergibt sich beispielsweise durch das Entpacken eines gewonnen Archivs. Eine Auswahl der hierfür vorgesehenen Werkzeuge kann auch hier unter Verwendung eines aus der strategischen Vorbereitung bereitgestellten Katalogs getroffen werden.

Während der Datenuntersuchung besteht die Möglichkeit, neue Datenquellen zu identifizieren, die eine erneute Datensammlung erfordern (symbolisiert in Abbildung 8 durch einen Rückpfeil auf den vorherigen Prozessabschnitt).

Datenanalyse

Die extrahierten potentiellen Spuren müssen zuletzt einer tiefgehenden Analyse unterzogen werden. Hierbei sollen mit Hilfe von ausgewählten Werkzeugen die Daten durch eine geeignete Korrelation in einen logischen Zusammenhang zueinander gebracht und in einen einheitlichen Zeitverlauf zusammengeführt werden. Ferner umfasst die Analyse eine inhaltliche Bewertung der Daten und deren Bezug zum gegebenen Vorfall. Eventuell kann es in diesem Baustein erforderlich sein, aufgrund fehlender Informationen eine weitere Datenuntersuchung und im Zuge dessen eine erneute Datensammlung durchzuführen (symbolisiert in Abbildung 8 durch einen Rückpfeil auf den vorherigen Prozessabschnitt).

Dokumentation

Der Prozessabschnitt zur Dokumentation unterscheidet gemäß dem BSI zwischen der Anfertigung eines Verlaufs- und eines Ergebnisprotokolls:

Die Protokollierung des Verlaufs ist prozessbegleitend durchzuführen und enthält zum einen die Beschreibung des Vorfalls sowie des daraus abgeleiteten Anfangsverdachts, der den Rahmen der Untersuchung vorgibt. Zum anderen sind darin alle durchgeführten Untersuchungsschritte und Methoden, verwendeten Werkzeuge, gesicherten Daten sowie gewonnen Spuren und Ergebnisse der einzelnen Prozessabschnitte aufgeführt. Ferner sollte das Protokoll auch Unvollständigkeiten und Verfälschungen, die besonders während der Datensammlung entstehen können, schriftlich festhalten und begründen.

Aus dem Verlaufsprotokoll wird abschließend ein Ergebnisprotokoll für eine vorgegebene Zielgruppe formuliert. Dieses soll schließlich einen Einblick in die durchgeführte forensische Untersuchung geben.

3.1.2 Modell „NIST“

Das National Institute for Standards and Technology veröffentlichte im Jahr 2007 den Leitfaden „Guidelines on Cell Phone Forensics“. Dieser widmet sich explizit der forensischen Untersuchung von Smartphones und informiert über die aktuellen Vorgehensweisen auf diesem Gebiet. Dabei wird das Ziel verfolgt, bestehende Forensik-Leitfäden zu ergänzen sowie aktuelle Herausforderungen in Bezug auf die forensische Untersuchung der Geräte zu vertiefen (vgl. [Nat07] S. 3). Aus den im Leitfaden übergeordneten Abschnitten „Beschlagnahmung“, „Erhebung“, „Untersuchung“, „Analyse“ sowie „Dokumentation“ (vgl. [Nat07] S. 29-67) lassen sich die nachfolgend aufgeführten zusammengefassten Untersuchungsschritte ableiten (siehe Abbildung 9):

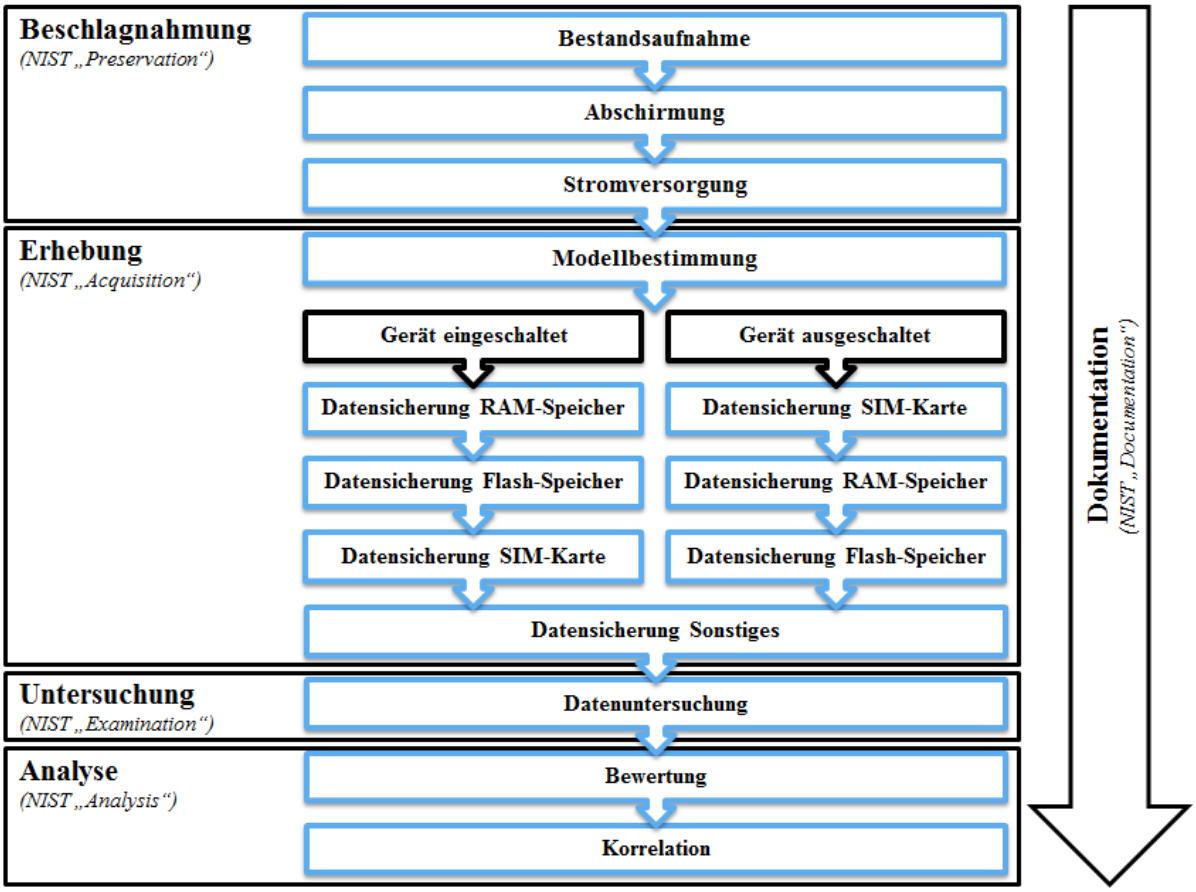

Abbildung 9: Forensisches Prozessmodell des NIST

Entsprechend des Leitfadens vom NIST werden nachfolgend die Kernaspekte der einzelnen Prozessabschnitte des Modells, beginnend mit der Beschlagnahmung, vorgestellt [Nat07].

Beschlagnahmung

Die Beschlagnahmung (vgl. [Nat07] S. 29-37) beginnt mit einer ausführlichen Bestandsaufnahme. Hierbei erfolgt die Erkundung des Fundorts, im Zuge derer Smartphone und Zubehör, wie externe Flash-Speicher, Ladekabel, Rechnungen, Handbücher sowie Verpackungen, sichergestellt werden sollen. Währenddessen ist es unerlässlich, eine Dokumentation anzufertigen, welche unter anderem Fotos der beschlagnahmten Gegenstände enthält. Auch gilt es, nach Möglichkeit den Besitzer des Geräts zu befragen, um bereits erste Informationen bezüglich verwendeter Sicherheitscodes, wie Passwörter, zu erhalten.

Anschließend muss das Smartphone abgeschirmt werden, um eine Veränderung der Daten und folglich eine Verletzung der Integrität zu vermeiden. Dies beinhaltet sowohl die Entfernung existierender Kabelverbindungen als auch die Isolation des Geräts gegenüber potentiellen Funkverbindungen. Ferner ist darauf zu achten, ein Transportbehältnis zu wählen, das eine Interaktion mit dem Gerät durch das Drücken beliebiger Tasten verhindert.

Zuletzt sollte, vor allem für den Transport, eine Stromversorgung bereitgestellt werden, die den Verlust von flüchtigen und temporär zugreifbaren Daten des Smartphones aufgrund eines niedrigen Ladezustands verhindert.

Mit der Beschlagnahmung beginnt die Etablierung der Chain of Custody, welche im gesamten Prozess der Untersuchung gewissenhaft fortgeführt werden muss.

Erhebung

Nach der Überführung des Geräts in ein forensisches Labor folgt der Prozessabschnitt zur Erhebung bzw. Sicherung der Daten (vgl. [Nat07] S. 38-55). Für die Wahl der zu diesem Zweck benötigten Werkzeuge ist es notwendig, vorerst das Modell des zu Grunde liegenden Smartphones zu bestimmen. Eigenschaften, Beschriftungen und Zubehör, wie Rechnungen, Handbücher und Verpackungen des Geräts, können dabei hilfreich sein.

Im Anschluss an die Modellbestimmung erfolgt die eigentliche Sicherung der Daten. Hierbei wird zwischen ein- und ausgeschalteten Smartphones unterschieden.

Während bei bereits eingeschalteten Geräten die Sicherung mit dem RAM und Flash-Speicher beginnen sollte, ist bei ausgeschalteten Geräten der Speicher der SIM-Karte zu bevorzugen. Mit dieser Unterscheidung wird das Ziel angestrebt, vor allem einen Datenverlust zu verhindern, der mit der Abschaltung des Geräts zur Entnahme der SIM-Karte einhergehen kann. Die Entnahme der Karte erlaubt schließlich eine vollständige Erhebung der Daten unter Verwendung eines Kartenlesers. Ein Zugriff auf den RAM und Flash-Speicher erfordert grundsätzlich den Aufbau einer Verbindung zwischen dem Smartphone und einer forensischen Workstation.

Zuletzt werden weitere für den Vorfall essentielle Medien wie externe Flash-Speicher angeführt, die ebenfalls auszubauen und mit Hilfe eines Kartenlesers zu sichern sind.

Unter Umständen erfordert die Erhebung aufgrund bestehender Schutzmaßnahmen, die einen Zugriff auf die Speichermedien einschränken, die Ergreifung weiterer Methoden.

Untersuchung

Die nach der Erhebung angefertigten Sicherungen werden anschließend einer Untersuchung unterzogen (vgl. [Nat07] S. 56-64). Diese sieht z. B. die Bestimmung von Zeichenketten (z. B. Kontaktnamen), Dateiendungen (z. B. `.docx` für Word-Dokumente) oder Dateisignaturen (z. B. `FFD8` als Hexadezimalwert für JPEG) vor, nach welchen mit Hilfe geeigneter Werkzeuge innerhalb der Sicherungen gesucht wird. Ferner umfasst dies auch bei Bedarf die Wiederherstellung gelöschter Dateien. Ziel ist es, digitale Spuren, die mit dem Vorfall in Verbindung stehen und als Beweismittel genutzt werden können, zu extrahieren. Zeitgleich findet hierbei eine Separierung von relevanten und irrelevanten Informationen statt.

Analyse

Die extrahierten Spuren müssen darauf folgend einer detaillierten Analyse unterzogen werden (vgl. [Nat07] S. 56-64). Diese setzt sich aus einer Korrelation und Bewertung der jeweiligen Daten zusammen.

Dokumentation

Letztendlich gilt es, sämtliche durchgeführten Tätigkeiten, gesicherten sowie veränderten Daten und gewonnenen Spuren sowie Ergebnisse schriftlich festzuhalten (vgl. [Nat07] S. 65-67).

Dies erfordert, dass von Beginn der Untersuchung an prozessbegleitend ein Verlaufsprotokoll geführt wird. Der abschließende Bericht soll einer dedizierten Zielgruppe schließlich einen Einblick in die vergangene forensische Untersuchung geben.

3.1.3 Modell „Geschonneck"

Der Autor Alexander Geschonneck veröffentliche 2011 die 5. Auflage seines Buchs „Computer-Forensik – Computerstraftaten erkennen, ermitteln, aufklären". Das Buch gibt eine wesentliche Einführung in die Grundlagen der IT-Forensik und gilt als Standardwerk auf diesem Gebiet. Einen darin fokussierten und im Detail betrachteten Teilaspekt stellt die forensische Untersuchung von mobilen Geräten wie Smartphones dar. Hierzu wurde ein Ablaufschema veröffentlicht (vgl. [Ges11] S. 283), welches vollständig im Anhang A dieser Ausarbeitung enthalten ist. Die Untersuchungsschritte des Ablaufs lassen sich wie folgt zusammenfassen und den Prozessabschnitten „Beschlagnahmung", „Erhebung, Untersuchung und Analyse" sowie „Dokumentation" zuordnen (siehe Abbildung 10):

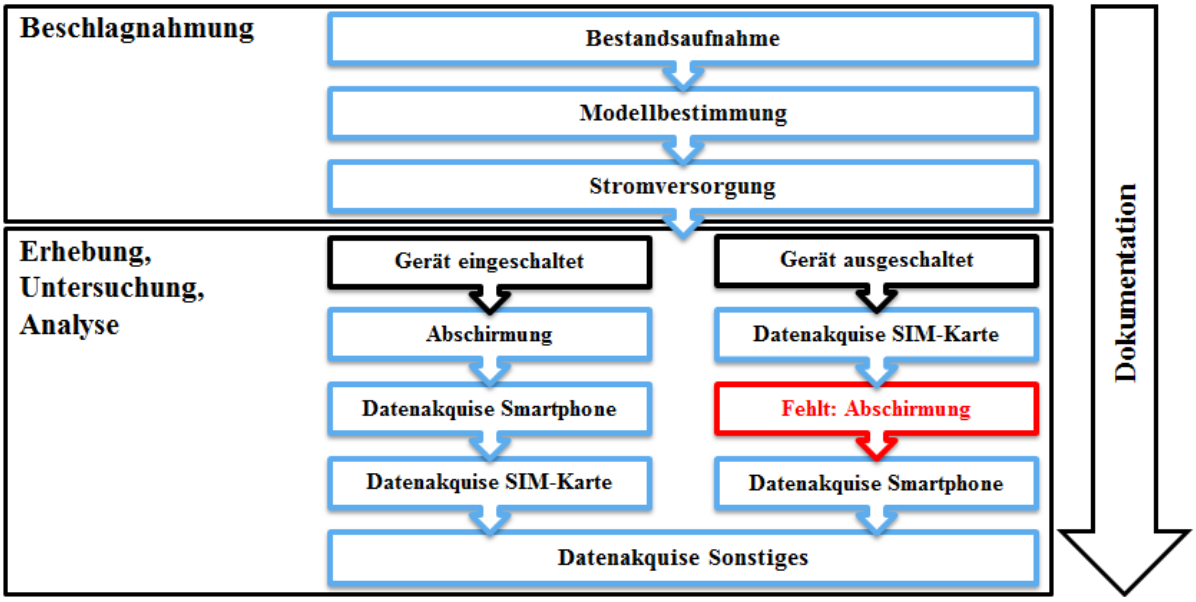

Abbildung 10: Forensisches Prozessmodell von Alexander Geschonneck

Angelehnt an diese vereinfachte Darstellung des Ablaufschemas erfolgt anschließend die Beschreibung der wesentlichen Kernaspekte der jeweiligen Prozessabschnitte (vgl. [Ges11] S. 283).

Beschlagnahmung

Die Beschlagnahmung beginnt mit der Bestandsaufnahme, die eine Erkundung sowie Dokumentation des Fundorts und der sicherzustellenden Gegenstände beinhaltet. Die Dokumentation sieht unter anderem die Anfertigung von Fotos vor.

An die Bestandsaufnahme schließt sich die Modellbestimmung an, in welcher das Modell des beschlagnahmten Smartphones identifiziert wird. Dies ist essentiell für sämtliche weiteren Untersuchungsschritte und die damit verbundene Wahl geeigneter Werkzeuge.

Auf Basis des Modells ist schließlich eine Stromversorgung herzustellen. Dadurch soll vor allem bei eingeschalteten Geräten der Verlust von flüchtigen und unter Umständen nur temporär zugreifbaren Daten verhindert werden.

Erhebung, Untersuchung und Analyse

Die im Anschluss an die Beschlagnahmung gebotene Vorgehensweise ist vom vorliegenden Betriebszustand des Smartphones abhängig.

Ist das Gerät eingeschaltet, sind zur Gewährleistung der Datenintegrität vorerst Tätigkeiten zur Abschirmung zu ergreifen. Dies beinhaltet sowohl die Entfernung existierender Kabelverbindungen als auch die Isolation des Geräts gegenüber potentiellen Funkverbindungen. Nach geeigneter Abschirmung kann die eigentliche Datenakquise des Smartphones durchgeführt werden. Der Begriff der Datenakquise umfasst die Erhebung der zu sichernden Daten, deren Untersuchung zur Extraktion potentieller Spuren und die Analyse der extrahierten Spuren in Form einer Korrelation und Bewertung. Danach ist das Gerät zur Entnahme und anschließenden Datenakquise des Speichers der SIM-Karte auszuschalten.

Sollte das Smartphone bereits während der Beschlagnahmung ausgeschaltet sein, beginnt die Datenakquise stattdessen mit dem Speicher der SIM-Karte. Anschließend soll das Gerät mit eingesetzter SIM-Karte zur Erhebung, Untersuchung und Analyse der darin vorliegenden Daten eingeschaltet werden. Entsprechend des vorgegebenen Ablaufschemas wird in diesem Fall keine Abschirmung benötigt. Durch das Einschalten kommt es jedoch üblicherweise zum Aufbau von Datenverbindungen, die maßgebliche Inhalte des internen Flash-Speichers verändern können. Aus diesem Grund sollte eine Abschirmung grundsätzlich, unabhängig vom Betriebszustand des Geräts, in Betracht gezogen werden.

Zuletzt sind alle weiteren für den Vorfall wichtigen Medien wie externe Flash-Speicher zu sichern, zu untersuchen und zu analysieren. Diesbezüglich werden auch Mailbox-Nachrichten angeführt, die potentielle Spuren enthalten und als Beweismittel dienen können.

Bestehende Schutzmaßnahmen, die einen Zugriff auf die Speichermedien einschränken, erfordern unter Umständen die Ergreifung weiterer Methoden während der Erhebung. Die Suche nach gelöschten Dateien wird aufgrund hoher Datenmengen, die damit einhergehen, nur empfohlen, wenn Beweismittel zur Aufklärung eines Vorfalls fehlen.

Dokumentation

Zuletzt sind unter Zuhilfenahme eines prozessbegleitenden Verlaufsprotokolls die zuvor durchgeführten Tätigkeiten, gesicherten sowie veränderten Daten und gewonnen Spuren sowie Ergebnisse in einem Bericht zusammenzufassen. Dieser soll einer ausgewählten Zielgruppe einen Einblick in die vergangene forensische Untersuchung geben.

3.2 Modellaufbau

Der direkte Vergleich der etablierten Modelle zur Durchführung einer forensischen Untersuchung zeigt, dass vorwiegend Unterschiede in der Unterteilung der Prozessabschnitte existieren. Ein Beispiel hierfür sind die Erhebung, Untersuchung und Analyse, die Alexander Geschonneck im Gegensatz zum NIST nicht voneinander separiert. Ferner unterscheidet sich die Zuordnung der einzelnen Untersuchungsschritte zu den jeweiligen Abschnitten. Demnach erfolgt die Abschirmung des Smartphones im Ansatz des NIST bereits während der Beschlagnahmung unabhängig vom Betriebszustand des Geräts (ein- oder ausgeschaltet). Alexander Geschonneck hingegen nimmt eine Abschirmung nur bei eingeschalteten Smartphones im Abschnitt der Erhebung, Untersuchung und Analyse vor, was schließlich zu einer Verletzung der Integrität der Daten führen kann (siehe Kapitel 3.1.3).

Aufgrund dieser und weiterer nicht zu vernachlässigenden Unterschiede ist es notwendig, ein neues Prozessmodell zu generieren, das als Grundlage für die weiteren Untersuchungen dient. Hierzu erfolgt mit Hilfe der eingeführten einheitlichen Terminologie ein direkter Vergleich zwischen den beschriebenen Ansätzen. Dies erlaubt die Ermittlung von Übereinstimmungen und Abweichungen sowie die inhaltliche Bewertung der Vorgehensweisen hinsichtlich ihrer Plausibilität.

Anschließend werden die jeweiligen Stärken und Schwächen herausgearbeitet. Auf Basis der Erkenntnisse ergibt sich schließlich ein neu entwickeltes Modell, welches die positiven Eigenschaften seiner Vorgänger vereint.

Das aus der Bewertung resultierende Modell untergliedert sich in die Prozessabschnitte „Strategische Vorbereitung“, „Operationale Vorbereitung“, „Datensammlung“, „Datenuntersuchung“, „Datenanalyse“ und „Dokumentation“. Diese Abschnitte gruppieren die in Abbildung 11 blau dargestellten Prozessbausteine, welche schließlich logisch zusammengehörige Untersuchungsschritte vereinigen.

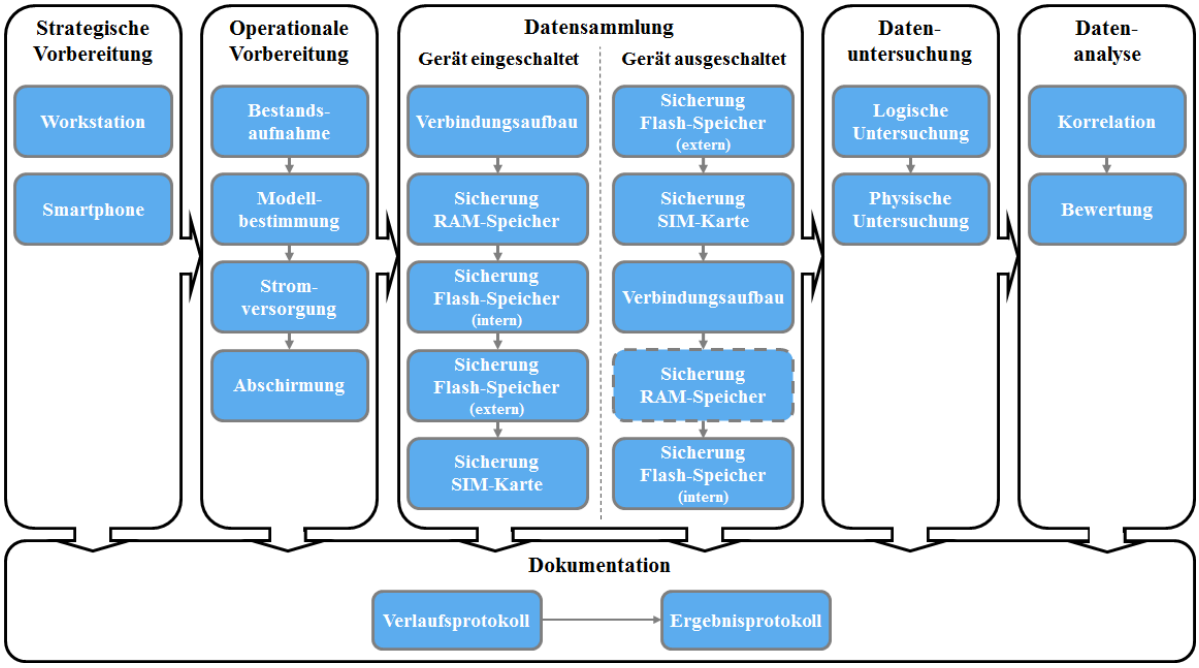

Abbildung 11: Entwickeltes forensisches Prozessmodell

Auf die Herleitung dieser Struktur wird im Detail in den nachfolgenden Kapiteln 3.2.1 „Prozessabschnitte“ und 3.2.2 „Prozessbausteine“ eingegangen.

3.2.1 Prozessabschnitte

Das Bundesamt für Sicherheit in der Informationstechnik bietet mit seiner Einteilung des Modells in Prozessabschnitte eine essentielle Grundlage zur Strukturierung, die im neu entwickelten Ansatz eingesetzt wird.

Flussdiagramme, wie das von Alexander Geschonneck (siehe Anhang A), eignen sich zwar vor allem zur Abbildung von Verzweigungen, verlieren jedoch mit zunehmendem Informationsgehalt an Übersichtlichkeit. Daraus resultieren im ungünstigsten Fall nicht zu vernachlässigende Fehler in der Vorgehensweise. Die übersichtliche Darstellungsform des BSI hingegen ermöglicht eine klare Strukturierung der Untersuchungsschritte und damit eine schnelle Erfassung des gesamten Prozesses.

Die Unterteilung in sechs Abschnitte erlaubt die Erfassung eines zeitlichen Verlaufs innerhalb der forensischen Untersuchung. Darüber hinaus bewirkt die Abgrenzung der Abschnitte eine geeignete inhaltliche Klassifizierung einzelner Untersuchungsschritte. Demzufolge wird zum einen zwischen vorbereitenden proaktiven und vorbereitenden reaktiven Tätigkeiten unterschieden. Zum anderen existiert eine Separierung zwischen Daten sichernden, extrahierenden und analysierenden Tätigkeiten. Eine vergleichbare Unterteilung erfolgt darüber hinaus sowohl im Modell des NIST (vgl. Abbildung 9) als auch im zu Anfang des Kapitels angegebenen Paper „Smartphone Forensic Investigation Process Model“ (vgl. [GTA12] S. 333).

Insgesamt lassen sich die Prozessabschnitte des BSI aufgrund ihrer Allgemeingültigkeit auf die forensische Untersuchung von Smartphones übertragen. Es ist lediglich notwendig, inhaltliche Ergänzungen, in Form von neuen Prozessbausteinen, vorzunehmen, die mit den besonderen Bedingungen der Geräte in Bezug auf die IT-Forensik einhergehen. Anzuführen wäre hier beispielsweise die Abschirmung der Geräte, welche bei traditionellen IT-Systemen wie Computern in der Regel keine Anwendung findet.

3.2.2 Prozessbausteine

Die untersuchten spezialisierten Modelle bilden die Basis für die einzelnen Prozessbausteine der zuvor dargestellten Prozessabschnitte. Diese werden anschließend, eingegliedert in die zugehörigen Abschnitte, im Detail betrachtet.

Strategische Vorbereitung

Die strategische Vorbereitung, die vor Eintreten eines Vorfalls zum Tragen kommt, wird in keinem der spezialisierten Modelle berücksichtigt.

Daraus resultiert eine fehlende Abdeckung von wesentlichen Untersuchungsschritten, welche aus der Perspektive eines Administrators von Interesse und zur Darstellung einer vollwertigen Vorgehensweise zu berücksichtigen sind. Die vom BSI angeführten Untersuchungsschritte widmen sich übergeordnet der Vorbereitung einer forensischen Workstation und des potentiell zu untersuchenden IT-Systems (in diesem Fall ein Smartphone). Entsprechend dieser Vorgabe wird der Abschnitt in die Prozessbausteine **Workstation** und **Smartphone** eingeteilt.

Operationale Vorbereitung

Der Prozessabschnitt zur operationalen Vorbereitung fällt in Bezug auf die spezialisierten Ansätze unter den Begriff der „Beschlagnahmung“ (siehe Kapitel 3.1.2 und 3.1.3). Diese beginnt einstimmig (ebenso im Paper [GTA12] S. 333) mit der Erkundung und Dokumentation des Fundorts sowie der sicherzustellenden Gegenstände. Mit der Sicherstellung geht ein erster Kontakt mit den potentiellen Beweismitteln einher. Um eine lückenlose Dokumentation des Verbleibs der Beweismittel und deren Einsichtnahme über den gesamten Untersuchungsverlauf zu gewährleisten, sollte daher an dieser Stelle die Etablierung einer Chain of Custody erfolgen (vgl. [GTA12] S. 334 und [Nat07] S. 33). Die aufgeführten Untersuchungsschritte werden künftig im Prozessbaustein **Bestandsaufnahme** zusammengefasst.

Für alle weiteren Vorgänge ist die Kenntnis des Modells des Smartphones unerlässlich. Die damit einhergehenden Tätigkeiten sind im Prozessbaustein **Modellbestimmung** vereinigt und sollten, wie im Ansatz von Alexander Geschonneck dargestellt, direkt der Bestandsaufnahme folgen. Das NIST strebt eine Modellbestimmung erst im Prozessabschnitt der Datensammlung an. Dies erschwert jedoch die Anbringung einer passenden Stromquelle sowie die Ergreifung von geeigneten abschirmenden Tätigkeiten und erweist sich somit als ungeeignet.

Zur Gewährleistung des ununterbrochenen Betriebs des Smartphones ist mit Hilfe der ermittelten Gerätespezifikationen eine Stromversorgung einzurichten. Dies hat vor allem eine Relevanz bezüglich der Abschirmung des Geräts. Strahlungsschutzmaßnahmen können zur Erhöhung der Sendeleistung und folglich zu einem gesteigerten Energieverbrauch führen. Ein Prozessbaustein **Stromversorgung** ist demgemäß, wie ebenfalls die analysierten Ansätze zeigen, essentiell.

Zuletzt ist in Übereinstimmung mit dem NIST eine Isolation des Smartphones durchzuführen. Der hierfür vorgesehene Baustein **Abschirmung** wird im Ansatz von Alexander Geschonneck erst im Zuge der Datensammlung berücksichtigt. Dies trifft jedoch nur dann zu, wenn das Gerät während der Beschlagnahmung eingeschaltet ist. In ähnlicher Weise wird dies auch im Modell des eingangs genannten Papers gehandhabt (vgl. [GTA12] S. 334-335). Ein nachträgliches Einschalten, welches zur Sicherung des internen Flash-Speichers notwendig ist, bewirkt üblicherweise den Aufbau von Datenverbindungen. Infolgedessen würden im ungünstigsten Fall bestehende Daten und somit relevante Spuren manipuliert werden. Wie in Kapitel 3.1.3 dargestellt, sollte daher frühzeitig und unabhängig vom Betriebszustand des Geräts für eine Abschirmung gesorgt werden.

Datensammlung

Zu den forensisch bedeutsamen Speichermedien zählen der RAM, der interne sowie externe Flash-Speicher und der Speicher der SIM-Karte. Diese Komponenten bedingen unterschiedliche Methoden sowie Werkzeuge zur Sicherung und sollten daher vergleichbar zu den analysierten Ansätzen in separate Bausteine gegliedert werden. Diese lauten **Sicherung RAM-Speicher**, **Sicherung Flash-Speicher (intern)**, **Sicherung Flash-Speicher (extern)** und **Sicherung SIM-Karte**. Eine ähnliche Einteilung wird auch im Modell des anfangs angegebenen Papers vorgenommen (vgl. [GTA12] S. 333).

Die Reihenfolge, in der eine Datensammlung erfolgen kann, ist entsprechend der spezialisierten Ansätze vorwiegend abhängig vom aktuellen Betriebszustand des Geräts. Darüber hinaus beeinflussen gegebene Schutzmaßnahmen maßgeblich die durchzuführenden Schritte. Diese Besonderheiten werden tiefgehend im Kapitel 4 „Android-Prozess“ beleuchtet. Idealerweise sollte im **eingeschalteten Zustand** aufgrund seiner Flüchtigkeit mit dem RAM begonnen werden. Zur Vermeidung eines weiteren Datenverlusts ist vor einer möglichen Abschaltung des Geräts zur Entnahme und Sicherung der entfernbaren Medien der interne Flash-Speicher zu sichern. Im **ausgeschalteten Zustand** gilt es stattdessen, mit der Entnahme und Erhebung der entfernbaren Speichermedien zu starten. Auf diese Weise bleiben die erfolgreich gewonnen Sicherungen der Medien unbeeinträchtigt von Veränderungen, die sich während der weiteren Datensammlung ergeben können.

Die Sicherung der Daten des RAMs und des internen Flash-Speichers erfordert den Aufbau einer Verbindung zwischen dem Smartphone und einer forensischen Workstation. Eine aktive Verbindung kann bei manchen Modellen eine automatische Einschaltung des Geräts bewirken. Der Zeitpunkt hierfür hängt somit ebenfalls maßgeblich vom Betriebszustand ab, weshalb ein gesonderter Prozessbaustein **Verbindungsaufbau** erstellt wurde. Dieser befindet sich unmittelbar vor der Sicherung der internen Medien.

Datenuntersuchung

Die Untersuchung der Sicherungen zur Extraktion digitaler Spuren kann im Wesentlichen auf zwei Arten erfolgen: logisch und physisch (vgl. [BZ13] S. 127).

Im logischen Ansatz wird das vorhandene Dateisystem eines erhobenen 1:1-Abbilds, vornehmlich des internen und externen Flash-Speichers, lesend in das Betriebssystem der forensischen Workstation eingebunden. Unter Verwendung des Dateisystems kann schließlich gezielt nach Dateien und folglich potentiellen Spuren gesucht werden. Nicht allokierte Speicherbereiche sind nicht über das Dateisystem zugänglich, weshalb z. B. eine Wiederherstellung von gelöschten Dateien auf diesem Weg ausgeschlossen ist.

Der physische Ansatz sieht die Untersuchung auf Basis der Rohdaten der Sicherungen vor. Bei einem vollständigen physikalischen 1:1-Abbild werden hierbei nicht allokierte Speicherbereiche miteingeschlossen. Somit können mit der physischen Untersuchung auch gelöschte Dateien wiederhergestellt werden. Besonders die Erstellung des vollständigen 1:1-Abbilds des internen Flash-Speichers stellt im Smartphone-Umfeld aufgrund der Datensammlung über bereitgestellte Betriebssystemschnittstellen eine Herausforderung dar.

Wie Alexander Geschonneck in seinem Modell andeutet, hat die Wahl der Untersuchungsart maßgebliche Auswirkungen auf die Datenmenge. Diese ist bei Anwendung des physischen Ansatzes deutlich höher, weshalb zu Beginn eine logische Untersuchung durchgeführt werden sollte. Zur Darstellung dieser Reihenfolge setzt sich der Prozessabschnitt aus den zwei Bausteinen **Logische Untersuchung** und **Physische Untersuchung** zusammen.

Datenanalyse

Die Datenanalyse untergliedert sich gemäß der bestehenden Ansätze in zwei Kernaufgaben.

Zum einen in eine **Korrelation**, die das Herstellen eines inhaltlichen und zeitlichen Zusammenhangs zwischen den digitalen Spuren vorsieht. Zum anderen in eine inhaltliche **Bewertung** der extrahierten Spuren und deren Bezug zum gegebenen Vorfall. Zur Trennung dieser Tätigkeiten erfolgt auch in diesem Modell eine Aufteilung dieser Untersuchungsschritte in zwei voneinander unabhängige Prozessbausteine.

Dokumentation

Ein prozessbegleitendes Verlaufsprotokoll ist entsprechend der analysierten Ansätzen ein zentraler Ausgangspunkt zur Anfertigung eines Ergebnisprotokolls (vgl. hierzu auch [GTA12] S. 334). Das Ergebnisprotokoll soll sämtliche für eine Zielgruppe entscheidenden Informationen zusammenfassen und somit einen umfassenden Einblick in die durchgeführte forensische Untersuchung geben (vgl. hierzu auch [GTA12] S. 337). In Anbetracht dieser Vorgabe wird somit zwischen den Prozessbausteinen **Verlaufsprotokoll** und **Ergebnisprotokoll** differenziert.

Im Kapitel „Prozessmodelle“ erfolgte zu Beginn eine Analyse anerkannter Ansätze von BSI, NIST und Alexander Geschonneck zur Durchführung einer forensischen Untersuchung. Auf Basis einer heuristischen Bewertung, in der die jeweiligen Stärken und Schwächen der Vorgehensweisen beleuchtet wurden, ist ein neues Prozessmodell entwickelt worden. Dieses untergliedert sich angelehnt an das BSI in die sechs Prozessabschnitte „Strategische Vorbereitung“, „Operationale Vorbereitung“, „Datensammlung“, „Datenuntersuchung“, „Datenanalyse“ und „Dokumentation“. Durch eine solche Einteilung sind eine übersichtliche Darstellung des gesamten Prozesses und die Erfassung eines zeitlichen Verlaufs möglich. Die Inhalte der Abschnitte bilden 17 Prozessbausteine, die aus den Ansätzen vom NIST und von Geschonneck ermittelt wurden und logisch zusammengehörige Schritte zur forensischen Untersuchung von Smartphones vereinigen. Das Prozessmodell dient schließlich als essentieller Rahmen für die tiefgehenden technischen Betrachtungen bezüglich Android.

4 Android-Prozess

Im Zuge dieser Ausarbeitung wird ein Leitfaden erstellt, der eine systematische forensische Untersuchung von Smartphones basierend auf Android erlaubt. Bisher wurden innerhalb eines neu entwickelten Prozessmodells lediglich die allgemein mit den Geräten einhergehenden Besonderheiten in Bezug auf die IT-Forensik abgebildet. Die spezifischen technischen Aspekte, die maßgeblich vom zu Grunde liegenden Betriebssystem abhängen, werden folglich noch nicht abgedeckt.

In diesem Kapitel werden nun die einzelnen Prozessbausteine der sechs Prozessabschnitte (in Abbildung 12 durch blaue Quadrate symbolisiert) ausführlich im Zusammenhang mit Android untersucht. Hierzu gilt es einleitend, den aktuellen Stand der Technik jedes Bausteins zu beleuchten. Dies beinhaltet auch die Analyse gegenwärtiger Problemstellungen und Lösungsansätze. Auf Grundlage der gewonnen Erkenntnisse werden schließlich Methoden aufgezeigt, die einen systematischen forensischen Umgang mit den spezifischen Smartphones ermöglichen.

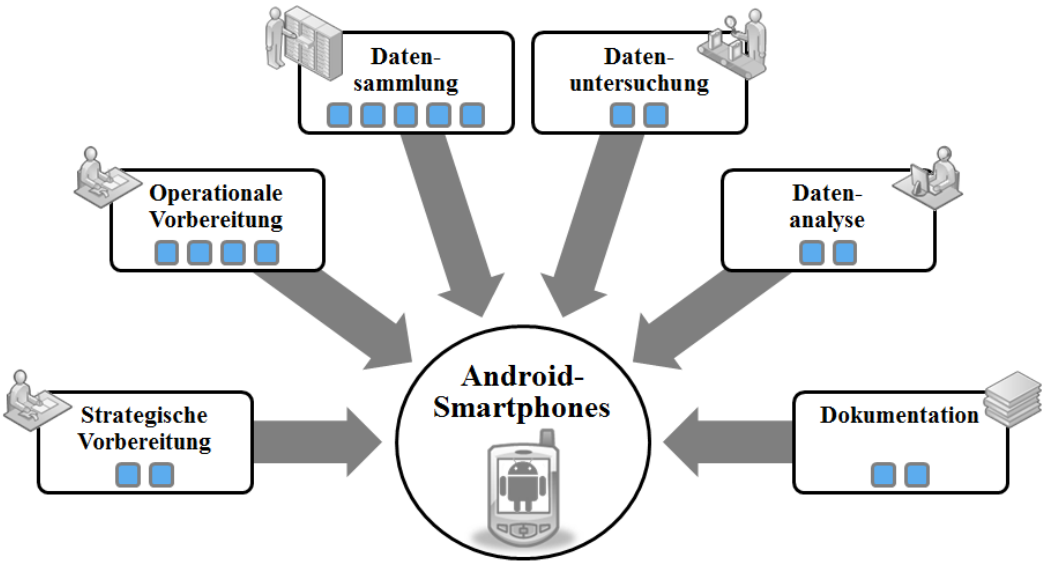

Abbildung 12: Anwendung des forensischen Prozessmodells auf Android-Smartphones

Ein besonderes Augenmerk wird im Leitfaden auf die Datensammlung gelegt, da diese die Basis für alle weiterführenden Untersuchungsschritte bildet. Essentielle Bestandteile des Prozessabschnitts werden demnach zusätzlich in Kapitel 5 „Datensammlung“ ausführlich beschrieben.

4.1 Strategische Vorbereitung

Die Strategische Vorbereitung untergliedert sich in die Prozessbausteine **Workstation** und **Smartphone**. Darin sind ausschließlich Untersuchungsschritte und Methoden enthalten, welche vor Eintreten eines Vorfalls auszuführen sind und eine anschließende forensische Untersuchung unterstützen sollen. Oftmals stehen die damit verbundenen Methoden im Spannungsfeld zwischen Effektivität, Sicherheit und Datenschutz. Die Auswahl an zu ergreifenden Möglichkeiten ist deshalb, besonders im Hinblick auf das zu untersuchende Smartphone, erheblich eingeschränkt.

4.1.1 Workstation

Die forensische Untersuchung erfolgt unter Zuhilfenahme einer Workstation. Bei der Einrichtung der Workstation sind verschiedene Aspekte zu beachten, die allgemein für die IT-Forensik und spezifisch für die Smartphone-Forensik von Bedeutung sind. Die betrachteten Aspekte bieten ein wertvolles Grundgerüst für die Workstation. Abhängig vom existierenden Vorfall gilt es, in den nachfolgenden Prozessbausteinen Ergänzungen, z. B. um zusätzliche Werkzeuge, vorzunehmen.

Allgemeine Aspekte

Das BSI und das NIST empfehlen, die Workstation zur forensischen Untersuchung als virtuelle Maschine aufzusetzen (vgl. [Bun11] S. 57 und [Nat07] S. 22). Dadurch ist es möglich, mit Hilfe von „Snapshots“ diskrete Systemzustände zu sichern und bei Bedarf auf diese zurückzugreifen. Ferner können mit diesem Ansatz entsprechend der Empfehlungen vom NIST verschiedene Werkzeuge mit einem geringfügigen Aufwand in voneinander isolierte Umgebungen installiert werden (siehe Kapitel 2.3.2). Beim Aufsetzen ist darauf zu achten, die Zeit über das „Network Time Protocol“ (NTP) zu synchronisieren, um eine vertrauenswürdige Referenzzeit im gesamten Untersuchungsverlauf zu schaffen.

Das Betriebssystem und die eingesetzten Werkzeuge der Workstation sind zur Beseitigung möglicher Sicherheitslücken stets auf dem aktuellen Stand zu halten. Damit eine Manipulation der Werkzeuge ausgeschlossen werden kann, ist es ferner notwendig, in regelmäßigen Abständen eine Integritätsüberprüfung durchzuführen. Hierbei ist ein kryptografischer Hash über die Ursprungsdaten und der zu überprüfenden Daten zu berechnen.

Die Übereinstimmung der Hashwerte belegt, dass keine Veränderungen vorgenommen wurden. Der überprüfte Systemzustand kann bei Verwendung einer virtuellen Maschine anschließend gesichert und als Ausgangspunkt für eine Untersuchung verwendet werden.

Die Zugriffsrechte sind auf das zur Ausführung der jeweiligen Tätigkeiten erforderliche Maß zu begrenzen (vgl. [Bun11] S. 57). Demzufolge sollten Werkzeuge ausschließlich auf die tatsächlich benötigten Ressourcen zugreifen dürfen. Damit wird das Ziel verfolgt, eine Manipulation am System weitestgehend einzudämmen. Gelangt beispielsweise Schadcode über das zu untersuchende Smartphone auf die Workstation, kann der Schaden dadurch erheblich begrenzt werden. Außerdem ist jegliche Art von Fernzugriffen (z. B. über Secure Shell) auf die Workstation zu unterbinden.

Spezifische Aspekte

Für die forensische Untersuchung eines auf Android basierenden Smartphones werden unter anderem Werkzeuge, wie die Android Debug Bridge, aus dem zugehörigen Software Development Kit benötigt (siehe Kapitel 2.2.3). Innerhalb der strategischen Vorbereitung sollte daher für die nachfolgenden Prozessbausteine die Installation des SDKs auf der Workstation vorgenommen werden.

An die Workstation angeschlossene Speichermedien werden in der Regel automatisch durch das Betriebssystem in das bestehende Dateisystem eingebunden. Infolgedessen kann es bereits zu einer Veränderung vorhandener Daten kommen. Dies betrifft ebenfalls den internen und externen Flash-Speicher eines Smartphones, das für eine Datensammlung mit der Workstation verbunden wird. Zur Gewährleistung der Integrität der gespeicherten Daten ist daher das automatische Einbinden vorzeitig abzuschalten (vgl. [Hoo11] S. 15 und S. 98).

Für eine Einsicht in den Kommunikationsverlauf zwischen forensischer Workstation und den zu sichernden Speichermedien des Smartphones empfiehlt sich die Installation eines Werkzeugs zum „Port-Monitoring“ (vgl. [Nat07] S. 15). Diese Werkzeuge zeichnen sämtliche über die Hardware-Schnittstelle transferierten Daten auf und speichern diese als Logs ab. Dadurch ist es möglich, den gesamten Untersuchungsprozess und somit auch mögliche Veränderungen an den Daten zu protokollieren. Entscheidend ist dies in Fällen, in denen die Methoden zur Datensammlung die Verwendung eines Writeblockers ausschließen. Ein in der Fachwelt akzeptiertes Werkzeug für diesen Zweck ist „Wireshark“ [Wir13].

4.1.2 Smartphone

Die Ergreifung von proaktiven Methoden durch den Administrator auf dem zu untersuchenden IT-System ist eine wesentliche Aufgabe während der strategischen Vorbereitung. Im Leitfaden des BSI werden diesbezüglich die Sicherstellung einer korrekten Systemzeit und die Aktivierung von Protokollierungsdiensten aufgeführt (vgl. [Bun11] S. 38 und S. 50). Diese Tätigkeiten lassen sich auch auf Android-Smartphones übertragen.

Zeiteinstellung

Eine besondere Relevanz in einer forensischen Untersuchung hat die vom Betriebssystem und Dateisystem verwendete Zeit des Smartphones. Durch eine falsche Konfiguration wird eine Datenanalyse, insbesondere die Korrelation der extrahierten Daten, erschwert. Die Sicherstellung einer korrekten und vertrauenswürdigen Systemzeit ist daher essentiell. Android bietet in der Regel zwei mögliche Zeitquellen an.

Die Systemzeit kann manuell durch den Besitzer in den Einstellungen des Smartphones konfiguriert werden. Eine Echtzeituhr (engl. Real Time Clock) bildet hierbei die Basis und sorgt typischerweise im ausgeschalteten Zustand für die Fortführung der aktuellen Zeit. Mögliche Abweichungen bei der Einrichtung, z. B. durch Gebrauch einer ungenauen Referenzzeit, werden nicht automatisch durch das System korrigiert. Daraus resultieren fehlerhafte Zeitstempel in Dateien und Verzeichnissen.

Alternativ kann die Systemzeit über das Mobilfunknetz bezogen werden. Durch regelmäßige Synchronisierungen werden Abweichungen weitestgehend ausgeschlossen. Folglich steht im Vergleich zur manuellen Variante eine deutlich vertrauenswürdigere Zeitbasis zur Verfügung. Die Verwendung einer vom Netz bereitgestellten Zeit sollte aus diesem Grund in der strategischen Vorbereitung stets präferiert werden.

Protokollierung

Im Leitfaden des BSI wird im IT-Umfeld als proaktive Methode stets die Protokollierung von Logdaten angeführt. Im Bereich der forensischen Untersuchung von Smartphones basierend auf Android ist vor allem das Werkzeug „Logcat“ von Bedeutung [And13g]. Logcat ermöglicht das Sammeln und Sichten von Logdaten, die durch das Betriebssystem oder installierte Applikationen generiert und in einzelnen Ringpuffern abgelegt werden.

Erreicht ein Ringpuffer seine maximale Größe, die vom zu Grunde liegenden Smartphone abhängt, dann werden bereits vorhandene Daten überschrieben. In Android befinden sich die angelegten Dateien der Puffer typischerweise unter `/dev/log`. Der Zugriff auf Logcat kann über die Android Debug Bridge durch Eingabe des Befehls `logcat` in eine geöffnete Shell oder durch den Android Device Monitor erfolgen.

Ursprünglich wird das Werkzeug innerhalb der Softwareentwicklung zum Debuggen verwendet. Aus forensischer Sicht können die in den Logdaten enthaltenen Informationen Aufschluss über die zuletzt ausgeführten Aktivitäten auf dem Smartphone geben. Beispielsweise kann überprüft werden, ob eine spezifische Applikation zu einem bestimmten Zeitpunkt aufgerufen wurde. Sofern die Daten nicht dauerhaft entweder lokal auf dem internen Flash-Speicher oder zentral auf einem Logserver gesichert werden, besteht die Gefahr, bei jeder Interaktion mit dem Gerät wertvolle Spuren zu vernichten. Die am Smartphone ausgeführten Aktivitäten sollten daher, wie in Kapitel 2.3.1 angeführt, auf ein Minimum begrenzt werden. Ferner liegen die Daten in einem Tmpfs, d. h. einem Dateisystem im RAM, und sind aufgrund ihrer Flüchtigkeit zu Beginn zu sichern. Danach eignet sich Logcat als Werkzeug zur Überwachung der Interaktionen mit dem Gerät, um mögliche Veränderungen bei der Ausführung weiterer Untersuchungsschritte zu identifizieren.

4.2 Operationale Vorbereitung

Die operationale Vorbereitung erfolgt, wie die nachfolgenden Prozessabschnitte, in Abhängigkeit eines zu Grunde liegenden Vorfalls, den es forensisch zu untersuchen gilt. Übergeordnet widmet sich der Abschnitt den Prozessbausteinen **Bestandsaufnahme**, **Modellbestimmung**, **Stromversorgung** und **Abschirmung**. Die darin enthaltenen Untersuchungsschritte und Methoden sind weitestgehend unabhängig vom zu Grunde liegenden Betriebssystem und orientieren sich vor allem an dem Vorgehen der klassischen IT-Forensik.

4.2.1 Bestandsaufnahme

Die Bestandsaufnahme widmet sich der Erkundung und Dokumentation des Fundorts. Die in diesem Prozessbaustein aufgeführten Untersuchungsschritte sind bevorzugt nach dem Vier-Augen-Prinzip durchzuführen. Dadurch soll die Authentizität der ergriffenen Untersuchungsschritte gewährleistet werden.

Wie bei der klassischen Forensik gilt während der Bestandsaufnahme das Gebot, keine Veränderungen an den sicherzustellenden Gegenständen vorzunehmen. Die Gegenstände sollen somit in ihrem Zustand, in dem sie vorgefunden wurden, belassen werden.

Beim ersten Betreten des Fundorts ist darauf zu achten, sämtliche drahtlosen Datenverbindungen, wie z. B. Bluetooth, des in den Bereich eingeführten Equipments zu deaktivieren (vgl. [Nat07] S. 31). Dadurch soll eine ungewollte Interaktion mit dem zu beschlagnahmenden Smartphone, die zu einer Veränderung relevanter digitaler Spuren führen kann, vermieden werden. Im Zuge der Erkundung ist neben dem Smartphone auch sämtliches Zubehör, wie externe Flash-Speicher, Verbindungskabel, Rechnungen, Handbücher und Verpackungen, zu beschlagnahmen. Die Verbindungskabel können vor allem dann von Nutzen sein, wenn ein Smartphone herstellerspezifische oder gar neue Schnittstellen aufweist, für die zum gegenwärtigen Zeitpunkt keine eigenen Kabel vorliegen (vgl. [Hoo11] S. 200-201). Das Zubehör ist allerdings vor der Benutzung zu untersuchen, um eine mögliche Manipulation auszuschließen. Ferner können die zusätzlichen Gegenstände äußerst hilfreich bei der Modellbestimmung sein und Informationen über bereitgestellte Funktionalitäten sowie verwendete Sicherheitscodes des Smartphones enthalten (vgl. [Nat07] S. 32). Darüber hinaus kann nach Möglichkeit der Gerätebesitzer vernommen werden, um die zuvor genannten Informationen in Erfahrung zu bringen.

Die Dokumentation setzt das Anfertigen von Fotos und eine Beschriftung aller Gegenstände mit einer Fallnummer, dem Fundort, einer kurzen Objektbeschreibung, einer Unterschrift und dem Datum der Beschlagnahmung voraus (vgl. [Ges11] S. 84-86, [GTA12] S. 334 und [Nat07] S. 32). Währenddessen ist eine Chain of Custody zu erstellen. Beim Smartphone sind ebenfalls Fingerabdrücke auf dem Touchscreen, unter anderem durch die Anfertigung von Fotos der Fettrückstände, zu sichern. Mit Hilfe der Abdrücke kann in einem späteren Schritt die Rekonstruktion eines zu zeichnenden Musters erfolgen, welches zur Aufhebung des Sperrbildschirms benötigt wird. Liegt das eingeschaltete Gerät bereits im entsperrten Zustand vor und fehlt der zugehörige Sicherheitscode, ist umgehend in den Einstellungen die Zeit zum automatischen Sperren auf den maximalen Wert zu setzen und das USB-Debugging zu aktivieren (vgl. [Hoo11] S. 199). Dadurch soll ein erfolgreicher Verbindungsaufbau zum Smartphone im nachfolgenden Prozessabschnitt gewährleistet werden. Darüber hinaus ist bei einem eingeschalteten Gerät der aktuelle Systemstatus zu dokumentieren.

Dies beinhaltet z. B. die Erfassung der angezeigten Zeit, des Netzstatus, des Batteriezustands und der LED-Aktivitäten (vgl. [Nat07] S. 32). Die Systemzeit gilt es, mit einer Referenzzeit zu vergleichen, um mögliche Abweichungen festzuhalten (vgl. [Nat07] S. 35).

4.2.2 Modellbestimmung

Die weiteren Prozessbausteine sind vom Modell des sichergestellten Smartphones abhängig. Ohne die Identifizierung des Modells ist die Ergreifung weiterer Untersuchungsschritte nur eingeschränkt oder im ungünstigsten Fall überhaupt nicht möglich. Im Unternehmensumfeld können die benötigten Informationen aus einem Mobile Device Management-System bezogen werden. Dieses dient zur zentralen Absicherung, Überwachung und Verwaltung von mobilen Endgeräten im Unternehmensumfeld. Steht ein solches System nicht zur Verfügung, ist das zu Grunde liegende Smartphone genauer zu untersuchen. Durch verschiedene Modifikationen am Gerät kann der Vorgang allerdings erschwert werden. Dazu zählen z. B. das Entfernen von angebrachten Beschriftungen und das Löschen oder Austauschen des originalen Startbildschirms des Betriebssystems (vgl. [Nat07] S. 39).

Die Bestimmung des Modells kann anhand vorhandener Beschriftungen und spezifischer Eigenschaften (z. B. Formfaktor, Gewicht, Design und verfügbare Schnittstellen) erfolgen. Liegt das Gerät bereits im ausgeschalteten Zustand vor, können die Informationen ggf. durch ein Typenschild, das sich unter dem Akku befindet, ergänzt werden. Darauf ist unter anderem der International Mobile Equipment Identifier (IMEI) abgebildet, der die eindeutige Identifikation des Geräts erlaubt. Ist das Gerät eingeschaltet und ein Zugriff gestattet, kann die IMEI auch durch Eingabe der Zeichenfolge `*#06#` abgerufen werden (vgl. [Nat07] S. 40). Ferner lassen sich einige Geräteinformationen im Bereich „Einstellungen“ einsehen. Das Problem bei den beiden zuletzt dargestellten Ansätzen besteht in der Interaktion mit dem Smartphone, die zu einer Veränderung bestehender Daten im RAM führt. Die ermittelten Eigenschaften können schließlich für eine weiterführende Spezifizierung verwendet werden. Das NIST führt in diesem Kontext verschiedene Webseiten[1] auf, die als Eingabe die ermittelten Eigenschaften erwarten (vgl. [Nat07] S. 39).

[1] z. B. *http://www.gsmarena.com/search.php3* oder *http://www.phonescoop.com/phones/finder.php*

Ferner ist für diesen Zweck die Software „UFED Phone Detective“ des auch von Andrew Hoog beschriebenen Herstellers „Cellebrite“ anzuführen [Cel13a]. Cellebrite ist seit 2007 auf dem Gebiet der IT-Forensik tätig und weist Kunden im Bereich der Strafverfolgungsbehörden, des Militärs und des Geheimdienstes auf [Cel13b].

Mit Hilfe des identifizierten Modells kann bereits auf das zu Grunde liegende Betriebssystem geschlossen werden, sofern das Smartphone nicht manipuliert wurde. Etablierte Hersteller, wie Samsung, Sony oder HTC, setzten als Plattform üblicherweise Android ein. Das angezeigte Layout auf dem Bildschirm eignet sich ebenso als Indiz für ein spezifisches Betriebssystem (vgl. [Nat07] S. 39). Anzuführen sind hier zum Beispiel die jeweiligen Sperrbildschirme samt Statusleisten am oberen Bildschirmrand von Google Android (links) und Apple iOS (rechts), die sich im Aufbau maßgeblich unterscheiden (siehe Abbildung 13):

Abbildung 13: Sperrbildschirm von Android und iOS

Anhand des Produktionsjahres des Smartphones kann darüber hinaus eine Eingrenzung auf potentiell installierte Versionen vorgenommen werden. Eine konkrete Bestimmung der Betriebssystemversion ist bei Android lediglich über die Sichtung der Geräteinformationen im Bereich „Einstellungen“ möglich.

4.2.3 Stromversorgung

Nach der Modellbestimmung ist es essentiell, eine Stromversorgung einzurichten, um den Ladezustand des Akkus auf einem angemessen Niveau zu halten.

Dadurch soll vor allem bei eingeschalteten Smartphones eine Abschaltung und schließlich ein Verlust von flüchtigen und möglicherweise temporär zugreifbaren Daten verhindert werden. Ferner erfordern durch den Besitzer eingerichtete Schutzmaßnahmen auf dem Gerät, wie z. B. eine gesetzte PIN zum Schutz der SIM-Karte, bei jedem Systemstart eine Authentifizierung. Die Gewährleistung des fortlaufenden Betriebs soll die Eingabe der hierfür notwendigen Sicherheitscodes vermeiden und so ein Zugriff auf die Daten weitestgehend gewahrt bleiben. Eine Ausnahme stellt in diesem Zusammenhang ein vorhandener Sperrbildschirm dar. Dieser wird aus Sicherheitsgründen automatisch nach Ablauf einer vorgegebenen Zeitspanne aktiviert. Demnach wird der dafür verwendete Sicherheitscode nicht nur während des Systemstarts sondern immer nach einer ausbleibenden Interaktion mit dem Smartphone abgefragt.

Bei den heutigen Geräten erfolgt die Aufladung typischerweise über eine USB-Schnittstelle, welche ebenfalls für die Verbindung zu einer Workstation zum Datenaustausch genutzt wird. Um einen ungewollten Verbindungsaufbau zu vermeiden, sollte eine Aufladung über ein Ladegerät für die Steckdose erfolgen. Das Modell bestimmt hierbei die Wahl des passenden Ladegeräts. Eine zu hohe Ladeschlussspannung kann bei einer fehlenden Überwachungselektronik zur irreversiblen Beschädigung des Akkus führen. Dies hat zur Folge, dass das Smartphone nicht mehr ohne Stromversorgung betrieben werden kann und zum Anschluss an die forensische Workstation abzuschalten ist. Die ursprünglich zu schützenden flüchtigen und temporär zugreifbaren Daten wären somit verloren. Auch würde eine Abschaltung des Geräts eine erneute Authentifizierung bedingen. Nach einem erfolgreichen Verbindungsaufbau erhält das Gerät anschließend über die USB-Schnittstelle den notwendigen Strom für den Betrieb.

4.2.4 Abschirmung

Als nächstes ist eine Abschirmung des Smartphones erforderlich. Dies setzt, auch im ausgeschalteten Zustand des Geräts, die Trennung aller Kabel- und Funkverbindungen voraus. Dadurch soll eine Veränderung bestehender Daten im RAM, Flash-Speicher und dem Speicher der SIM-Karte zu jedem Zeitpunkt der Untersuchung ausgeschlossen werden. Ebenfalls verhindert eine unterbrochene Internetverbindung die Löschung der Daten und die Wiederherstellung der Werkseinstellungen durch Ausführung eines „Remote Wipes“.

Die Funktion wird von verschiedenen Herstellern, wie Google und Samsung, angeboten. Das übergeordnete Ziel besteht darin, bei Verlust oder Diebstahl des Smartphones die sensiblen Daten des Gerätebesitzers vor unberechtigten Zugriffen zu schützen.

Kabelverbindungen

Beim Beschlagnahmen des Geräts kann es vorkommen, dass dieses zur Synchronisation oder zum Aufladen über die USB-Schnittstelle mit einem Computer verbunden ist. Die damit einhergehende Datenübertragung gilt es, durch die Entfernung des Kabels vom Computer und dem Smartphone zu unterbrechen (vgl. [Nat07] S. 33). Es ist aufgrund der direkten Verbindung zwischen den Geräten naheliegend, dass der Computer eventuell Daten enthält, die im Zuge der forensischen Untersuchung von Bedeutung sind. Aus diesem Grund ist auch dieser sicherzustellen und entsprechend der Vorgehensweise zur klassischen IT-Forensik (z. B. des BSI) zu untersuchen. Ein besonderes Augenmerk sollte an dieser Stelle auf eine potentielle Synchronisationssoftware gelegt werden. Die Software generiert dedizierte Verzeichnisse für die transferierten Daten des Smartphones, welche für den Vorfall relevante Spuren enthalten können.

Funkverbindungen

Heutige Smartphones verfügen über zahlreiche Funkverbindungen, die für eine Datenübertragung genutzt werden. Zu den gängigsten Übertragungsverfahren heutiger Geräte gehören:

- Wireless Personal Area Networks (WPANs) wie Bluetooth für den Datentransfer über kurze Distanzen
- Wireless Local Area Networks (WLANs) gemäß des Standards IEEE 802.11 für den Datentransfer über große Distanzen
- Global Positioning System (GPS) zur Positionsbestimmung, Zeitmessung und Navigation nach ausgewählten Routen
- Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS) sowie das neu auf dem Markt verfügbare Long Term Evolution (LTE) für Telefonie, zum Versand sowie Empfang von Kurzmitteilungen und zur Anbindung an das mobile Internet

- Near Field Communication (NFC) zum kontaktlosen Datenaustausch über kurze Distanzen von wenigen Zentimetern

Die Abschirmung des Smartphones zur Trennung der aufgeführten Datenverbindungen kann software- und hardwarebasiert erfolgen. Eine maßgebliche Anforderung ist hierbei, dass das Gerät gegenüber allen aufgeführten Funkarten abgeschirmt wird. Sowohl das NIST als auch Andrew Hoog führen in diesem Zusammenhang Methoden auf, die Vor- und Nachteile in Bezug auf die IT-Forensik haben (vgl. [Nat07] S. 34).

Ein erster Ansatz besteht in der Abschaltung eines eingeschalteten Smartphones. Dies hätte den Verlust von flüchtigen sowie temporär zugreifbaren Daten und bei Einschaltung die Eingabe sämtlicher eingerichteter Sicherheitscodes zur Authentifizierung zur Folge. Darüber hinaus wird für die weitere Untersuchung in der Regel ein lauffähiges Gerät benötigt. Android bietet stattdessen die Möglichkeit einen „Offline-Modus“ auszuwählen, wodurch lediglich die Kommunikationsfunktionen des Smartphones deaktiviert werden. Zusätzlich verringert sich dadurch der Energieverbrauch. Typischerweise kann die Option durch ein längeres Drücken der Taste zum Einschalten des Smartphone aufgerufen werden. Die Entsperrung eines möglicherweise aktivierten Sperrbildschirms ist nicht notwendig. Die softwarebasierte Methode erfordert lediglich eine Interaktion mit dem eingeschalteten Gerät, welche prinzipiell aus Gründen der Datenintegrität vermieden werden sollte.

Um eine Interaktion mit dem Smartphone weitestgehend auszuschließen, kann alternativ ein metallisches Transportbehältnis zur Abschirmung verwendet werden. Mögliche Ausgaben des Geräts lassen hierbei auf eine erfolgreiche Unterbrechung bestehender Verbindungen schließen. In der Regel erhöht die hardwarebasierte Methode die Sendeleistung, woraus ein stärkerer Energieverbrauch resultiert. Die Anbringung einer Stromversorgung ist somit unerlässlich. Nicht isolierte Kabelverbindungen können jedoch die Wirkung der Schirmung erheblich beeinträchtigen. Aufgrund dessen sollte ein solches Transportbehältnis mit Bedacht und möglichst nur für den Transport gewählt werden. Die weitere forensische Untersuchung ist nach der Überführung der beschlagnahmten Gegenstände in einem abgeschirmten Arbeitsbereich durchzuführen.

4.3 Datensammlung

Die Datensammlung umfasst neben dem **Verbindungsaufbau** die Prozessbausteine **Sicherung RAM-Speicher**, **Sicherung Flash-Speicher (intern)**, **Sicherung Flash-Speicher (extern)** und **Sicherung SIM-Karte.** Die darin zu ergreifenden Untersuchungsschritte und Methoden sind weitestgehend abhängig vom zu Grunde liegenden Betriebssystem des Smartphones und beziehen sich in dieser Ausarbeitung ausschließlich auf die technischen Gegebenheiten von Android. Eine Reihenfolge zur Durchführung der Datensammlung, im Speziellen zur zeitlichen Anordnung der Prozessbausteine, wurde bereits im zuvor entwickelten Prozessmodell aufgezeigt. An dieser Stelle wird ein besonderes Augenmerk auf die Umsetzung innerhalb der Prozessbausteine gelegt, welche maßgeblich von den in Android verfügbaren Schutzmaßnahmen und dem aktuellen Betriebszustand des Smartphones (ein- oder ausgeschaltet) abhängt.

4.3.1 Verbindungsaufbau

Vor dem Verbindungsaufbau ist zwischen der forensischen Workstation und dem Smartphone ein hardwarebasierter Writeblocker anzubringen. Hierbei ist zu berücksichtigen, dass die Systeme ausschließlich einen Schutz gewährleisten, wenn das Smartphone als Massenspeicher mit der Workstation verbunden wird [Rei10]. Eine Filterung von schreibenden Operationen, welche – z. B. unter Einsatz der ADB – direkt über das Smartphone ausgeführt werden, ist folglich nicht möglich ist. Demnach besteht das Risiko einer Integritätsverletzung. Die Untersuchungsschritte sind deshalb stets mit Bedacht auszuführen. Ferner empfiehlt sich die Aufzeichnung des Kommunikationsverlaufs zwischen forensischer Workstation und dem Smartphone zur Dokumentation möglicher Schreibvorgänge. Hierfür eignet sich das in der strategischen Vorbereitung installierte Werkzeug zum Port-Monitoring und der im Betriebssystem des Smartphones intergierte Protokollierungsdienst Logcat.

Der Verbindungsaufbau zum Smartphone zur Datensammlung sollte stets über die bereitgestellte USB-Schnittstelle erfolgen. Eine Grundvoraussetzung in Android ist hierbei die Aktivierung des USB-Debuggings, das schließlich auch die Verwendung der Android Debug Bridge ermöglicht (siehe Kapitel 2.2.3). Von einer Entfernung der Abschirmung und einer Aktivierung verschiedener Funkverbindungen auf dem Gerät zur Kommunikation mit der forensischen Workstation ist zum Schutz der zu untersuchenden Daten abzusehen.

Wie bereits beschrieben birgt eine bestehende Funkverbindung unter anderem die Gefahr eines Remote Wipes, den es zu verhindern gilt. Durch den in Android implementierten Sperrbildschirm und das Berechtigungssystem kann der Verbindungsaufbau über die USB-Schnittstelle allerdings maßgeblich beeinträchtigt werden.

Sperrbildschirm

Ein aktivierter Sperrbildschirm verhindert die Interaktion mit dem eingeschalteten Smartphone bis zur Entsperrung. Ohne eine Authentifizierung kann somit das USB-Debugging in den Entwickleroptionen nicht aktiviert werden, woraus erhebliche Beeinträchtigungen in der nachfolgenden Datensammlung resultieren. Eine Ausnahme stellt in diesem Kontext ein bereits aktiviertes USB-Debugging dar. In diesem Fall sollte das Smartphone, welches mit der forensischen Workstation verbunden ist, nach Absetzen des Befehls `adb devices` in der geöffneten Shell aufgelistet werden. Infolgedessen kann auch ohne Authentifizierung am Gerät ein Zugriff auf das Smartphone über die USB-Schnittstelle erfolgen. Da dies ein enormes Sicherheitsrisiko darstellt, ist der Modus jedoch standardmäßig deaktiviert und ein Zugang zu den Entwickleroptionen erforderlich. Fehlt der entsprechende Sicherheitscode, sind zur Aufhebung der Sperre zusätzliche Schritte einzuleiten. Während das NIST diesbezüglich allgemeine Methoden aufführt, widmet sich Andrew Hoog explizit den Gegebenheiten von Android (vgl. [Hoo11] S. 203-211 und [Nat07] S. 48-53). Im Zuge dieser Ausarbeitung konnten, angelehnt an die Vorschläge von Andrew Hoog, die folgenden weitestgehend modellunabhängigen Methoden ermittelt und unter forensischen Aspekten bewertet werden.

- **Mobile Device Management:** Der Sperrbildschirm kann über eine vorhandene Mobile Device Management-Software aufgehoben werden. Hierzu wählt der Administrator das entsprechende Smartphone aus und fordert eine Zurücksetzung des Sicherheitscodes an. Beim Zurücksetzen wird eine temporäre PIN generiert, mit dieser sich der Administrator am Gerät authentifizieren muss. Nach erfolgreicher Authentifizierung ist ein neuer Sicherheitscode in Form einer PIN oder eines Passworts zu setzen. Der Mechanismus funktioniert bei Smartphones, die über eine PIN, ein Passwort oder ein zu zeichnendes Muster gesperrt werden. Für das Zurücksetzen wird allerdings eine Verbindung zum Mobile Device Management benötigt. Infolgedessen ist für diesen Zeitraum auf eine Abschirmung des Smartphones zu verzichten, was aus forensischer Sicht in Bezug auf die Integrität der Daten ein enormes Risiko darstellt.

- **Smudge Attack (vgl. [Hoo11] S. 207):** Entwickelt wurde Smudge Attack von Wissenschaftlern der Universität von Pennsylvania (vgl. [AGM+10] S. 1). Die Grundlage der Methode bilden die in der Bestandsaufnahme angefertigten Fotos der Fettrückstände auf dem Touchscreen des Smartphones. Die Fotos sind anschließend mit einer Bildbearbeitungssoftware auszuwerten. Anhand der nicht reflektierenden Fettrückstände erfolgt die Rekonstruktion des zu zeichnenden Musters. Auf diese Weise kann das Muster auf beliebigen Geräten mit einer Erfolgsrate von 68% vollständig wiederhergestellt werden (vgl. [AGM+10] S. 5). Unter forensischen Gesichtspunkten ist Smudge Attack besonders empfehlenswert, da die Interaktion mit dem Smartphone bei einem korrekt ermittelten Muster auf ein Mindestmaß beschränkt wird und die eingerichtete Abschirmung von der Methode unbeeinflusst bleibt.
- **Google-Benutzerkonto (vgl. [Hoo11] S. 217):** Üblicherweise sind auf Android basierende und von Google lizenzierte Smartphones bei Google Play registriert. Mit Hilfe des zugehörigen Benutzerkontos ist es möglich, den Sperrbildschirm, der ein zu zeichnendes Muster als Sicherheitscode erwartet, aufzuheben. Die entsprechende Option erscheint nach einer mehrfachen Falscheingabe des Musters (z. B. nach fünf Versuchen beim Google Nexus 4). Infolgedessen sind die Anmeldedaten des Google-Benutzerkontos einzugeben. Heutige Smartphones benötigten, anders als dies Andrew Hoog im Jahr 2011 in seinem Buch beschrieb (vgl. [Hoo11] S. 210), zum Abgleich der Anmeldedaten eine Verbindung zum Google Server. Folglich ist auch in dieser Methode vorübergehend von einer Abschirmung abzusehen, was schließlich zu einer Verletzung der Integrität der auf dem Smartphone vorliegenden Daten führen kann.
- **Screen Lock Bypass (vgl. [Hoo11] S. 209):** Thomas Cannon, Leiter der Forschungs- und Entwicklungsabteilung von viaForensics, stellt in Google Play die Applikation „Screen Lock Bypass Pro“ bereit. Im Kern wartet die Applikation auf den Empfang eines vom Entwickler festgelegten Broadcasts, der durch eine dedizierte Aktion am Gerät ausgelöst wird. In der aktuellen Version ist demnach das Ladegerät anzuschließen [Can13]. Empfängt die Applikation den damit einhergehenden Broadcast, kann sie über eine von Android implementierte API den Sperrbildschirm aufheben. Die API wird benötigt, um z. B. einen eingehenden Anruf ohne die vorherige Aufhebung des Sperrbildschirms entgegen zu nehmen.

Die Installation der Applikation auf dem Smartphone wird über das Internetportal von Google Play unter Verwendung eines Computers gestartet. Daher muss auch in dieser Methode ein Google-Benutzerkonto vorhanden sowie bekannt sein und das Gerät zum Erhalt der Applikation eine Verbindung zum Google Server aufbauen. Ferner werden während der Installation Daten im RAM und im internen Flash-Speicher verändert. Aufgrund dieser Gegebenheiten sollte die Methode in einer forensischen Untersuchung erst nach Ausschluss der Lösungen zuvor in Erwägung gezogen werden.

- **Cellebrite UFED:** Cellebrite bietet mit seinem Gerät „UFED“, welches zur Datensammlung eingesetzt wird, unter Android die Möglichkeit, einen aktiven Sicherheitscode in Form einer PIN, eines Passworts oder eines zu zeichnenden Musters aufzuheben [Cel13c]. Die Methode funktioniert zum aktuellen Zeitpunkt nur auf vereinzelten Modellen von Samsung und erfordert einen Neustart des Smartphones. Besonders der Neustart ist im Hinblick auf den Verlust der Daten im RAM von Nachteil. Trotz dieser nicht zu vernachlässigenden Einschränkungen sind die Entwicklungen auf diesem Gebiet bezüglich Cellebrite zu beobachten. Neben einer Ergänzung des Portfolios um weitere Smartphones ist auch eine Überarbeitung der bestehenden Techniken nicht ausgeschlossen.

Im ungünstigsten Fall kann es vorkommen, dass keine der aufgeführten Methoden die aktive Sperre aufhebt. Unter diesen Umständen ist eine Datensammlung im normalen Betrieb des Smartphones ausgeschlossen. Dies hat erhebliche Auswirkungen auf die Sicherung des RAMs, des internen sowie externen Flash-Speichers und des Speichers der SIM-Karte, die in den nachfolgenden Kapiteln aufgegriffen werden.

Berechtigungen

In Kapitel 2.2.2 „Sicherheit“ wurde bereits erläutert, dass Android ein Berechtigungssystem implementiert, welches den Zugriff auf die Ressourcen des Smartphones benutzerbasiert reguliert. Auch für die Verbindung zum Gerät wird ein Benutzer mit eingeschränkten Zugriffsrechten verwendet, wodurch der Handlungsspielraum auf dem Smartphone erheblich begrenzt ist. Besonders die Durchführung einer umfassenden Datensammlung erfordert in vielen Fällen einen vollständigen Systemzugriff. Ausschließlich der Benutzer `root` besitzt die hierfür notwendigen Rechte.

Seitens der Hersteller ist eine Verwendung des Benutzers nicht erwünscht. Das Smartphone muss zu diesem Zweck einem Vorgang unterzogen werden, der in Android als „Rooten“ bezeichnet wird.

Beim Rooten werden vorhandene Schwachstellen im System ausgenutzt, um anschließend mit privilegierten Rechten auf dem Smartphone zu agieren. Die hierfür erforderlichen Methoden sind prinzipiell vom zu Grunde liegenden Modell abhängig und setzen ein aktiviertes USB-Debugging voraus. Vor der Ergreifung möglicher Methoden ist zu überprüfen, ob das zu untersuchende Smartphone bereits einem Root-Vorgang unterzogen wurde. Infolgedessen sollte der Befehl `adb shell su` statt einer Fehlernachricht „Permission denied“ eine Shell mit privilegierten Rechten zurückgeben. Der Parameter `su` bewirkt hierbei den gewünschten Benutzerwechsel zu `root`. Fällt der Test negativ aus, bietet sich zur Suche einer spezifischen Lösung das Forum „XDA Developers“ an (vgl. [Hoo11] S. 272). XDA Developers besteht aus einer großen Community von Softwareentwicklern im mobilen Bereich und widmet sich technischen Themen rund um Android.

Vor der Umsetzung einer ausgewählten Methode sind die damit verbundenen Manipulationen am Gerät in einer Testumgebung zu analysieren, um einem Verlust und einer Veränderung forensisch relevanter Daten vorzubeugen. So ist zur Erhaltung der Daten im RAM darauf zu achten, dass das Rooten keinen Systemneustart erfordert. Auch sollte der Vorgang nicht zu einer Wiederherstellung der Werkseinstellungen führen, was eine Löschung aller benutzerspezifischen Daten zur Folge hätte. Besonders empfehlenswert im forensischen Bereich sind temporäre Lösungen, da diese die zuvor aufgeführten Anforderungen erfüllen und das Gerät nach einem Systemneustart in den Ursprungszustand zurückversetzen.

4.3.2 Sicherung RAM-Speicher

Die Sicherung des RAMs erfordert, dass die Interaktion mit dem Smartphone auf ein Mindestmaß begrenzt wird und kein Systemneustart erfolgt. Dieser würde letztlich zu einem Verlust der im Speicher vorliegenden Daten führen. Die Einhaltung der Anforderungen ist im Bereich der forensischen Untersuchung von Smartphones zum aktuellen Zeitpunkt und im Hinblick auf die in Jelly Bean implementierten sicherheitstechnischen Restriktionen nur schwer umzusetzen. Viele der Methoden bedingen die Modifikation des Geräts und infolgedessen den unerwünschten Neustart.

Während der Untersuchungen konnten für Smartphones basierend auf Jelly Bean zwei potentielle Methoden erarbeitet werden. Beide verwenden zur Umsetzung die ADB und setzen somit ein aktiviertes USB-Debugging und privilegierte Rechte voraus. Ist dies nicht gegeben, kann keine Sicherung des RAMs durchgeführt werden. Darüber hinaus ist zu berücksichtigen, dass die Komplexität und folglich der Aufwand zur Umsetzung der zweiten Methode deutlich höher ist. Bevorzugt sollte daher in einer forensischen Untersuchung mit der ersten Variante begonnen und bei Misserfolg auf die zweite zurückgegriffen werden.

Methode (1)

Eine Sicherung des RAM-Speichers kann, wie auch Björn Roos in seiner Masterarbeit aufgreift, unter Verwendung des Befehls `kill` in einer über die ADB geöffneten Shell erfolgen (vgl. [Roo11] S. 100-101). Der Befehl sendet mit Hilfe eines „Process Identifier“, kurz PID, Signale an einen laufenden Prozess, welche diesen und die damit in Verbindung stehende Applikation beenden sollen. Entsprechend der vorgegebenen Definition setzt sich `kill` aus folgenden Parametern zusammen: `kill -Signal PID`. Damit ein „Dump“, d.h. ein Auszug des Speicherinhalts erzeugt wird, ist als Signal der Wert `10` bzw. die Zeichenkette `SIGUSR1` anzugeben. Die PID kann mittels des Befehls `ps`, der eine Auflistung aller aktiven Prozesse bewirkt, ermittelt werden.

Sämtliche im Binärformat generierten Dumps besitzen das Format `heap-dump-tm<timestamp>-pid<PID>.hprof` und befinden sich unter Android im Verzeichnis `/data/misc`. Damit eine Speicherung möglich ist, benötigt der zur Ausführung des Befehls verwendete Benutzer Schreibrechte auf dem Verzeichnis. Diese sind standardmäßig nicht vorhanden und können ausschließlich durch den privilegierten Benutzer `root` des Systems mit `chmod 777 /data/misc` erteilt werden. Einige Smartphones erstellen, unabhängig von der zu Grunde liegenden Betriebssystemversion, trotz eingerichteter Rechte keine Dumps. Darunter fallen unter anderem die in dieser Ausarbeitung untersuchten Modelle „Nexus 4“ und „Nexus S“ von Google. Zurückzuführen ist dieser Umstand auf einen im Dezember 2010 bereitgestellten Patch, den Gerätehersteller zur Deaktivierung der Funktion in der Dalvik Virtual Machine nutzen können [Ber11]. Sehr wahrscheinlich handelt es sich hierbei um eine Schutzmaßnahme, die ein Auslesen sensibler Daten aus dem RAM auf die zuvor beschriebene Art und Weise durch Unbefugte verhindern soll.

Methode (2)

Eine weitere Möglichkeit besteht in der Ergänzung des in Android bereitgestellten Linux Kernels um ein zusätzliches Modul namens „Linux Memory Extractor“ (LiME). LiME wurde 2010 auf der Konferenz „ShmooCon“ veröffentlicht und erlaubt als erstes Werkzeug die vollständige Sicherung des RAMs von Smartphones basierend auf Android [Syl12]. Demnach werden nicht nur, wie in der ersten Methode dargestellt, Auszüge von konkret ausgewählten Applikationen gesichert. Im Gegenzug erfordert LiME jedoch eine aufwendige Entwicklung des Kernelmoduls für ein spezifisches Smartphone. Die detaillierte Anleitung hierfür kann unter [Syl13] eingesehen werden. Im Wesentlichen sind zu Beginn der Kernel und die Konfigurationsdatei des Geräts zu laden. Anschließend ist dieser unter Einbindung der Konfigurationsdatei mit LiME über einen Cross-Compiler zu kompilieren. Daraus resultiert schließlich das gerätespezifische Kernelmodul mit der Endung `.ko`, welches mit Hilfe der ADB auf das Gerät transferiert und unter Verwendung des Befehls `insmod` in den bestehenden Kernel eingebunden wird. Zur Ausführung sind ebenfalls privilegierte Rechte auf dem Smartphone erforderlich.

Eventuell kann es vorkommen, dass der Kernel des Smartphones eine Einbindung zusätzlicher Module unterbindet. Zur Aufhebung dieser Einschränkung ist es notwendig, den werksseitigen Kernel durch eine modifizierte Version auszutauschen. Der damit einhergehende Neustart des Smartphones hätte einen Verlust der aktuellen Speicherinhalte zur Folge, weshalb eine solche Maßnahme ausgeschlossen ist. Ist eine Einbindung des Kernels möglich, dann kann eine Sicherung des RAMs mit LiME entweder über TCP auf die forensische Workstation oder direkt auf den Flash-Speicher des Smartphones erfolgen (vgl. [Syl13] S. 6). Die Variante über TCP nutzt zum Datenaustausch einen frei ausgewählten Port und als Übertragungskanal die bestehende USB-Verbindung zur forensischen Workstation. Dadurch können jedoch Inhalte im Netzwerkpuffer des Smartphones überschrieben werden, weshalb der Entwickler des Werkzeugs als Speicherort vorzugweise die Verwendung eines gesonderten externen Flash-Speichers empfiehlt (vgl. [Syl13] S. 7). In einer geöffneten ADB-Shell ist zur Einrichtung der jeweiligen Sicherungsart einer der beiden beispielhaft aufgeführten Befehle abzusetzen:

```
# insmod lime.ko "path=tcp:4444 format=lime"
```

```
# insmod lime.ko "path=/sdcard/RAM.lime format=lime"
```

Abbildung 14: Sicherung des RAMs durch Einsatz von Linux Memory Extractor

Der Parameter `path` steht hierbei für das Übertragungsziel der Sicherung und der Parameter `format` für das Format der zu erzeugenden Datei. Als Formattyp ist `lime` zu präferieren, da in dieser Konfiguration jeder Speicherbereich einen eigenen Header mit Informationen zum zugehörigen physikalischen Adressraum enthält (vgl. [Syl13] S. 6). Weitere Formate können in der Dokumentation des Entwicklers eingesehen werden (siehe [Syl13] S. 6).

4.3.3 Sicherung Flash-Speicher (intern)

Der interne Flash-Speicher stellt innerhalb der Datensammlung die zentrale Komponente dar. Zurückzuführen ist dies auf seine persistente Eigenschaft, hohen Speicherkapazitäten und Rolle als primäres Speichermedium im Smartphone. Demzufolge befinden sich in Bezug auf IT-Forensik nicht nur die meisten, sondern auch wichtigsten Daten auf dem internen Flash-Speicher. Die Durchführung der Sicherung kann grundsätzlich auf zwei Arten erfolgen: hardwarebasiert und softwarebasiert.

Hardwarebasiert

Die hardwarebasierte Datensammlung sieht die Sicherung des Flash-Speichers ohne die Verwendung von bestehenden Betriebssystemschnittstellen vor. Stattdessen wird direkt auf die Hardware zugegriffen und infolgedessen die Abschaltung des Smartphones notwendig. Zwar stellt demgemäß ein aktiver Sperrbildschirm kein Hindernis mehr da, jedoch setzt die hardwarebasierte Sicherung ein unverschlüsseltes internes Speichermedium voraus. Allenfalls sind die gesicherten Daten in der weiteren Untersuchung nicht lesbar. Die Umsetzung erfolgt entweder durch eine Extraktion des Speichers oder über den Standard „Joint Test Action Group“ (JTAG) (vgl. [Hoo11] S. 268 und [Nat07] S. 46). Nachfolgend wird auf die grundlegenden Funktionsweisen der Methoden, angelehnt an die Beschreibungen von Andrew Hoog, näher eingegangen (vgl. S. 268-270).

Bei JTAG handelt es sich um einen Standard des Institute of Electrical and Electronics Engineers, kurz IEEE.

Der Standard beschreibt die Verfahrensweise zum Testen integrierter Schaltungen auf Leiterplatinen. Die Leiterplatine des Geräts verfügt hierzu über „Test Access Ports“ (TAPs), die einen Zugriff auf die Central Processing Unit (CPU) erlauben. Ein spezielles Gerät (z. B. eine „Flasher Box[2]“), welches mit den TAPs und der forensischen Workstation zu verbinden ist, fordert über die CPU schließlich die Daten an und erzeugt daraus ein vollständiges 1:1-Abbild des internen Flash-Speichers. Eine mitgelieferte und auf der Workstation zu installierende Software dient hierbei zur Steuerung der Box. Besonders der Verbindungsaufbau zu den TAPs erweist sich als äußerst schwierig, da sich diese typischerweise in Form von kleinen, nicht beschrifteten Pins auf der Leiterplatine befinden. Daraus resultiert, neben einer zeitintensiven Suche der Pins, eine aufwändige Bestimmung der Belegungen. Für den Betrieb der Leiterplatine muss darüber hinaus eine passende Versorgungsspannung ermittelt werden. Die Datensammlung über JTAG ist demnach äußerst zeitintensiv und sollte daher in einer forensischen Untersuchung als Letztes in Erwägung gezogen werden.

Im Zuge der Extraktion wird der auf der Hauptplatine fest verdrahtete Flash-Speicher herausgelötet (vgl. [Hoo11] S. 270). Beim Löten kann es zu einer Beschädigung der leitenden Kontakte unterhalb des Speichers kommen. Diese Kontakte müssen für eine erfolgreiche Datenübertragung in einem nachfolgenden Schritt wiederhergestellt werden. Anschließend ist es möglich, den extrahierten Speicher unter Verwendung eines speziell programmierten Kartenlesers auszulesen und ein vollständiges 1:1-Abbild zu erzeugen. Durch die Anbringung eines Writeblockers zwischen Kartenleser und forensischer Workstation können hierbei Schreibzugriffe unterbunden werden. Während der physischen Extraktion besteht jedoch, vor allem im Zuge des Lötvorgangs und der damit einhergehenden hohen Temperaturen, das Risiko einer Zerstörung des Speichers und somit eines Datenverlusts. Aufgrund dieses Risikos und der irreparablen Schäden am Smartphone ist die Methode allenfalls an defekten Geräten auszuführen, die keine andere Form der Sicherung zulassen.

Beide aufgeführten Methoden erzeugen ein vollständiges physikalisches 1:1-Abbild des internen Flash-Speichers.

[2] Eine Flasher Box erlaubt das Lesen und Überschreiben der Inhalte eines Flash-Speichers. Im Smartphone-Umfeld setzten Hersteller die Geräte zur Fehlersuche, Reparatur und Aktualisierung ihrer Produkte ein.

Demgemäß umfasst das Abbild auch mögliche nicht allokierte Bereiche, die zur Wiederherstellung von z. B. gelöschten Dateien erforderlich sind. Auch stellen der Sperrbildschirm und die Berechtigungen von Android kein Hindernis dar. Dem gegenüber steht jedoch ein erheblicher technischer und zeitlicher Aufwand, verbunden mit der Gefahr eines vollständigen Datenverlusts durch die Zerstörung der Hardware. Darüber hinaus erfordern die Methoden die Abschaltung eines laufenden Smartphones, wodurch bei einem erneuten Systemstart sämtliche eingerichteten Sicherheitscodes einzugeben sind. Letztlich werden neben informationstechnischen sowie forensischen Fertigkeiten tiefgehende Kenntnisse aus dem Bereich der Elektrotechnik benötigt. Die softwarebasierten sollten daher den hardwarebasierten Methoden vorgezogen werden.

Softwarebasiert

Die softwarebasierte Datensammlung verwendet zur Anfertigung einer Sicherung des internen Flash-Speichers vorhandene Betriebssystemschnittstellen. Folglich muss das Smartphone während des Sicherungsvorgangs im eingeschalteten Zustand vorliegen. Wie in Kapitel 2.3.1 „Anforderungen“ angeführt, kann aufgrund dieser Gegebenheiten die Veränderung von bestehenden Daten auf dem Gerät nicht ausgeschlossen werden. Um das Risiko einer Integritätsverletzung zu verringern, sind die softwarebasierten Methoden gegeneinander abzuwägen und stets diejenige mit dem geringsten Einfluss auf die Daten zu wählen.

Die Methoden zur Sicherung des internen Flash-Speichers hängen prinzipiell von den Eigenschaften des Geräts ab, welches in der Bestandsaufnahme der operationalen Vorbereitung sichergestellt wurde. So entscheiden prinzipiell der Betriebszustand (ein- oder ausgeschaltet) und aktivierte Schutzmaßnahmen (z. B. ein aktivierter Sperrbildschirm, eingeschränkte Berechtigungen und eine Verschlüsselung des Speichers) über die Art der Durchführung sowie die Vollständigkeit der Sicherung. Aufgrund der zahlreichen softwarebasierten Möglichkeiten und der herausragenden Bedeutung des internen Flash-Speichers innerhalb der Datensammlung erfolgt in Kapitel 5 eine weiterführende Betrachtung.

4.3.4 Sicherung Flash-Speicher (extern)

Die Art der Sicherung des externen Flash-Speichers hängt von den aktivierten Schutzmaßnahmen in Android 4.2 ab. In der aktuellen Betriebssystemversion ist die vollständige Verschlüsselung des externen Speichermediums möglich.

Ein erstes Indiz für das Vorliegen einer solchen Schutzmaßnahme ist ein Sicherheitscode in Form einer PIN oder eines Passworts. Absolute Gewissheit kann durch einen Blick in die Einstellungen unter der Rubrik „Sicherheit“ des Smartphones erlangt werden. Eine aktive Verschlüsselung hat enorme Auswirkungen auf die Durchführung der Datensammlung, weshalb zwischen einem verschlüsselten und unverschlüsselten Fall zu unterscheiden ist.

Verschlüsselter Speicher

Ein verschlüsselter externer Flash-Speicher kann in Android ausschließlich über das Smartphone ausgelesen werden und ist daher im Smartphone zu belassen. Zurückzuführen ist dies auf den zur Entschlüsselung erforderlichen Schlüssel, der auf dem Gerät gespeichert wird. Die Sicherung muss demnach gemeinsam mit dem internen Flash-Speicher unter Verwendung der vorhandenen Betriebssystemschnittstellen erfolgen (siehe Kapitel 5). Hierbei kann, anders als beim internen Flash-Speicher, aufgrund des Dateisystems FAT32 und der fehlenden Unterstützung von Benutzerrechten, in der Regel uneingeschränkt lesend sowie schreibend auf die Daten des externen Speichers zugegriffen werden. Da die Entschlüsselung bereits beim Systemstart erfolgt, sind die Daten bei einem im laufenden Betrieb beschlagnahmten Smartphone ohne eine erneute Authentifizierung zugänglich. Ein aktiviertes USB-Debugging ist hierbei jedoch zwingend erforderlich. Ist dies nicht gegeben, muss ein vorhandener Sperrbildschirm aufgehoben und in den Einstellungen eine Aktivierung des USB-Debuggings vorgenommen werden. Liegt der entsprechende Sicherheitscode, welcher auch zur Entschlüsselung verwendet wird, nicht vor, dann sind die im Prozessbaustein **Verbindungsaufbau** aufgeführten Methoden zur Aufhebung des Sperrbildschirms anzuwenden. Ferner ist unter diesen Voraussetzungen ein Systemneustart, der eine erneute Entschlüsselung erfordern würde, prinzipiell zu vermeiden. Befindet sich das Smartphone während der Beschlagnahmung bereits im ausgeschalteten Zustand und kann der Sicherheitscode zur Entschlüsselung nicht ermittelt werden, so sind die Daten auf dem externen Flash-Speicher in einer forensischen Untersuchung unbrauchbar.

Unverschlüsselter Speicher

Ein unverschlüsselter externer Flash-Speicher sollte aus dem Smartphone ausgebaut werden. Befindet sich das Speichermedium in einer Vorrichtung unter dem Akku, ist für die Entnahme die Abschaltung des Smartphones erforderlich.

Die Sicherung ist anschließend mit Hilfe eines an die forensische Workstation angeschlossenen Kartenlesers durchzuführen. Hierzu muss der externe Speicher für gewöhnlich in einen geeigneten Adapter eingesteckt werden. Zur Gewährleistung der Integrität sollte der Adapter einen Schreibschutz besitzen und ein kryptografischer Hash, z. B. mit dem standardmäßig in Linux-Systemen integrierten Werkzeug „sha1sum“, über die Daten des Speichermediums gebildet werden. Über die forensische Workstation kann anschließend unter Verwendung des Werkzeugs „dd“ das Speichermedium ausgelesen und ein vollständiges physikalisches 1:1-Abbild, welches auch nicht allokierte Speicherbereiche einschließt, erzeugt werden. Das Werkzeug ist im Bereich der IT-Forensik weit verbreitetet und standardmäßig in Systemen basierend auf Linux zum Konvertieren sowie Kopieren von Dateien integriert. Die Sicherung eines externen Flash-Speichers mit 16 Gigabyte durch Einsatz des Befehls `dd` sieht beispielhaft wie folgt aus (siehe Abbildung 15):

```
# dd if=/dev/mmcblk0p1 of=/home/dmuth/FlashExtern.dd

30670848+0 Datensätze ein

30670848+0 Datensätze aus

15703474176 Bytes (16 GB) kopiert, 3068,8 s, 5,1 MB/s
```

Abbildung 15: Sicherung des externen Flash-Speichers durch Verwendung von dd

Der Parameter `if` (Inpute File) steht für die Quelle und enthält den Pfad zum Flash-Speicher im Kartenleser. Für den Zugriff auf `/dev/mmmcblk0p1` sind privilegierte Rechte auf dem IT-System erforderlich. Der Speicherort der Sicherung wird mit dem Parameter `of` (Output File) übergeben. Abschließend ist nochmals ein Hash über die Daten der angefertigten Sicherung zu bilden und mit dem zu Beginn berechneten Wert zu vergleichen.

4.3.5 Sicherung SIM-Karte

Die SIM-Karte ist standardmäßig mit einer PIN geschützt und muss beim Start des Smartphones durch den Besitzer freigeschaltet werden. Die Aktivierung der Schutzmaßnahme kann in den Einstellungen unter der Rubrik „Sicherheit“ des Smartphones eingesehen werden. Daneben erlauben verschiedene Geräte das Auslesen des PIN- und PUK-Status.

Hierbei sind jedoch ausschließlich Lösungen zu verwenden, die an die USB-Schnittstelle des Smartphones angeschlossen werden und somit keine vorzeitige Entnahme der SIM-Karte erfordern. Diese Bedingung erfüllt derzeit lediglich das UFED von Cellebrite. Da eine vorhandene PIN-Eingabe maßgeblich das Vorgehen zur Datensammlung beeinflusst, ist es notwendig, zwischen einer geschützten und ungeschützten SIM-Karte zu unterscheiden.

Geschützte SIM-Karte

Die Daten einer geschützten SIM-Karte sind bei einer unbekannten PIN und einem ausgeschalteten Smartphone nur mit Hilfe der PUK zugänglich. Liegt das Gerät im eingeschalteten Zustand vor, kann davon ausgegangen werden, dass die Eingabe der PIN bereits durch den Besitzer erfolgte. Infolgedessen sind zumindest die Kontaktdaten sowie Kurzmitteilungen über das Betriebssystem des Geräts zugänglich. Von einer Entnahme der SIM-Karte wird daher, so lange PIN und PUK nicht vorliegen, abgeraten. Zum Erhalt der PUK und der Erhebung einer vollständigen Sicherung unter Einsatz eines Kartenlesers sollte angelehnt an die Empfehlung von Alexander Geschonneck eine Anfrage an den Provider gerichtet werden (siehe Anhang A). Bleibt dies aufgrund zu langer Wartezeiten erfolglos, können im eingeschalteten Betriebszustand des Smartphones die nachfolgenden Methoden ergriffen werden.

Der erste Ansatz sieht die Installation der modellspezifischen Software, die vom Hersteller zur Verwaltung des Smartphones zur Verfügung gestellt wird, auf der forensischen Workstation vor. Darüber lassen sich in den meisten Fällen die Kontakte und Kurzmitteilungen der SIM-Karte auslesen und exportieren. Die angebotenen Funktionalitäten hängen jedoch vom entsprechenden Hersteller ab. Beispielsweise liest die von Samsung bereitgestellte Software keine Kurzmitteilungen aus. Auch ist der Einsatz nicht forensischer Werkzeuge immer mit einem Risiko verbunden, da deren Wirkungsweise oftmals nur schwer nachzuvollziehen ist. Die Folge wären im ungünstigsten Fall Veränderungen an den Daten des Smartphones, die zu einer Minderung der Beweiskraft führen. Bei forensischen Lösungen ist wiederum darauf zu achten, dass die Daten tatsächlich von der SIM-Karte bezogen werden können und nicht auf die Daten des internen Flash-Speichers zurückgegriffen wird. Anzuführen wäre in diesem Kontext das UFED von Cellebrite.

Die alternative Methode besteht darin, die Applikationen zur Verwaltung der Kontakte und Kurzmitteilungen im Smartphone aufzurufen und die Daten der SIM-Karte zu importieren.

Beim Portieren werden die jeweiligen Datenbanken um die zusätzlichen Daten ergänzt. Die damit verbundenen Veränderungen sind in einer forensischen Untersuchung genau zu dokumentieren, um die Beweiskraft der potentiellen Spuren zu bewahren. Typischerweise werden in Android die Datenbanken der Kontakte unter `/data/data/com.android.providers.contacts/databases/contacts2.db` und Kurzmitteilungen unter `/data/data/com.android.providers.telephony/databases/mmssms.db` abgelegt. Mit Hilfe der ADB ist anschließend eine Übertragung der Datenbanken auf die forensische Workstation möglich. Der Zugriff auf das Verzeichnis `/data/data/` erfordert allerdings privilegierte Rechte auf dem Smartphone. Ferner setzt die Methode neben einem aktivierten USB-Debugging eine Interaktion mit dem Gerät voraus. Ein vorhandener Sperrbildschirm kann demzufolge auch in diesem Prozessbaustein ein Hindernis bei der Sicherung darstellen.

Besonders bei der zweiten Variante ist der Schutz der Daten durch den Writeblocker nicht gegeben, da die erforderlichen Schreiboperationen direkt über das Smartphone abgesetzt werden und folglich nicht den hardwarebasierten Writeblocker passieren. In diesem Fall wird vor allem der Einsatz des Protokollierungsdienstes Logcat empfohlen. Dieser befindet sich auf dem Smartphone und zeichnet somit sämtliche Interaktionen mit dem Gerät, d. h. auch die durchgeführten Schreiboperationen, auf. Dadurch lassen sich die erforderlichen Veränderungen an den Daten des Smartphones anschließend nachweisen.

Ungeschützte SIM-Karte

Eine SIM-Karte, deren PIN bzw. PUK bekannt oder die ungeschützt ist, kann ausgebaut und mit einem geringfügigen Aufwand über einen speziellen forensischen Kartenleser gesichert werden. Das NIST führt diesbezüglich einige, primär auf Windows basierende, Werkzeuge auf, die einen solchen Kartenleser mit zugehöriger Software bereitstellen (vgl. [Nat07] S. 17-18). Die verschiedenen Lösungen arbeiten hierbei, gemäß den Beschreibungen des NIST, nach dem gleichen Prinzip (vgl. [Nat07] S. 47-48). Ein direkter Zugriff auf die Daten der SIM-Karte wird durch integrierte Schutzmechanismen unterbunden. Die Sicherung der Elementary Files des Dateisystems erfolgt daher durch das Versenden von „Application Protocol Data Units“, kurz APDUs (siehe Kapitel 2.1.4).

APDUs sind einfache Kommunikationseinheiten, die Befehle zur Durchführung verschiedener Operationen (z. B. das Lesen von Daten) enthalten. Eindeutige nummerische Bezeichner ermöglichen es schließlich, jedes Elementary File über die APDUs zu adressieren und auszulesen. Ein maßgeblicher Unterschied der angebotenen Lösungen besteht in den Daten, die ermittelt werden. Während einige Werkzeuge ausschließlich die aus forensischer Sicht wichtigsten Daten beziehen, sichern andere wiederum vollständig die zu Grunde liegende SIM-Karte (vgl. [Nat07] S. 48). Die passende Lösung sollte demnach anhand des Vorfalls und der damit in Verbindung stehenden Daten ausgewählt werden.

4.4 Datenuntersuchung

Bevor eine Untersuchung beginnt, sollten zur Gewährleistung der Integrität Hashwerte über die Daten der erstellten Sicherungen berechnet werden. Von den Sicherungen sind wiederum Arbeitskopien zu erzeugen, für die ebenfalls vor und nach Abschluss der Datenuntersuchung ein Hashwert zu bilden ist. Mit der Übereinstimmung der Werte wird sichergestellt, dass im Zuge der Untersuchungen keine Veränderungen an den Daten vorgenommen wurden. Anschließend kann die Durchführung der Prozessbausteine **Physische Untersuchung** und **Logische Untersuchung** erfolgen. Währenddessen sind die für den Vorfall relevanten Spuren aus den Sicherungen des Smartphones zu extrahieren. Zur Beantwortung der kriminalistischen Fragestellungen wird in Android typischerweise nach den in Tabelle 1 aufgeführten Daten gesucht, die potentielle Spuren enthalten können (vgl. [Nat07] S. 59):

	Was?	**Wo?**	**Wann?**	**Wie?**	**Wer?**
Kontaktdaten					✓
Kurzmitteilungen	✓	✓	✓	✓	✓
E-Mails	✓	✓	✓	✓	✓
Termine	✓	✓	✓		✓
Anruflisten			✓		✓
Standortdaten		✓	✓		
Webbrowser-Inhalte	✓	✓	✓	✓	✓

Bilder	✓	✓	✓	✓	✓
Videos	✓	✓	✓	✓	✓
Dokumente	✓	✓	✓	✓	✓

Tabelle 1: Gegenüberstellung von Quellen für Spuren zu kriminalistischen Fragen

Nach welchen Spuren letztlich innerhalb der Datenuntersuchung gesucht wird, ist prinzipiell abhängig vom gegebenen Vorfall. In den folgenden Prozessbausteinen wird daher lediglich ein Überblick über die möglichen Untersuchungsschritte und Methoden gegeben. Dadurch soll ein grundlegender Eindruck vermittelt werden, an welchen Stellen in Android beispielhaft Spuren zu finden sind und wie diese aus den Sicherungen extrahiert werden können. Die Basis bildet hierbei eine Workstation mit Linux als Betriebssystem. Für eine weiterführende Betrachtung dieser Thematik empfiehlt sich das Buch von Andrew Hoog oder allgemeine Literatur zur Untersuchung von IT-Systemen basierend auf Ext4 und FAT32 wie von Eoghan Casey oder Brian Carrier (vgl. [Car05] S. 211-271, [Cas11] S. 551-585 und [Hoo11] S. 285-364).

4.4.1 Logische Untersuchung

Die logische Untersuchung erfolgt unter Verwendung der Dateisysteme der angefertigten physikalischen Sicherungen des internen und externen Flash-Speichers. Die Dateisysteme sind ausschließlich lesend in die forensische Workstation einzuhängen, wodurch mögliche Veränderungen der Daten während der Untersuchung vermieden werden sollen. Das Einhängen kann in Linux über eine Shell mit Hilfe des Befehls `mount` realisiert werden. Beispielhaft sieht dies wie folgt aus (siehe Abbildung 16):

```
# mount –o ro Partition.dd /Android/Dateisystem
```

Abbildung 16: Einhängen eines Dateisystems unter Android

Der Parameter `ro` (Read Only) gewährleistet den geforderten lesenden Zugriff auf das Dateisystem der Sicherung `Partition.dd`. Der Inhalt der Sicherung kann nach Absetzen des Befehls unter `/Android/Dateisystem`, z. B. über eine Shell, eingesehen werden.

Flash-Speicher (intern)

Im Hinblick auf den Flash-Speicher ist besonders die logische Untersuchung der Data-Partition von Bedeutung. Diese enthält die Datenbanken sämtlicher Applikationen auf dem Smartphone und stellt demzufolge eine wichtige Bezugsquelle für potentielle Spuren dar. Die nachfolgende Tabelle enthält einen Ausschnitt forensisch relevanter Datenbanken, die auf den meisten Smartphones basierend auf Android zu finden sind:

	Datenbank	Speicherort
Kontaktdaten	`contacts2.db`	`/data/data/com.android.providers.contacts/databases/contacts2.db`
Kurzmitteilungen	`mmssms.db`	`/data/data/com.android.providers.telephony/databases/mmssms.db`
E-Mails	`<Konto>.db`	`/data/data/com.google.android.gm/databases/<Konto>.db`
Termine	`calendar.db`	`/data/data/com.android.providers.calendar/databases/calendar.db`
Anruflisten	`logs.db`	`/data/data/com.sec.android.provider.logsprovider/databases/logs.db`
Standortdaten	`search_history.db`	`/data/data/com.google.android.apps.maps/databases/search_history.db`
	`da_destination_history.db`	`/data/data/com.google.android.apps.maps/databases/da_destination_history.db`
Webbrowser-Inhalte	`webview.db`	`/data/data/com.google.android.gm/databases/webview.db`
	`browser.db`	`/data/data/com.android.browser/databases/browser.db`

Tabelle 2: Forensisch relevante Datenbanken unter Android

Die aufgelisteten Datenbanken können beispielsweise mit dem Befehl `find –iname "Datenbankname.db"` gesucht und durch `dd` extrahiert werden. Anschließend ist es möglich, unter Einsatz der Werkzeuge „SQLite“ in einer Shell oder „SQLiteBrowser“ über eine grafische Oberfläche die Daten zu sichten und potentielle Spuren für den vorliegenden Vorfall zu ermitteln. Hierbei können auch gelöschte Einträge, deren Daten bis zum Überschreiben in einer Datenbank verbleiben, wiederhergestellt werden.

Neben den Datenbanken ist das Verzeichnis `/data/media` ein beliebter Speicherort für aufgenommene Bilder sowie Videos und verschiedene Dokumente, die wertvolle Schlussfolgerungen in einer forensischen Untersuchung zulassen. Die Ermittlung nach den Dateitypen kann zusätzlich um eine Durchsuchung der gesicherten Partitionen, wie System, Data und Cache, mit `find –iname "*.Dateityp"` ergänzt werden. Der Befehl erlaubt eine beliebige Erweiterung um zusätzliche Parameter, wie z. B. die Dateigröße, wodurch eine Begrenzung der Ergebnismenge erreicht wird. Ferner bietet sich zur Suche nach Zeichenketten innerhalb von Dateien der Befehl `grep` an. Denkbar ist auf diese Weise eine Ermittlung von Dateien, die ausgewählte Namen, Telefonnummern oder Adressen enthalten. Die extrahierten Dateien sind schließlich auf potentielle Spuren zu untersuchen.

Flash-Speicher (extern)

Der externe Flash-Speicher wird in Teilen von Android zur Auslagerung verschiedener Daten genutzt. So kann beispielsweise die Kamera-Applikation aufgenommene Bilder und Videos dort ablegen. Primär obliegt die Verwaltung des Speichers jedoch dem Besitzer des Smartphones. Daraus resultiert eine sehr dynamische Verzeichnisstruktur. Eine Auflistung typischer Speicherorte, die in einer logischen Untersuchung aufgesucht werden können, ist somit nicht möglich. Vielmehr ist mit den bereits aufgeführten Befehlen `find` und `grep` zu arbeiten. Dadurch können für den Vorfall relevante Bilder, Videos und Dokumente ermittelt werden, die typischerweise auf dem externen Speichermedium aufzufinden sind.

4.4.2 Physische Untersuchung

Die physische Untersuchung bezieht sich auf die Rohdateninhalte der angefertigten Sicherungen und wird folglich ohne Verwendung des Dateisystems durchgeführt. Auf diese Weise können bei Vorliegen eines vollständigen 1:1-Abbilds des internen und externen Flash-Speichers ebenfalls gelöschte Dateien wiederhergestellt werden.

Im Vergleich zum logischen Ansatz, in dem eine deutlich gezieltere Suche nach potentiellen Spuren durchführbar ist, kann die Datenmenge der physischen Methoden deutlich höher ausfallen. Die Wahl der Parameter zur Eingrenzung der Suchanfragen ist somit essentiell.

RAM-Speicher

Zur Untersuchung der erzeugten Dumps des RAMs eignet sich, wie Björn Roos in seiner Arbeit anführt, der Befehl `strings` (vgl. [Roo11] S. 108). Dieser erlaubt die Extraktion sämtlicher auf ASCII basierenden Zeichenketten und deren Überführung in eine neue Textdatei. Um eine erfolgreiche Extraktion zu gewährleisten, sollten verschiedene zusätzliche Parameter in Betracht gezogen werden (vgl. [Hoo11] S. 294). Mit Hilfe der Angabe von `--all` wird demzufolge die Untersuchung der vollständigen Dumps sichergestellt. Der Parameter `--radix` ergänzt die extrahierten Zeichenketten um ihre Position im Dump. Unter Verwendung eines Hex-Editors ist dadurch eine leichte Ermittlung der Zeichenketten im Dump möglich. Zuletzt ist mit `--encoding`, sofern bekannt, die Kodierung der zu suchenden Zeichenketten anzugeben. Die erstellte Textdatei kann anschließend auf wertvolle Daten, wie z. B. Anmeldedaten, und Spuren untersucht werden, die im Zuge einer forensischen Untersuchung von Bedeutung sind.

Liegt ein mit LiME erzeugter Dump vor, dann sollte die physische Untersuchung mit dem „Volatility Framework“ durchgeführt werden. Das Framework wurde von Experten aus den Bereichen IT-Forensik, Incident Response sowie Malware in der Programmiersprache Python entwickelt und umfasst eine Zusammenstellung kommandozeilenbasierter Werkzeuge [ACC+13]. Darüber ist eine Interpretation und formatierte Ausgabe der im Dump vorliegenden Daten und auf Basis derer eine einfache Extraktion der potentiellen Spuren möglich. Demgemäß können beispielhaft die zum Zeitpunkt der Sicherung laufenden Prozesse auf dem Smartphone eingesehen werden (siehe Abbildung 17):

```
Offset      Name         PID  UID  Start Time
0xd3c13c00  init         1    0    Sun, 05 Aug 2012 04:20:04
0xd3c13800  kthreadd      2    0    Sun, 05 Aug 2012 04:20:04
0xd3c13400  ksoftirqd/0   3    0    Sun, 05 Aug 2012 04:20:04
```

```
0xd3c13000    watchdog/0   4    0    Sun, 05 Aug 2012 04:20:04
0xd3c1bc00    events/0     5    0    Sun, 05 Aug 2012 04:20:04
0xd3c1b800    khelper      6    0    Sun, 05 Aug 2012 04:20:04
0xd3c1b400    async/mgr    7    0    Sun, 05 Aug 2012 04:20:04
```

Abbildung 17: Physische Untersuchung des RAMs mit Hilfe des Volatility Frameworks

Flash-Speicher

Zur physischen Untersuchung der physikalischen Sicherungen des internen und externen Flash-Speichers eignen sich verschiedene Methoden. Ein erster Ansatz besteht darin, sämtliche Zeichenketten aus den Rohdaten mit dem bereits aufgeführten Befehl `strings` zu extrahieren und anschließend zu sichten. In Kombination mit `grep` kann die Extraktion auf dedizierte Zeichenketten beschränkt und infolgedessen die zurückgegebene Datenmenge reduziert werden. Auf diese Weise ist es beispielsweise möglich, Passwörter von Benutzern aus einer Sicherung zu erhalten. Die Syntax einer solchen Anfrage sieht beispielsweise wie folgt aus (siehe Abbildung 18):

```
$ strings --all --radix=x Sicherung.dd | grep "Zeichenkette"> 
/home/dmuth/Sicherung.txt
```

Abbildung 18: Physische Untersuchung der Flash-Speicher mit strings und grep

Aufgrund der Verwendung von traditionellen Dateisystemen, wie Ext4 und FAT32, auf den Smartphones können ferner klassische Untersuchungswerkzeuge, wie das von Brian Carrier entwickelte „The Sleuth Kit“, eingesetzt werden. Hierbei handelt es sich um eine Zusammenstellung von kommandozeilenbasierten Werkzeugen zur Extraktion potentieller Spuren in einer forensischen Untersuchung [Car13]. Der Umgang mit den Werkzeugen und die korrekte Interpretation der ausgegebenen Daten erfordert ein grundlegendes Verständnis der Struktur des zu untersuchenden Dateisystems. Für eine solche Betrachtung von Ext4 und FAT32 sei an dieser Stelle beispielhaft auf [Fai12] und [Mic06] verwiesen.

Zur Wiederherstellung von gelöschten oder absichtlich versteckten Dateien, die sich außerhalb des Dateisystems befinden, bieten sich Werkzeuge zum „File Carving“ an. Beim File Carving werden die Rohdaten der Sicherung nach Signaturen durchsucht, die den Header und, falls vorhanden, den Footer eines bekannten Dateiformats angeben. Die Suche schließt hierbei auch Speicherbereiche ein, die vom Dateisystem nicht allokiert sind, jedoch in der Sicherung vorliegen. Auf dem Gebiet der IT-Forensik zum File Carving sind die Werkzeuge „Foremost“ und „Scalpel“ weit verbreitet. Die Grundlage der Werkzeuge bildet jeweils eine Konfigurationsdatei, die eine Auflistung der zu suchenden Dateitypen sowie die zugehörigen Signaturen des Headers und Footers enthält. Während der Ausführung der Werkzeuge werden die extrahierten Dateien entsprechend ihres Typs in einem Ordner abgelegt. Zusätzlich wird eine `audit.txt` angelegt, die weitere Informationen, wie die Größe jeder extrahierten Datei, enthält. Anschließend sind möglicherweise beschädigte Dateien zu entfernen und die gewonnen Daten nach potentiellen Spuren zu sichten.

SIM-Karte

Die Rohdaten der SIM-Karte, welche mit Hilfe eines forensischen Kartenlesers gesichert wurden, sind bevorzugt mit der zugehörigen Software zu untersuchen. Die Software übernimmt für gewöhnlich die Interpretation der Rohdaten und erlaubt durch die Bereitstellung einer grafischen Oberfläche einen einfachen Zugriff auf die potentiellen Spuren. Alternativ kann die Sicherung, deren Dateiformat in der Regel vom Hersteller der Software abhängt, mit den bereits angeführten Befehlen `strings` und `grep` nach für den Vorfall relevanten Spuren durchsucht werden.

4.5 Datenanalyse

Nachdem die potentiellen Spuren aus den angefertigten Sicherungen extrahiert wurden, erfolgt im Zuge der Datenanalyse eine **Korrelation** und **Bewertung** dieser Daten. Das Ziel besteht darin, aus den ermittelten digitalen Spuren, wie Kurzmitteilungen, E-Mails, Bildern und beliebigen Dokumenten, endgültige Befunde und Schlussfolgerungen zu ziehen, welche einen Tatbestand sowohl belegen als auch widerlegen können. Besonders die Untersuchungsschritte und Methoden zur Bewertung orientieren sich an dem Vorgehen der klassischen IT-Forensik.

4.5.1 Korrelation

Im Zuge der Korrelation sollen die extrahierten Spuren in einen inhaltlichen und zeitlichen Zusammenhang gebracht werden. Darüber sollen schließlich weiterführende Befunde über einen Vorfall aus den vorliegenden Spuren gewonnen werden. Zwei wesentliche Aufgaben, die vor allem in der Smartphone-Forensik zum Tragen kommen, liegen in der Erstellung von Bewegungsprofilen und Zeitlinien.

Bewegungsprofile

Die Bewegungsprofile können aus den extrahierten Standortdaten abgeleitet werden. Dadurch ist es möglich, den Aufenthaltsort des Smartphones und somit wahrscheinlich auch des Besitzers zu eindeutigen Zeitpunkten zu bestimmen. In früheren Versionen von Android eigneten sich hierzu die Dateien `cache.cell` und `cache.wifi` im Verzeichnis `/data/data/com.google.android.location`. Darin befand sich jeweils eine Historie der mit dem Smartphone über WLAN verbundenen Funkzellen und Zugangspunkte. Vermutlich aufgrund der damit einhergehenden datenschutzrechtlichen Diskussionen im April 2011 wurde die Erstellung der Dateien unter Jelly Bean unterbunden [Hei11].

Alternativ besteht die Möglichkeit, die Bewegungsprofile mit den aus der SIM-Karte extrahierten Standortdaten zu erstellen. Die Standortdaten ergeben sich hierbei ebenfalls aus den zuletzt mit dem Smartphone verbundenen Funkzellen, deren Position durch eine eindeutige Identifikationsnummer geografisch bestimmt werden kann. Beispielsweise auf der Internetseite von „Google Maps“ können anschließend die Standorte chronologisch abgetragen, visualisiert und zu einem Bewegungsprofil zusammengeführt werden.

Weiterhin lassen sich Bewegungsprofile mit Hilfe der Daten aus den angeführten Datenbanken `search_history.db` und `da_destination_history.db` ermitteln. Diese werden von der Applikation „Google Maps“ erzeugt, welche auf Android-Smartphones standardmäßig installiert ist. In den Datenbanken werden unter anderem die vom Benutzer aufgerufenen Routen und Suchanfragen bezüglich ausgewählter Standorte gespeichert. Bei diesem Ansatz ist allerdings zu berücksichtigen, dass die Spuren keinen Aufschluss darüber geben, ob das Smartphone tatsächlich am Zielort angekommen ist. Lediglich können hiermit der aktuelle Aufenthaltsort des Gerätes und folglich des Besitzers zum Zeitpunkt des Aufrufs der Route oder des Standortes und ein potentieller Zielort identifiziert werden.

Zuletzt bieten sich zur Entwicklung eines Bewegungsprofils Bilder an, die über Standortdaten verfügen. Die Daten werden mit weiteren Informationen, wie dem Erstellungsdatum, bei der Bildaufnahme im „Exchangeable Image File Format“ im Header des Bildes gespeichert und können mit dem Befehl `exif` in Linux ausgelesen werden (vgl. [Cam10] S. 68-77). Anschließend ist es beispielsweise über die Internetseite von Google Maps möglich, mit Hilfe der daraus gewonnen Koordinaten den genauen Gerätestandort zum Zeitpunkt der Bildaufnahme zu bestimmen. Anhand des Datums der Bilder kann ebenfalls eine chronologische Reihenfolge der Standorte festgelegt werden.

Ebenso können zusätzlich installierte Applikationen zur Positionsbestimmung oder Navigation eine wertvolle Bezugsquelle für Standortdaten darstellen. Die Liste der Möglichkeiten kann somit beliebig ergänzt werden.

Zeitlinien

Eine Zeitlinie im Bereich der IT-Forensik gibt Aufschluss darüber, welche Aktivitäten in den Dateisystemen des Smartphones zu welchen Zeitpunkten stattgefunden haben. Die Grundlage zur Anfertigung einer Zeitlinie bilden die verschiedenen Zeitstempel einer Datei innerhalb eines Dateisystems. Hierbei wird zwischen den Zeiten „modified“ (m), „accessed“ (a), „changed“ (c) und „created“ (b) unterschieden, welche in der Literatur auch unter dem Begriff „MAC Times“ zu finden sind (vgl. [Hoo11] S. 286). Diese werden auf Basis der vorliegenden Systemzeit gebildet. Eine korrekte Zeitbasis auf dem Smartphone erspart demnach die Berücksichtigung von Abweichungen und kann folglich die Untersuchung maßgeblich erleichtern. Zur Extraktion der Zeitinformationen aus dem physikalischen Abbild kann der im The Sleuth Kit enthaltene Befehl `fls` verwendet werden. Der Befehl listet die Namen der Dateien und Verzeichnisse innerhalb eines physikalischen Abbilds auf. Durch Angabe des Parameters `-m` wird die Ausgabe unter anderem um die benötigten Zeitstempel ergänzt. Die Ergebnisse der Befehlsausführung sind in eine Datei zu schreiben, die anschließend zur Erstellung der Zeitlinie genutzt wird.

Zur Erstellung bietet sich die Ausführung des im Kit enthaltenen Befehls `mactime` an. Dieser sortiert die Daten nach den vorhandenen Zeitenstempeln und erstellt eine formatierte Ausgabe.

Die Berücksichtigung aller Zeitstempel erlaubt beispielsweise eine Differenzierung zwischen der Erstellung und der Veränderung einer Datei oder eines Verzeichnisses. Die folgende Abbildung enthält einen Ausschnitt einer solchen Ausgabe:

Data	Zeit	Größe	MAC Times		Name
Thu May 16 2013	14:37:23	4096	.a.b	…	/data
	14:37:23	4096	.a.b	…	/media
	14:37:23	4096	.a.b	…	/misc
	14:37:23	4096	.a.b	…	/system
Thu Jun 13 2013	16:45:50	4096	m...	…	/data
Mon Jun 17 2013	12:05:11	4096	..c.	…	/misc
Thu Jun 27 2013	21:08:32	4096	..c.	…	/media
Fri Jun 28 2013	03:05:21	4096	m.c.	…	/system

Abbildung 19: Erstellung einer tabellarischen Zeitlinie

Daneben kann über das Werkzeug „Simile Timeline“ die Anfertigung einer visualisierten Zeitlinie, wie beispielhaft in Abbildung 20 dargestellt, erfolgen.

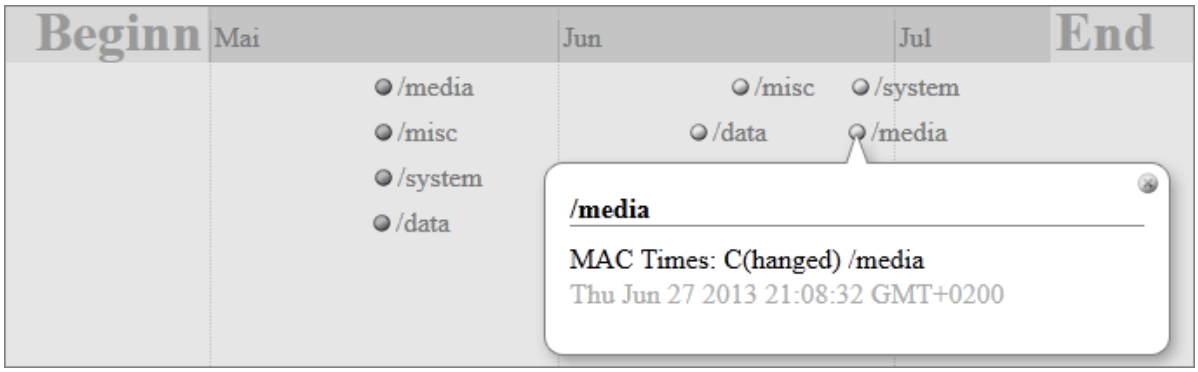

Abbildung 20: Erstellung einer visualisierten Zeitlinie

Simile Timeline wurde innerhalb eines Forschungsprojekts am Massachusetts Institute of Technology entwickelt [Mas13].

Die Basis bildet eine Datei im „Extensible Markup Language“-Format, die zuvor anhand der Ausgabedatei von `fls` zu erstellen ist. Für diesen Zweck bietet sich das Werkzeug „Log2Timeline“ an, welches im Jahr 2010 im SANS Reading Room veröffentlicht wurde [Gud10]. Dieses generiert ebenfalls eine tabellarische Zeitlinie unter Berücksichtigung aller vorhandenen Zeitstempel. Im Unterschied zur vorherigen Variante kann hierbei jedoch ein spezifisches Ausgabeformat, wie z. B. Extensible Markup Language, festgelegt werden. Simile erlaubt aufbauend auf der Datei anschließend die Visualisierung der gegebenen Daten.

4.5.2 Bewertung

Eine erste objektive Bewertung der extrahierten Spuren findet bereits innerhalb der Datenuntersuchung statt. Dadurch werden erste für den Vorfall relevante und irrelevante Spuren voneinander getrennt. Nach dieser Vorverarbeitung ist anschließend eine fortsetzende Bewertung im Zusammenhang mit dem Tatbestand durchzuführen (vgl. [Ges11] S. 83-84). Eine objektive Herangehensweise ist, wie im gesamten Untersuchungsverlauf, maßgebliche Grundvoraussetzung. Daran knüpft auch das Gebot an, sowohl digitale Spuren zu ermitteln, die einen Tatbestand belegen, als auch solche, die ihn widerlegen können. Ferner steht es dem Forensiker nicht zu, eigenmächtig das Bestehen oder Nichtbestehen einer Tat zu beurteilen. Lediglich können die Befunde und Schlussfolgerungen dargelegt werden, die sich aus den gewonnenen Spuren nachweislich ableiten lassen (vgl. [Bun11] S. 64). Hierbei ist es essentiell, die abgeleiteten Schlussfolgerungen kritisch zu überprüfen, um Fehler oder gar Lücken in der Argumentationskette frühzeitig zu identifizieren und nach Möglichkeit zu beseitigen [Bun09].

4.6 Dokumentation

Die Dokumentation muss prozessbegleitend während des gesamten Untersuchungsverlaufs und abschließend zur Zusammenfassung der Ergebnisse für eine ausgewählte Zielgruppe durchgeführt werden. Repräsentiert wird dieser Sachverhalt durch die zwei Prozessbausteine **Verlaufsprotokoll** und **Ergebnisprotokoll**. Im Vorfeld empfiehlt es sich, für diesen Zweck Vorlagen anzufertigen, auf die stets in forensischen Untersuchungen zurückgegriffen werden kann.

Der Prozess der Protokollierung, welcher zur Vermeidung von Fehlinterpretationen ein hohes Maß an Genauigkeit voraussetzt, kann klassisch mit Papier und Stift oder in elektronischer Form erfolgen. Während die klassische Variante in der Regel zu einer präziseren Arbeitsweise führt, ermöglicht der elektronische Ansatz eine einfache Weiterverarbeitung. Wird eine elektronische Protokollierung präferiert, sollten in regelmäßigen Abständen kryptografische Hashwerte bezüglich der Dateien gebildet werden (vgl. [Ges11] S. 83). Dadurch kann fortlaufend überprüft werden, ob seit der letzten Bearbeitung der Dateien Manipulationen vorgenommen wurden. Darüber hinaus ist es ratsam, unabhängig von der gewählten Dokumentationsform, zur Gewährleistung der Authentizität stets nach dem Vier-Augen-Prinzip zu arbeiten (vgl. [Ges11] S. 84).

4.6.1 Verlaufsprotokoll

Eine essentielle Aufgabe zur Anfertigung eines Ergebnisprotokolls ist die prozessbegleitende Protokollierung des gesamten Untersuchungsverlaufs. Diese sollte, neben dem Anfangsverdacht, sämtliche durchgeführten Untersuchungsschritte und Methoden, verwendeten Werkzeuge, gesicherten Daten sowie gewonnen Spuren und Ergebnisse enthalten. Besonders wichtig ist in diesem Zusammenhang, neben dem Aufbau und der Aufrechterhaltung einer Chain of Custody, auch die Erfassung von möglichen Unvollständigkeiten und Verfälschungen, die während der Untersuchung entstehen können. Eine tabellarische Aufzeichnung dieser Punkte, wie beispielhaft in Tabelle 3 gezeigt, trägt hierbei erheblich zur Übersichtlichkeit und Nachvollziehbarkeit bei (vgl. [Ges11] S. 84).

Datum	**Uhrzeit**	**Aktion**	**Kommentar**	**Person**
23.06.13	12:00:00	`dd if=/dev/mmcblk0p1 of=/home/dmuth /FlashExtern.dd`	Vollständige Sicherung des externen Flash-Speichers des sichergestellten Smartphones mittels `dd`. Speicherung der Sicherung im Verzeichnis `/home/dmuth` unter dem Namen `FlashExtern.dd`.	D. Muth

23.06.13	12:45:05	`sha1sum FlashExtern. dd`	Erstellung eines kryptografischen Hashwerts für die angefertigte Sicherung des externen Flash-Speichers mit `sha1sum`. Ergebnis der Hashwert-Bildung: c1fa3bc 14299e5016f8d50dac0a9d0b d40cad933	D. Muth
…	…	…	…	…

Tabelle 3: Beispielhafte Formatvorlage zur Protokollierung der Untersuchungsschritte

Der Vorgang kann durch verschiedene Werkzeuge unterstützt werden. Beispielsweise eignet sich zur Aufzeichnung der Ein- und Ausgaben in einer Shell der kommandozeilenbasierte Befehl `script` unter Linux. Auch ist die Anfertigung von Bildern und Videos denkbar. Ferner bieten in der Regel forensische Untersuchungswerkzeuge integrierte Funktionen zur Berichtserstellung an. Insgesamt wird auf diese Weise sichergestellt, dass im abschließenden Ergebnisprotokoll keine entscheidenden Aspekte vergessen werden. Darüber hinaus erleichtert eine solche Vorgehensweise den Aufbau einer validen Argumentationskette.

4.6.2 Ergebnisprotokoll

Die Erstellung des Ergebnisprotokolls ist nach Abschluss einer forensischen Untersuchung durchzuführen. Darin sollen sämtliche für eine dedizierte Zielgruppe essentiellen Informationen zusammengefasst werden. Aufgrund der Abhängigkeit zur Zielgruppe kann die Form des Protokolls sehr unterschiedlich ausfallen. Eine ausführliche Auflistung der potentiellen Inhalte kann im NIST eingesehen werden (vgl. [Nat07] S. 66-67). Im Wesentlichen sollten darin die Untersuchungsumgebung (vor allem die verwendeten Werkzeuge samt Versionsnummer), die Vorgehensweise bei der forensischen Untersuchung sowie die gewonnen Spuren und daraus abgeleiteten Schlussfolgerungen beschrieben sein. Ersteres ist besonders wichtig, um dem Leser die Möglichkeit zu geben, die Vorgehensweise nachzuvollziehen und gleichzeitig die Beweiskraft der Ergebnisse bewerten zu können (vgl. [Bun11] S. 62).

Eine präzise und verständliche Formulierung der Inhalte ist hierbei – vor allem, wenn es um die gerichtliche Verwertbarkeit geht – maßgebliche Grundvoraussetzung und steigert darüber hinaus die Glaubwürdigkeit.

Im Kapitel „Android-Prozess“ erfolgte die Untersuchung der Prozessabschnitte des erarbeiteten Prozessmodells im Zusammenhang mit Android. Im Zuge dessen wurden Methoden und Werkzeuge zur forensischen Untersuchung erarbeitet.

Die Methoden der strategischen und operationalen Vorbereitung sind weitestgehend unabhängig vom Betriebssystem. Spezifisch für Android ist vor allem der integrierte Protokollierungsdienst Logcat, welcher im Anschluss an die strategische Vorbereitung ein wichtiges Werkzeug zur Überwachung der Interaktionen mit dem Gerät darstellt. Darüber hinaus bietet der Offline-Modus des Betriebssystems in der operationalen Vorbereitung eine effektive Methode zur Abschirmung der Funkverbindungen.

Für eine Datensammlung unter Android ist ein aktiviertes USB-Debugging Grundvoraussetzung. Die etablierten Schutzmaßnahmen, wie z. B. ein vorhandener Sperrbildschirm, erschweren nicht nur die Aktivierung dieser essentiellen Option, sondern beeinflussen zusammen mit dem Betriebszustand des Geräts insgesamt die Methoden zur Sicherung der Daten. Die Datenuntersuchung der Sicherungen der Flash-Speicher kann aufgrund von Dateisystemen wie Ext4 und FAT32 mit Methoden aus der klassischen IT-Forensik erfolgen. Von besonderer Relevanz ist hierbei die Data-Partition des internen Flash-Speichers. Der Bereich enthält die von den Applikationen gespeicherten benutzerspezifischen Daten und somit die meisten potentiellen Spuren. In der Datenanalyse können aus den Spuren Bewegungsprofile und Zeitlinien erstellt sowie Schlussfolgerungen hinsichtlich des dedizierten Vorfalls abgeleitet werden.

Ein wichtiger Bestandteil der Dokumentation ist die Erfassung von Unvollständigkeiten und Verfälschungen, die während der Untersuchung von Android-Smartphones nicht ausgeschlossen werden können.

5 Datensammlung

Der interne Flash-Speicher ist innerhalb der Datensammlung aufgrund seiner persistenten Eigenschaft, hohen Speicherkapazitäten und Rolle als primäres Speichermedium im Smartphone eine zentrale Komponente. Wie bereits in Kapitel 4.3.3 angeführt, erfolgt die Sicherung des internen Speichermediums zum gegenwärtigen Zeitpunkt bevorzugt über softwarebasierte Methoden. Grund hierfür ist vor allem die Gefahr einer Zerstörung des Speichers und somit eines vollständigen Datenverlusts, welche mit den hardwarebasierten Ansätzen einhergeht. Prinzipiell beruhen die softwarebasierten Methoden auf drei wesentlichen Ausgangspunkten, die innerhalb dieser Ausarbeitung erarbeitet werden konnten: der Recovery Mode, die Content Provider und die Android Debug Bridge. Die Wahl des Ausgangspunkts und folglich der Methoden wird hierbei maßgeblich vom Systemzustand beeinflusst, in dem das Smartphone während der Beschlagnahmung vorgefunden wird. Demnach geben, wie in der nachfolgenden Abbildung skizziert, der aktuelle Betriebszustand und darauf folgend die gegebenen Schutzmaßnahmen die Durchführung der softwarebasierten Datensammlung vor (siehe Abbildung 21):

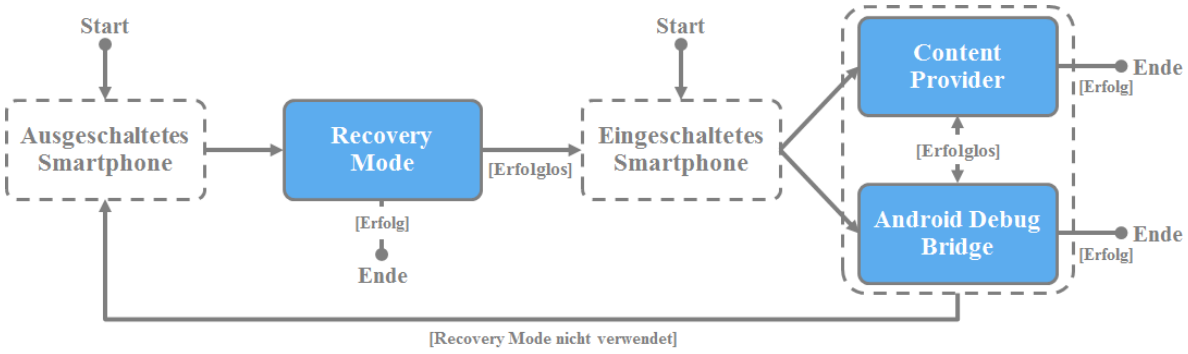

Abbildung 21: Durchführung der Datensammlung des internen Flash-Speichers

Nachfolgend wird genauer auf diesen im Zuge der Masterarbeit entwickelten Ablauf und die jeweiligen Methoden innerhalb der Ausgangspunkte eingegangen.

5.1 Ausgeschaltetes Smartphone

Liegt das auf Android basierende Smartphone im ausgeschalteten Zustand vor, sollte über den Bootloader des Geräts der Recovery Mode gestartet werden (vgl. [Hoo11] S. 203). Der Bootloader wird durch das Drücken einer gerätespezifischen Tastenkombination aufgerufen.

Das Google Nexus 4 erfordert beispielsweise das Drücken der Tasten „Lautstärke hoch“ und „Einschalten“. Über den Recovery Mode wird angestrebt, eine Sicherung des internen und bei Notwendigkeit auch des externen Flash-Speichers durchzuführen. Dadurch sollen die forensisch relevanten Daten vor Veränderungen geschützt werden, die bei einem herkömmlichen Bootvorgang nicht auszuschließen sind. Ferner entfällt damit die Aufhebung eines vorhandenen Sperrbildschirms. Lediglich eine Entschlüsselung des internen Flash-Speichers, d.h. der Data-Partition, und des externen Flash-Speichers ist in diesem Modus nicht möglich. Zwar könnte ein Abbild der Rohdaten erstellt werden, jedoch wären die gewonnen Daten nicht lesbar und somit in einer forensischen Untersuchung unbrauchbar.

Standardmäßig verfügen die vom Hersteller ausgelieferten Smartphones nur über einen äußerst rudimentären Recovery Mode, der lediglich eine Wartung des Geräts erlaubt. Eine Erweiterung des Funktionsumfangs, vornehmlich um eine ADB mit privilegierten Rechten zur Datensammlung, kann jedoch durch die Modifikation der Recovery-Partition erreicht werden (vgl. [Hoo11] S. 272 und [VZC11] S. 18). Daran angelehnt ist im ausgeschalteten Zustand zwischen zwei Fällen zu unterscheiden. Entweder die Partition wurde bereits durch den Gerätebesitzer modifiziert oder eine Veränderung ist innerhalb der forensischen Untersuchung notwendig.

5.1.1 Recovery Mode (1)

Im privaten Umfeld kann es vorkommen, dass der Besitzer des Smartphones eine „Custom ROM“, d.h. eine angepasste Version von Android, installiert hat. Im Zuge der Installation wird für gewöhnlich auch die Recovery-Partition und demgemäß der Recovery Mode ersetzt (vgl. [Hoo11] S. 272). Zur Überprüfung dieses Sachverhalts ist auf der Workstation, die mit dem im Recovery Mode befindlichen Smartphone verbunden ist, eine Shell zu öffnen und der Befehl `adb devices` abzusetzen. Wird das angeschlossene Gerät in der Shell aufgelistet, kann darauf geschlossen werden, dass ein Austausch der Partition bereits stattgefunden hat (vgl. [Hoo11] 273). Infolgedessen ist ein Zugriff auf die Daten des Smartphones über die ADB möglich. Mit Hilfe der privilegierten Rechte können im Recovery Mode die zu sichernden Partitionen mit `umount` ausgehängt oder über den Befehl `mount` ausschließlich lesend eingehängt werden. Ziel ist es, auf diese Weise eine Veränderung der Daten und folglich eine Integritätsverletzung im weiteren Untersuchungsverlauf zu verhindern.

Unter Verwendung des Befehls `dd` kann danach die Sicherung der relevanten Partitionen des internen und externen Flash-Speichers durchgeführt werden. Zum Nachweis, dass währenddessen keine Veränderungen vorgenommen wurden, sind die vor und nach der Sicherung berechneten Hashwerte des Originals und des physikalischen Abbilds zu vergleichen.

Aus forensischer Sicht erzielt diese Methode zur Datensammlung die besten Ergebnisse, da mit Hilfe der im Betriebssystem vorhandenen Werkzeuge ohne die Manipulation des Smartphones eine Sicherung erfolgen kann. Das Problem einer geminderten Beweiskraft der digitalen Spuren besteht demzufolge nicht. Auch erweist sich die Umsetzung des Ansatzes als vergleichsweise einfach. Es wird kein spezielles Wissen zur Modifikation der Recovery-Partition benötigt. Darüber hinaus entfällt mit dieser Variante das Risiko einer Beschädigung des Smartphones, die aus der Installation einer modifizierten Version resultieren kann.

5.1.2 Recovery Mode (2)

Besonders im geschäftlichen Umfeld ist die Modifikation der Recovery-Partition aufgrund bestehender Unternehmensrichtlinien in der Regel ausgeschlossen. Zur Durchführung einer Sicherung im Recovery Mode ist es daher erforderlich, eigenhändig eine modifizierte Version zu installieren. Im Bootloader kann, sofern dieser ab Werk nicht gesperrt ist, über herstellerspezifische Protokolle, wie „Fastboot“ oder „Odin“, mit dem über die USB-Schnittstelle angeschlossenen Smartphone kommuniziert werden (vgl. [Hoo11] S. 208-209 und [VZC11] S. 19-21). Die Steuerung der Kommunikation erfolgt über eine zugehörige Software, die auf der forensischen Workstation einzurichten ist und das Überschreiben bestehender Partitionen ermöglicht. Üblicherweise beheben Hersteller auf diese Weise softwareseitige Fehler in nicht mehr betriebsfähigen Smartphones. Alternativ kann damit die Recovery-Partition überschrieben und somit ein modifizierter Recovery Mode installiert werden. Dies ist lediglich dann ausgeschlossen, wenn eine Sperrung des Bootloaders vorliegt. Ein Indiz hierfür können die im Recovery Mode angezeigte Signatur, welche beispielsweise den Inhalt „Lock State: Locked“ enthält, oder ein fehlerhafter Verbindungsaufbau sein (vgl. [Hoo11] S. 208). Zwar ist die Aufhebung der Sperrung denkbar, jedoch geht damit auch die Wiederherstellung der Werkseinstellungen und folglich die Entfernung der forensisch essentiellen Daten einher. Aus diesem Grund sollte von einer solchen Maßnahme abgesehen und stattdessen das Smartphone herkömmlich gestartet werden.

Ist der Austausch aufgrund eines offenen Bootloaders möglich, bietet sich zur Suche nach einem alternativen Recovery Mode für ein spezifisches Smartphone ebenfalls das Forum XDA Developers an. Nach erfolgreicher Überschreibung der Partition kann die Sicherung analog zur vorherigen Variante erfolgen.

Die mit der Überschreibung der Partition in Verbindung stehenden Manipulationen können ggf. den Kernel des Smartphones betreffen, sollten jedoch keine Auswirkungen auf die forensisch relevanten Bereiche des internen Flash-Speichers haben. Insofern ist auch hier von einer unbeeinträchtigten Beweiskraft auszugehen. Allerdings muss berücksichtigt werden, dass eine falsche Umsetzung, beispielsweise die Installation eines fehlerhaften oder inkompatiblen Recovery Modes, zu einer irreparablen Beschädigung des Smartphones führen kann. Um solche Komplikationen auszuschließen, ist der Vorgang zuvor in einer Testumgebung für den konkreten Anwendungsfall zu untersuchen. Ein grundlegendes Verständnis des Aufbaus der Recovery-Partition ist hierfür zwingend erforderlich. Zur Schaffung eines ersten Überblicks empfiehlt sich in diesem Kontext das Paper „Toward a general collection methodology for Android devices“ (siehe [VZC11] S. 17-18).

5.2 Eingeschaltetes Smartphone

Ein eingeschaltetes Smartphone sollte zur Durchführung der Datensammlung in diesem Betriebszustand belassen werden (vgl. [Hoo11] S. 208). Im konkreten Anwendungsfall ist nicht überschaubar, welche Folgen die Abschaltung des in der Bestandsaufnahme beschlagnahmten Geräts mit sich bringt. Auch ist zu bedenken, dass bereits beim Systemstart essentielle Sicherheitscodes, wie die PIN der SIM-Karte und das Passwort zur Entschlüsselung des internen und externen Flash-Speichers, eingegeben werden. Demnach sind die Daten bei einem eingeschalteten USB-Debugging über das Smartphone zugänglich. Liegen die entsprechenden Sicherheitscodes dem Forensiker nicht vor, wäre der Zugriff auf die Daten nach der Abschaltung des Geräts vorerst ausgeschlossen.

Die Sicherung des internen und bei Bedarf des externen Flash-Speichers kann im laufenden Betrieb entweder über die im Betriebssystem integrierten Content Provider oder die Android Debug Bridge erfolgen. Die Wahl des Ausgangspunkts hängt maßgeblich von den benötigten Daten ab. Bei vorhandenen privilegierten Rechten kann eine Datensammlung über die Android Debug Bridge wesentlich ergiebiger sein.

Der Recovery Mode und dementsprechend die Abschaltung des Smartphones sollten erst in Erwägung gezogen werden, wenn die mit den nachfolgenden Methoden gewonnen Daten nicht ausreichend sind.

5.2.1 Content Provider

Die Content Provider sind fester Bestandteil des Applikationsrahmens in der Architektur von Android. Wie bereits in den Grundlagen angeführt, werden sie über Uniform Ressource Identifier eindeutig adressiert und erlauben den Datenaustausch zwischen Applikationen (siehe Kapitel 2.2.1 „Architektur“ und Kapitel 2.2.2 „Sicherheit“). Die Nutzung der Schnittstelle zur Datensammlung erfordert keine privilegierten Rechte auf dem Smartphone. Zum Datenaustausch wird lediglich ein aktiviertes USB-Debugging benötigt. Zusätzliche Manipulationen, die zu einer Integritätsverletzung führen können, sind also nicht erforderlich. Zwar sind im Zuge der Datensammlung nur jene Daten zugänglich, die explizit von den auf dem Smartphone befindlichen Applikationen bereitgestellt werden. Allerdings sind dies hauptsächlich Daten der Data-Partition, welche über die Android Debug Bridge ausschließlich mit privilegierten Rechten bezogen werden können. Unter anderem existieren Provider zum Austausch von Kontaktdaten, Kurzmitteilungen, E-Mails und Bildern.

Lösungen

Auf Basis von Content Providern sind auf den Markt zahlreiche kommerzielle Werkzeuge verfügbar (vgl. [Hoo11] S. 228-229). Im Fokus stehen vor allem Produkte, die federführend von Strafverfolgungsbehörden eingesetzt werden. Zu den Marktführern zählen vornehmlich „.XRY“ von Micro Systemation, „UFED“ von Cellebrite und das bereits in der klassischen IT-Forensik etablierte „EnCase“ von Guidance Software. Darüber hinaus ist die kostenfreie Lösung „AFLogical“ von viaForensics anzuführen. Das Produkt wurde von Experten aus dem Bereich der mobilen Sicherheit sowie Forensik entwickelt und im Vergleich zu kommerziellen Lösungen hinsichtlich der Datenmenge, die während des Sicherungsvorgangs bezogen werden kann, als sehr gut befunden [KO10].

Der wesentliche technische Unterschied in den aufgeführten Produkten besteht, neben den unterstützten Content Providern, in der Installation. Die Methode zur Durchführung der Datensammlung auf Basis der Content Provider erfolgt jedoch in allen Lösungen auf die gleiche Art und Weise (vgl. [Hoo11] S. 220).

Methode

Während das UFED ein eigenständiges Gerät darstellt, das direkt mit dem Smartphone verbunden werden kann, erfordern .XRY und EnCase zuvor die Installation einer Applikation auf der Workstation (vgl. [Hoo11] S. 230, S. 243, S. 249). Nach erfolgreicher Einrichtung kann eine Verbindung über die USB-Schnittstelle zum Smartphone aufgebaut werden. Bei AFLogical ist die Applikation hingegen auf dem internen Flash-Speicher des zu untersuchenden Geräts zu installieren (vgl. [Hoo11] S. 223). Dies erfolgt unter Einsatz der ADB beispielhaft mit dem Befehl `adb install AFLogical.apk`. Demgemäß benötigt die Lösung bereits in diesem Schritt ein aktiviertes USB-Debugging. Auch werden im Zuge der Installation Veränderungen an den Inhalten des internen Flash-Speichers vorgenommen, die aus forensischer Sicht nach Möglichkeit zu vermeiden sind. Im Vergleich zu den kommerziellen Produkten weist AFLogical somit bezüglich der Einrichtung ein Defizit auf.

Zur Datensammlung ist bei allen Lösungen ein aktiviertes USB-Debugging Grundvoraussetzung. Grundsätzlich implementieren die Applikationen der Produkte zur Anforderung der Daten sogenannte „Content Resolver“ [And13b]. Für den Zugriff auf die Daten muss die Applikationen über die notwendigen Leserechte des jeweiligen Content Providers verfügen. Diese sind aus den zugehörigen Dokumentationen zu entnehmen und bei der Entwicklung im Manifest der Applikation unter dem Tag `uses-permissions` einzutragen [And13b]. Der Content Provider wiederum empfängt die Anfragen, verarbeitet diese bei gegebenen Rechten und sendet die Ergebnisse zurück an den Content Resolver. Während die kommerziellen Produkte die empfangenen Daten auf der forensischen Workstation ablegen, speichert AFLogical diese auf dem internen oder – falls vorhandenen – dem externen Flash-Speicher. Zum Schutz der forensisch relevanten Speichermedien sollte, sofern möglich, ein gesonderter externer Flash-Speicher für diesen Zweck verwendet werden. Anschließend können die gewonnen Daten einer tiefgehenden Datenuntersuchung unterzogen werden.

5.2.2 Android Debug Bridge

Die Android Debug Bridge ermöglicht, vorausgesetzt das USB-Debugging ist eingeschaltet, einen umfassenden Zugriff auf das zu Grunde liegende Smartphone. Dadurch kann mittels der in Android integrierten Befehle `adb pull` und `adb shell` eine Datensammlung des internen und bei Bedarf des externen Flash-Speichers durchgeführt werden.

Die Installation zusätzlicher Werkzeuge zur Anfertigung der Sicherungen ist in der Regel nicht erforderlich. Allerdings kann je nach benötigten Daten die Notwendigkeit bestehen, das Smartphone einem Root-Vorgang zu unterziehen, um privilegierte Rechte einzuräumen und den Zugriff beispielsweise auf die Data-Partition zu erweitern.

adb pull

Mit Hilfe des Befehls `adb pull` werden Daten über die Android Debug Bridge vom Smartphone auf die forensische Workstation übertragen. Auf diese Weise lassen sich logische Objekte, wie Dateien und ganze Verzeichnisse, mit einem geringen Aufwand sichern. Zur Gewährleistung der Integrität sind zuvor über die Originaldaten und später über die Daten der Sicherungen kryptografische Hashwerte zu bilden. Beispielhaft kann die Anwendung der Methode wie folgt aussehen (siehe Abbildung 22):

```
$ adb pull /data /home/dmuth
```

Abbildung 22: Sicherung des internen Flash-Speichers durch Verwendung von adb pull

Während der erste Pfad das zu übertragende Objekt auf dem Smartphone angibt, steht der zweite Pfad für den Speicherort auf der forensischen Workstation. Ein grundlegendes Verständnis der Partitionen und Verzeichnisstrukturen ist bei der Auswahl der zu übertragenden Daten notwendig. Die Methode eignet sich folglich primär zum Sichern von explizit ausgewählten Daten. Darüber hinaus empfiehlt Andrew Hoog, den Befehl nicht auf große Verzeichnisse anzuwenden (vgl. [Hoo11] S. 219). Zurückzuführen ist dies auf Fehler, die im Verlauf der Übertragung, z. B. aufgrund fehlender Berechtigungen, auftreten können. Um einen besseren Überblick über die angefertigten Sicherungen zu erhalten, sollte daher ein großes Verzeichnis durch mehrfache Anwendung des Befehls übertragen werden. Anschließend kann eine Sichtung der gewonnen Dateien innerhalb der Datensammlung erfolgen.

adb shell

Durch den Einsatz von `adb shell` und das in Android integrierte Werkzeug dd kann über das Smartphone eine Datensammlung durchgeführt werden. Der Ansatz wurde zuvor im Recovery Mode erwähnt und soll in diesem Bereich nochmals vertieft werden. Zur Erhebung der Sicherungen ist die Partitionierung des beschlagnahmten Geräts zu analysieren.

Anhand der Ausgabe von `mount` kann abgeleitet werden, dass sich unter dem Pfad `/dev/block` auf dem Smartphone die Rohdaten der relevanten Partitionen befinden. Auf diese kann ausschließlich mit privilegierten Rechten über `dd` zugegriffen werden. Grundvoraussetzung hierfür ist ein Flash-Speicher, der über einen integrierten Controller mit einem Flash Translation Layer verfügt. Die Bezeichnung der Rohdaten in Form von `mmcblkXpXX` lässt hierbei allerdings keine Schlüsse auf die jeweilige Partition zu. Für eine genaue Zuordnung und Auswahl sind neben der Ausgabe von `mount` die Unterverzeichnisse von `/dev/block/platform` zu sichten.

Ein Aushängen der zu sichernden Partitionen ist im laufenden Betrieb aufgrund verschiedener aktiver Prozesse oftmals nicht möglich. Die Berechnung von Hashwerten zum Nachweis der Integrität erweist sich infolgedessen als schwierig, da permanent Veränderungen an den bestehenden Daten stattfinden. Ein Hash sollte somit erst nach Anfertigung der Sicherung erstellt werden, um zumindest eine Integritätsverletzung in der Datenuntersuchung ausschließen zu können. Ferner sind die Daten von verschlüsselten Partitionen nur im eingehängten Zustand lesbar. Die eigentlichen Rohdaten liegen verschlüsselt vor, weshalb dem Befehl `dd` zur Sicherung stattdessen die Einhängepunkte der Partitionen zu übergeben sind.

Als Speicherort kann ein vorbereiteter externer Flash-Speicher oder die Workstation genutzt werden (vgl. [Hoo11] S. 279). Ist die Nutzung eines externen Flash-Speichers ausgeschlossen oder ein Zugriff auf dem im Gerät vorhanden externen Flash-Speichers notwendig, sollte eine Sicherung auf die Workstation mit dem Werkzeug „Netcat“ über die USB-Verbindung erfolgen. Hierzu sind eine Portweiterleitung durch `adb forward` auf dem Smartphone einzurichten und die Ausgabe des Befehls `dd` an `netcat` zu übergeben:

```
$ adb forward tcp:4444 tcp:4444

$ adb shell su

# dd if=/dev/block/mmcblk0p23 | netcat -l -p 4444
```

Abbildung 23: Sicherung des internen Flash-Speichers durch Verwendung von adb shell

Bei `mmcblk0p23` handelt es sich in diesem Fall um die Data-Partition des Smartphones.

Durch den Parameter `-l` wird eine Datenübertragung erst gestartet, wenn eine Verbindung zu einem Zielsystem besteht. Folglich sind in einer zweiten Shell auf der Workstation der Befehl `netcat` abzusetzen und der eingehende Datenstrom in eine Datei zu schreiben:

```
$ netcat 127.0.0.1 4444 > data.dd
```

Abbildung 24: Übertragung der Sicherung des internen Flash-Speichers mit Netcat

Die gewonnen Abbilder können anschließend einer logischen und physikalischen Datenuntersuchung unterzogen werden.

Im Kapitel „Datensammlung“ wurde die softwarebasierte Sicherung des internen Flash-Speichers vertieft. Zur Sicherung konnten die Ausgangspunkte „Recovery Mode“, „Content Provider“ und „Android Debug Bridge“ erarbeitet werden.

Ein im ausgeschalteten Betriebszustand beschlagnahmtes Smartphone ist stets über den Recovery Mode zu sichern. Erreicht wird dies durch die Erweiterung der Funktionalitäten des Modus um eine ADB mit privilegierten Rechten. Während im privaten Umfeld eine solche Erweiterung möglicherweise bereits durch den Gerätebesitzer vollzogen wurde, ist im geschäftlichen Umfeld in der der Regel eine Modifikation der Recovery-Partition notwendig. Anschließend kann, bei erfolgreicher Realisierung, eine Sicherung der Daten mit Hilfe des Werkzeugs dd über die ADB erfolgen.

Bei ausbleibendem Erfolg oder eingeschaltetem Smartphone ist auf Basis der Content Provider oder der Android Debug Bridge vorzugehen. Content Provider erfordern zwar keine privilegierten Rechte auf dem Smartphone, allerdings können nur jene Daten bezogen werden, die von den darauf befindlichen Applikationen bereitgestellt werden. Die Android Debug Bridge ermöglicht über `adb pull` oder `adb shell` in Kombination mit dem Befehl `dd` bei privilegierten Rechten eine ergiebigere Datensammlung. Diese kann ausschließlich unter Einsatz von Werkzeugen des Betriebssystems durchgeführt werden. Ein Wechsel in den Recovery Mode ist denkbar, wenn die mit den zuvor aufgeführten Methoden gewonnen Daten nicht ausreichend sind.

6 Anwendbarkeit

Die Datensammlung bildet die essentielle Basis für die weiterführende forensische Untersuchung. Aufgrund dieser Wichtigkeit wird abschließend die Anwendbarkeit der in dieser Ausarbeitung vorgestellten Handlungsanweisungen in Bezug auf die Datensammlung demonstriert. Hierzu erfolgt die Erstellung eines praxisnahen Beispielszenarios, in dem die forensische Workstation und das zu untersuchende Smartphone beschrieben werden. Das Szenario bildet die Grundlage für die beispielhafte Durchführung der Datensammlung. Im Zuge dessen werden anhand der Handlungsanweisungen die zu ergreifenden Methoden bestimmt, welche eine minimalinvasive Sicherung des vorliegenden Smartphones erlauben.

6.1 Beispielszenario

Die Vorgehensweisen zur Sicherung der Daten eines privaten und geschäftlichen Smartphones unterscheiden sich nur geringfügig. Typischerweise sind die Aufwände zur Durchführung im privaten Umfeld, z. B. hinsichtlich der Aufhebung eines vorhandenen Sperrbildschirms, deutlich höher. In diesem Beispielsszenario wird daher ausschließlich von einem privat genutzten Google Nexus 4 mit Android in der Version 4.2, Jelly Bean, ausgegangen. Das Smartphone, welches im eingeschalteten Zustand beschlagnahmt wurde, verfügt über einen unverschlüsselten internen Flash-Speicher. Die Nutzung eines externen Flash-Speichers ist aufgrund einer fehlenden Vorrichtung ausgeschlossen. Im Gerät befindet sich ferner eine SIM-Karte, die mit einer vierstelligen PIN geschützt ist. Ein aktivierter Sperrbildschirm in Form eines zu zeichnenden Musters verhindert darüber hinaus die direkte Interaktion mit dem eingeschalteten Smartphone. Während der Bestandsaufnahme konnten das Google-Benutzerkonto aus einem Dokument mit einer Auflistung von Benutzernamen inklusive Passwörtern des Besitzers sowie PIN und PUK aus einem Anschreiben des Providers entnommen werden.

Zur forensischen Untersuchung des Smartphones stehen eine mit „Lubuntu 12.10“ und eine mit „Windows 7“ ausgestattete virtuelle Maschine mit jeweils 4 GB Arbeitsspeicher sowie 20 GB Festplattenspeicher zur Verfügung. Bei Lubuntu handelt es sich um ein offizielles und ressourcenschonendes Derivat des auf Linux basierenden Betriebssystems „Ubuntu“. Für die Verwendung von Linux spricht vor allem das umfassende Berechtigungssystem, welches ausschließlich dem Benutzer `root` einen vollständigen Systemzugriff gestattet.

Dieser kann von berechtigten Benutzern standardmäßig nur durch Anfügen des Ausdrucks `sudo` vor einem auszuführenden Befehl und die anschließende Eingabe eines Passworts verwendet werden. Darüber hinaus sind in Linux bereits Werkzeuge zur Datensammlung, wie z. B. dd, integriert. Demnach muss die virtuelle Maschine lediglich um das Android SDK zur Interaktion mit dem Smartphone und Wireshark zur Überwachung der Kommunikation über die USB-Schnittstelle ergänzt werden. Der Einsatz von Windows ist in diesem Kontext erforderlich, um eine Sicherung der SIM-Karte durchzuführen. Die Synchronisation der Zeit erfolgt auf beiden virtuellen Maschinen über NTP, das bei der Systemeinrichtung aktiviert wurde. Ferner ist zur Vermeidung einer unbeabsichtigten Datenveränderung das automatische Einbinden angeschlossener Speichermedien deaktiviert.

6.2 Datensammlung

Auf Basis der zuvor beschriebenen Ausgangsbedingungen in Bezug auf die forensische Workstation und das zu untersuchende Smartphone wird nachstehend beispielhaft eine Datensammlung durchgeführt. Diese untergliedert sich, angelehnt an das entwickelte Prozessmodell, in die Bereiche **Verbindungsaufbau**, **Sicherung RAM-Speicher**, **Sicherung Flash-Speicher (intern)** und **Sicherung SIM-Karte**. Um einem möglichen Datenverlust vorzubeugen, ist im eingeschalteten Betriebszustand nach dem erfolgreichen Verbindungsaufbau entsprechend der im Modell beschriebenen Sicherungsreihenfolge mit den flüchtigen Speichermedien, d. h. dem RAM, zu beginnen. Daran knüpfen die Sicherung des internen Flash-Speichers und der SIM-Karte an (siehe Abbildung 25):

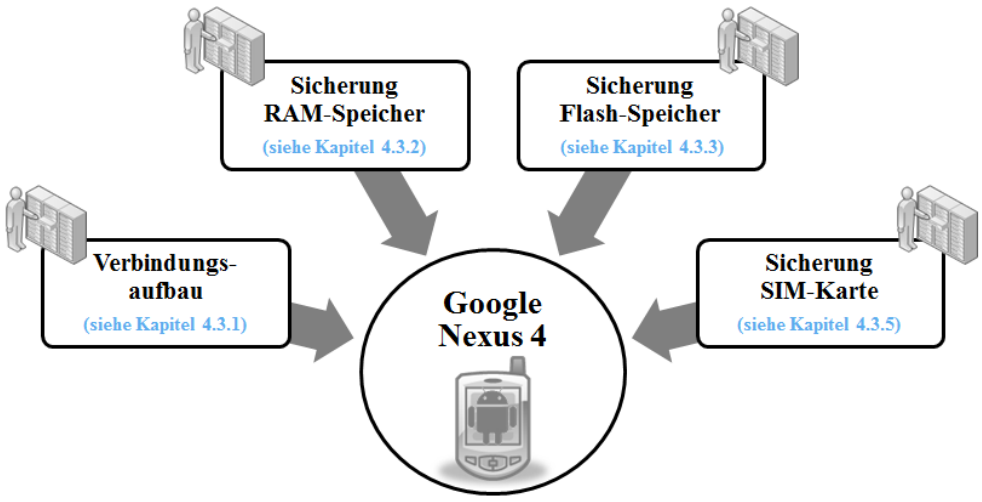

Abbildung 25: Anwendung der Datensammlung auf das Google Nexus 4

Ein besonderes Augenmerk wird während der Datensammlung aufgrund seiner Bedeutung innerhalb einer forensischen Untersuchung auf den internen Flash-Speicher gelegt.

6.2.1 Verbindungsaufbau

Vor dem Verbindungsaufbau wird empfohlen, zwischen forensischer Workstation und dem Smartphone einen hardwarebasierten Writeblocker von den im Bereich der IT-Forensik etablierten Herstellern „WiebeTech“ oder „Tableau“ anzubringen. Hierbei ist zu berücksichtigen, dass ein Schutz lediglich bei Einbindung des Smartphones als Massenspeicher gewährleistet werden kann. Aufgrund dieser eingeschränkten Funktionalität wurde auf den gesonderten Kauf eines solchen Systems speziell für die Umsetzung dieser Ausarbeitung verzichtet. Zur Sichtung möglicher Schreibvorgänge wird jedoch, wie vorgegeben, der Kommunikationsverlauf zwischen forensischer Workstation und dem Smartphone mittels Wireshark aufgezeichnet. Für diesen Zweck ist es notwendig, zuvor das im Betriebssystem der forensischen Workstation vorhandene Kernelmodul `usbmon` durch Eingabe des Befehls `sudo modprobe usbmon` in einer Shell zu laden [Wir13]. Das Kernelmodul ist Bestandteil von Lubuntu und wird standardmäßig zum Debuggen der USB-Schnittstelle bereitgestellt. Nach erfolgreicher Einbindung ist Wireshark mit privilegierten Rechten durch `sudo wireshark` zu starten und die Aufzeichnung der Kommunikation durchzuführen.

Sperrbildschirm

Nach Einschaltung von Wireshark kann der Verbindungsaufbau zwischen Smartphone und forensischer Workstation erfolgen. Zur Überprüfung der Verbindung wird mit dem Befehl `adb devices` der Status des USB-Debuggings getestet. Die fehlende Auflistung des Smartphones in der Shell der Workstation lässt hierbei auf ein deaktiviertes USB-Debugging schließen. Eine Aktivierung erfordert einen Zugriff auf die Entwickleroptionen und somit die Aufhebung des vorhandenen Sperrbildschirms. Durch das vorhandene Google-Benutzerkonto eignen sich hierzu entsprechend der in dieser Ausarbeitung bereitgestellten Handlungsanweisungen „Smudge Attack“ und „Google-Benutzerkonto“. Zwar ist Smudge Attack aufgrund der Beibehaltung der Abschirmung zu präferieren, jedoch wird wegen zeitlicher und technischer Einschränkungen eine Aufhebung des Sperrbildschirms mittels des Google-Benutzerkontos durchgeführt.

Zur Aufhebung ist die bestehende Abschirmung zu Beginn zu entfernen, sodass das Smartphone eine Verbindung zum Google Server aufbauen kann. Damit die Option „Muster vergessen“ erscheint, wird beim Google Nexus 4 eine fünfmalige Falscheingabe getätigt. Angekündigt wird die mehrfache Falscheingabe durch eine gerätespezifische Meldung auf dem Bildschirm. Anschließend können nach Auswahl von „Muster vergessen“ der Benutzername und das Passwort des auf dem Smartphone aktiven Google-Benutzerkontos eingegeben werden. Durch das Drücken des Felds „Anmelden“ erfolgt schließlich die Authentifizierung des Benutzerkontos am Google Server (siehe Abbildung 26).

Abbildung 26: Aufhebung des Sperrbildschirms des Google Nexus 4

Nach erfolgreicher Authentifizierung ist entweder ein neuer Sicherheitscode oder eine einfache Bewegung zum Aufheben des Sperrbildschirms zu definieren. Zum Schutz des Smartphones wird, neben der sofortigen Wiederherstellung der Abschirmung, ein neuer Sicherheitscode hinterlegt. Darauf folgend kann das USB-Debugging für die Durchführung der Datensammlung aktiviert werden.

Berechtigungen

Zur Durchführung einer umfassenden Datensammlung, insbesondere zur Sicherung des RAMs, werden privilegierte Rechte benötigt. Die Rückmeldung „Permission denied“ nach Absetzen des Befehls `adb shell su` in einer auf der forensischen Workstation geöffneten Shell lässt darauf schließen, dass der Besitzer das Smartphone keinem Root-Vorgang unterzogen hat.

Zur Ermittlung einer geeigneten Methode zum Rooten des Google Nexus 4 wird, wie empfohlen, eine tiefgehende Recherche im Forum XDA Developers durchgeführt. Gemäß der dort ermittelten Informationen existieren zum gegenwärtigen Zeitpunkt ausschließlich Lösungen, die einen Systemneustart voraussetzen. Aus diesem Grund sollte das Rooten des Google Nexus 4 erst in Betracht gezogen werden, wenn die gesicherten Daten zur Aufklärung des Vorfalls nicht ausreichend sind. Hierbei wird eine Methode fokussiert, die im Vergleich zu anderen existierenden Ansätzen nur geringfügige Manipulationen vornimmt und keine Entsperrung des Bootloaders erfordert, sodass eine Wiederherstellung der Werkseinstellungen ausgeschlossen ist.

Die Skripte der ausgewählten Methode zum Rooten des Google Nexus 4 können im Forum von XDA Developers bezogen werden [XDA13]. Bei Ausführung der Skripte wird eine Schwachstelle im Chipsatz des Smartphones genutzt. Angelehnt an die Beschreibungen aus der „National Vulnerability Database“ des NIST kommt es in einer spezifischen Funktion des Chipsatzes zu einem „Ganzzahlüberlauf[3]“ [Nat13]. Aus dem Ganzzahlüberlauf resultiert ein schreibender Zugriff auf den flüchtigen Speicherbereich des Kernels und somit die Möglichkeit, Systembefehle mit privilegierten Rechten auszuführen. Auf diese Weise wird die System-Partition des Smartphones mit Schreibrechten eingehängt und die Datei `su` im Systemverzeichnis `/system/xbin` abgelegt. Die Datei stellt den aus Linux bekannten Befehl `su` bereit, der standardmäßig, ohne die Ergänzung weiterer Parameter, einen Benutzerwechsel zu `root` innerhalb einer laufenden Shell ermöglicht. Damit `su` als einfacher Benutzer ausgeführt werden kann, wird durch `chmod 4755 /system/xbin/su` das Bit `SetUserID` gesetzt. Mit Hilfe des Bits kann jeder auf dem System befindliche Benutzer die Datei mit den Rechten des Eigentümers, in diesem Fall `root`, ausführen. Der Root-Vorgang wird zuletzt durch einen Neustart des Smartphones abgeschlossen, wodurch die Manipulationen im flüchtigen Speicherbereich des Kernels verloren gehen.

[3] Bei einem Ganzzahlüberlauf wird der gegebene Zahlenraum überschritten. Eine solche Überschreitung entsteht durch eine arithmetische Operation, deren Ergebnis einen Zahlenwert annimmt, der zu groß ist, also mehr Stellen aufweist, als im ganzzahligen Datentyp gespeichert werden kann. Die überzähligen Stellen führen letztlich zu fehlerhaften Zahlenwerten.

6.2.2 Sicherung RAM-Speicher

Zur Datensammlung des RAMs mit den verfügbaren Methoden sind privilegierte Rechte auf dem Smartphone erforderlich. Der Systemneustart, der mit dem Rooten des Google Nexus 4 einhergeht, hat jedoch den Verlust der flüchtigen Speicherinhalte zur Folge. Eine Sicherung des RAMs erübrigt sich in diesem Zusammenhang. Ergibt sich jedoch aufgrund der schnelllebigen technischen Entwicklungen in Zukunft eine Möglichkeit, privilegierte Rechte auf dem Smartphone ohne Systemneustart zu erlangen, kann eine Sicherung des RAMs mit LiME durchgeführt werden. Der alternative Ansatz auf Basis des Befehls `kill` ist beim Google Nexus 4 ausgeschlossen, da keine Erstellung von Dumps beim Beenden von laufenden Prozessen erfolgt. Der mit LiME erzeugte Dump muss aufgrund der fehlenden Unterstützung eines externen Flash-Speichers beim Google Nexus 4 über TCP und die bestehende USB-Verbindung auf die forensische Workstation übertragen werden. Hierzu ist, wie in den Handlungsanweisungen dieser Ausarbeitung beschrieben, bei der Einbindung des Kernelmoduls im Parameter `path` das Übertragungsprotokoll sowie der zugehörige Port und im Parameter `format` der präferierte Formattyp anzugeben (siehe Abbildung 27):

```
# insmod lime.ko "path=tcp:4444 format=lime"
```

Abbildung 27: Sicherung des RAMs des Google Nexus 4 mit Linux Memory Extractor

Damit die forensische Workstation den Dump empfängt, muss der Befehl `netcat` in einer geöffneten Shell eingegeben und der eingehende Datenstrom in eine Datei, wie z.B. `RAM.lime`, geschrieben werden (siehe Abbildung 28):

```
$ netcat 127.0.0.1 4444 > RAM.lime
```

Abbildung 28: Übertragung der Sicherung des RAMs unter Verwendung von Netcat

Anschließend kann die Sicherung durch Einsatz des Volatility Frameworks einer physischen Datenuntersuchung unterzogen werden. Zuvor ist jedoch entsprechend der Vorgaben innerhalb der Datenuntersuchung zur Gewährleistung der Integrität ein kryptografischer Hashwert mittels `sha1sum` über die Daten der Sicherung zu bilden.

6.2.3 Sicherung Flash-Speicher (intern)

Die Methode zur Sicherung des internen Flash-Speichers ist abhängig von den zu Grunde liegenden Rechten auf dem Google Nexus 4. Da ein Root-Vorgang auf dem Smartphone einen Systemneustart erfordert, ist zu Beginn eine Datensammlung entsprechend der Empfehlungen in dieser Ausarbeitung unter Verwendung der im Betriebssystem verfügbaren Content Provider durchzuführen. Auf diese Weise können Daten bezogen werden, die von den Applikationen bereitgestellt werden. Dies beinhaltet auch Daten der Data-Partition, welche über die Android Debug Bridge ausschließlich mit privilegierten Rechten zugänglich sind. Die Durchführung des Root-Vorgangs und die Sicherung mit Hilfe der Android Debug Bridge werden empfohlen, wenn die bereits gewonnen Daten zur Aufklärung des vorliegenden Vorfalls nicht ausreichend sind. Der Recovery Mode, welcher aufgrund des Systemneustarts in Frage kommt, stellt wegen eines ab Werk gesperrten Bootloaders beim Google Nexus 4 keine Alternative dar.

Eingeschränkte Rechte

Die Datensammlung auf Basis der Content Provider erfolgt aus Kostengründen mit dem bereits angeführten Werkzeug AFLogical von viaForensics. Mit Hilfe einer auf der forensischen Workstation geöffneten Shell ist die Installation der Applikation auf dem Smartphone durchzuführen (siehe Abbildung 29):

```
$ adb install AFLogical_1.5.2.apk

205 KB/s (28794 bytes in 0.136s)

    pkg: /data/local/tmp/AFLogical_1.5.2.apk

Success
```

Abbildung 29: Installation von AFLogical auf dem Google Nexus 4

Der interne Flash-Speicher des Geräts, insbesondere die Data-Partition, wird infolgedessen um die Daten der Applikation ergänzt. Entweder durch die Auswahl der Applikation auf dem Smartphone oder den Einsatz der Shell auf der forensischen Workstation kann anschließend die Datensammlung gestartet werden.

Um die Interaktion mit dem Gerät auf ein Mindestmaß zu begrenzen, wird zur Applikationsausführung die Variante auf Basis der Shell empfohlen (siehe Abbildung 30):

```
$ adb shell am start -n com.viaforensics.android.aflogical_
ose/com.viaforensics.android.ExtractAllData

Starting: Intent {cmp=com.viaforensics.android.aflogical_
ose/com.viaforensics.android.ExtractAllData}
```

Abbildung 30: Sicherung des Flash-Speichers des Google Nexus 4 mit AFLogical

Der Parameter `-n` enthält hierbei den Namen des Packages `com.viaforensics.android.aflogical_ose` und – mittels eines Schrägstrichs separiert – die auszuführende Aktivität `com.viaforensics.android.ExtractAllData` der Applikation. Die Aktivität `ExtractAllData` veranlasst die Sicherung aller über die Content Provider verfügbaren Daten. Diese werden standardmäßig im Verzeichnis `/sdcard/forensics` auf dem internen Flash-Speicher des Google Nexus 4 abgelegt. Darunter befinden sich beispielsweise gespeicherte Kontaktdaten, Kurzmitteilungen, E-Mails und Anruflisten. Zur Datenuntersuchung ist, neben der Ausführung von `sha1sum` zur Berechnung von Hashwerten über die Daten, eine Übertragung auf die forensische Workstation vorzunehmen:

```
$ adb pull /sdcard/forensics/ /home/dmuth/FlashIntern

pull: building file list...
```

Abbildung 31: Übertragung der Sicherungen des Flash-Speichers mittels adb pull

Privilegierte Rechte

Für die Durchführung der Datensammlung auf Grundlage der Android Debug Bridge wird das Gerät zuvor gemäß der im Verbindungsaufbau beschriebenen Methode einem Root-Vorgang unterzogen. Mit Hilfe der privilegierten Rechte ist es möglich, ein vollständiges physikalisches 1:1-Abbild der bestehenden Partitionen anzufertigen. Zu Beginn wird hierzu über die forensische Workstation eine Shell auf dem Smartphone geöffnet und die zu Grunde liegende Partitionierung analysiert.

Die folgende Abbildung enthält einen Ausschnitt der auf dem Google Nexus 4 eingehängten Partitionen, welche unter Verwendung von `mount` eingesehen werden können:

```
$ mount

rootfs / rootfs ro,relatime 0 0

tmpfs /dev tmpfs rw,nosuid,relatime,mode=755 0 0

/dev/block/platform/msm_sdcc.1/by-name/system /system ext4
ro,relatime,data=ordered 0 0

/dev/block/platform/msm_sdcc.1/by-name/cache /cache ext4
rw,nosuid,nodev,noatime,data=ordered 0 0

/dev/block/platform/msm_sdcc.1/by-name/userdata /data ext4
rw,nosuid,nodev,noatime,noauto_da_alloc,data=ordered 0 0
```

Abbildung 32: Eingehängte Partitionen im Google Nexus 4

Durch die tiefgehende Sichtung des in der vorherigen Abbildung angegebenen Verzeichnisses `/dev/block/platform/msm_sdcc.1/by-name` auf dem Gerät kann der tatsächliche Speicherort der Rohdaten der jeweiligen Partitionen ermittelt werden:

```
$ ls -l

… 2013-07-29 12:58 system   -> /dev/block/mmcblk0p21

… 2013-07-29 12:58 cache    -> /dev/block/mmcblk0p22

… 2013-07-29 12:58 userdata -> /dev/block/mmcblk0p23
```

Abbildung 33: Speicherort der Rohdaten der Partionen des Google Nexus 4

Eine Grundvoraussetzung für den Zugriff auf die angegebenen Rohdaten mit dd sind privilegierte Rechte auf dem Smartphone. In diesem Beispiel werden der gesamte interne Flash-Speicher durch Angabe von `mmcblk0` und explizit die Data-Partition, also `mmcblk0p23`, gesichert.

Die gesonderte Sicherung von Data soll während der Datenuntersuchung das Einhängen erleichtern, da keine Anfangs- und Endadresse zur Bestimmung der Partition im gesamten Abbild des Flash-Speichers berechnet werden muss. Aufgrund des fehlenden externen Speichers erfolgt eine Übertragung der Daten auf die forensische Workstation. Entsprechend dieser Vorgaben und der dargelegten Handlungsanweisungen sieht die Umsetzung wie folgt aus (siehe Abbildung 34):

Sicherung des gesamten Flash-Speichers:

```
(Shell 1)$ adb forward tcp:4444 tcp:4444

(Shell 1)$ adb shell su

(Shell 1)# dd if = /dev/block/mmcblk0 | netcat –l –p 4444

(Shell 2)$ netcat 127.0.0.1 4444 > FlashIntern.dd
```

Sicherung der Data-Partition des Flash-Speichers:

```
(Shell 1)$ adb forward tcp:4444 tcp:4444

(Shell 1)$ adb shell su

(Shell 1)# dd if = /dev/block/mmcblk0p23 | netcat –l –p 4444

(Shell 2)$ netcat 127.0.0.1 4444 > data.dd
```

Abbildung 34: Sicherung des Flash-Speichers des Google Nexus 4 mit adb shell

Über die Daten der gewonnenen Sicherungen sind anschließend mit `sha1sum` Hashwerte zu bilden. Diese dienen während der nachfolgenden Datenuntersuchung als Nachweis, dass bei der Extraktion der potentiellen Spuren keine Veränderungen an den Daten vorgenommen wurden.

6.2.4 Sicherung SIM-Karte

Die SIM-Karte des Google Nexus 4 kann, wie in der Ausarbeitung angeführt, durch die Kenntnis von PIN sowie PUK ausgebaut und mit einem Kartenleser ausgelesen werden.

Die hierfür geeigneten forensischen Werkzeuge setzen vorwiegend den Einsatz von Windows voraus, weshalb eine zusätzliche virtuelle Maschine zur Verfügung steht. Die Ermittlung eines geeigneten Werkzeugs erfolgt auf Basis der Empfehlungen des NIST (vgl. [Nat07] S. 17-18). Hierbei wird innerhalb dieser Ausarbeitung zur Kostenersparnis eine Lösung präferiert, die für einen dedizierten Zeitraum kostenlos zu Testzwecken erhältlich ist. Demgemäß wird die Sicherung der vollständigen Speicherinhalte der SIM-Karte unter Einsatz des im forensischen Umfeld bekannten Werkzeugs „SIM Card Seizure“ des Herstellers „Paraben Corporation“ durchgeführt. SIM Card Seizure besteht aus einem vom Hersteller bereitgestellten Kartenleser mit zugehöriger Software. Der Kartenleser kann hierbei auch durch ein beliebiges Gerät eines Drittanbieters ersetzt werden. Zur Datensammlung ist es erforderlich, nach erfolgreicher Installation der Software auf der mit Windows ausgestatteten virtuellen Maschine den Kartenleser und die zu sichernden Daten der SIM-Karte auszuwählen. Neben Daten, wie Anruflisten und Kurzmitteilungen, kann hierbei auch das gesamte Dateisystem ausgelesen werden (siehe Abbildung 35):

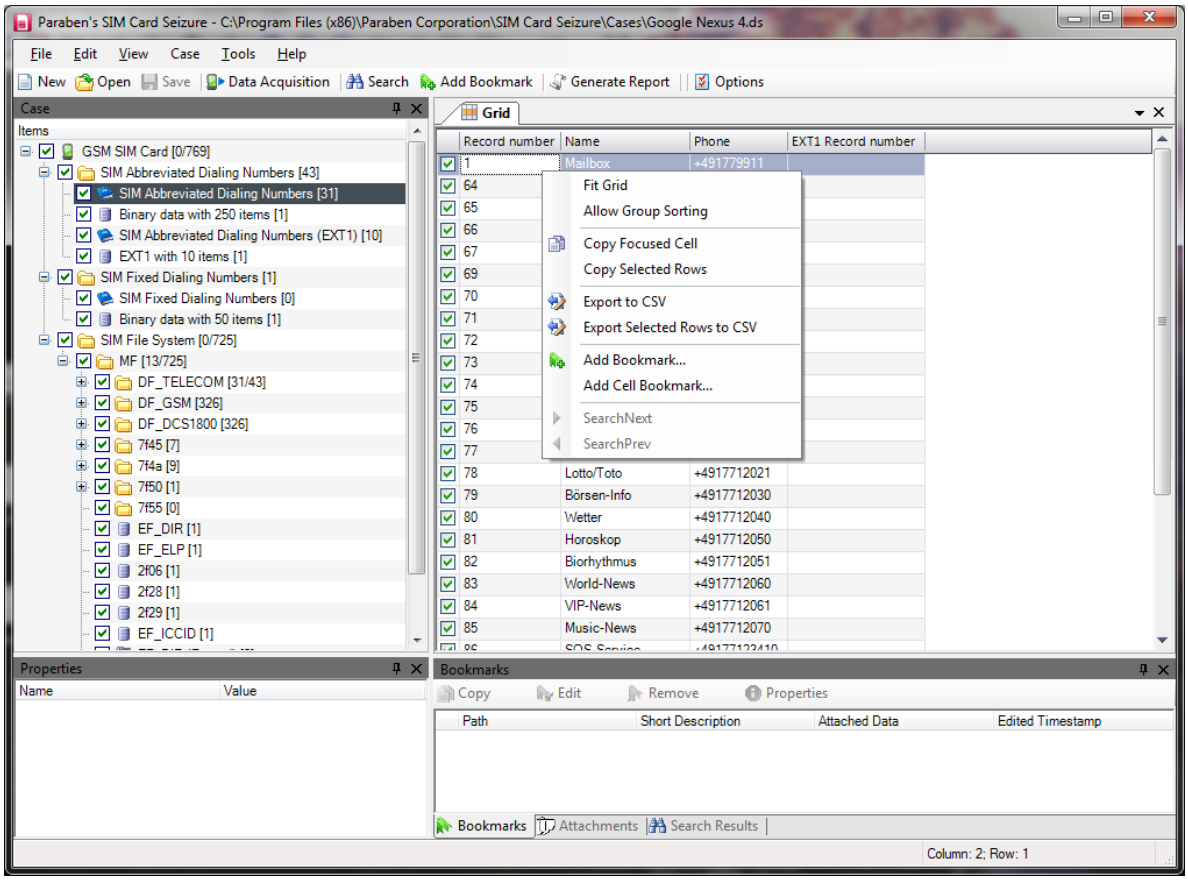

Abbildung 35: Sicherung der SIM-Karte des Google Nexus 4 mit SIM Card Seizure

Die Anfertigung der Sicherung erfordert zuvor die Eingabe der PIN oder nach deren dreimaliger Falscheingabe der PUK der SIM-Karte. Anschließend befinden sich die Daten in aufbereiteter Form in SIM Card Seizure und können einer tiefgehenden Untersuchung unterzogen werden.

Im Kapitel „Anwendbarkeit“ wurde die Anwendung der in dieser Ausarbeitung erarbeiteten Handlungsanweisungen zur Durchführung einer minimalinvasiven Datensammlung demonstriert. Die Basis hierzu bildete ein privates Google Nexus 4 mit Android 4.2, das sich zum Zeitpunkt der Bestandsaufnahme im eingeschalteten Betriebszustand befand und einen aktiven Sperrbildschirm in Form eines Musters aufwies. Ferner besaß das Gerät eine geschützte SIM-Karte, deren PIN und PUK anhand eines Schreibens des Providers ermittelt werden konnte. Unter Einsatz einer mit Lubuntu 12.10 und einer mit Windows aufgesetzten virtuellen Maschine erfolgte die beispielhafte Datensammlung. Hierbei konnten die beschriebenen Handlungsanweisungen in Bezug auf den Verbindungsaufbau, die Sicherung des RAMs, des internen Flash-Speichers sowie des Speichers der SIM-Karte praktisch umgesetzt und deren Funktionsweise sowie Plausibilität mit Erfolg dargelegt werden.

7 Resümee

Das Resümee fasst den Kern der Masterarbeit zusammen und beleuchtet die erlangten Ergebnisse. Hierzu werden nochmals die wesentlichen Gesichtspunkte der Motivation und Zielsetzung herausgearbeitet. Darauf aufbauend erfolgt die Darstellung des Ergebnisses dieser Ausarbeitung, welches auf die Beantwortung der zu Beginn gestellten zentralen Fragestellungen zu überprüfen ist. Zuletzt wird ein Ausblick gegeben, in dem denkbare Folgearbeiten aufgezeigt werden, die sich im Zuge der Untersuchungen herauskristallisiert haben. Abgerundet wird diese Masterarbeit durch ein persönliches Fazit hinsichtlich der forensischen Untersuchung von Android-Smartphones.

7.1 Rückblick

Smartphones gewinnen durch ihre vielseitigen Einsatzmöglichkeiten, die auf eine leistungsstarke Hardware, hohe Speicherkapazitäten und ein offenes Betriebssystem zurückzuführen ist, zunehmend im Bereich der IT-Forensik an Wichtigkeit. Entsprechend ihrer Stellung am Markt sind hierbei vor allem Geräte basierend auf Android bedeutsam. Die darauf gespeicherten Daten dienen als potentielle Spuren, die innerhalb eines Ermittlungsverfahrens zur Aufklärung einer Straftat genutzt werden können. Allerdings weichen die Methoden zur Untersuchung aufgrund der besonderen technischen Gegebenheiten der Systeme von denen der klassischen IT-Forensik ab. Bisherigen Ausarbeitungen auf diesem Gebiet fehlt entweder der technische Fokus auf Android oder eine prägnante und an anerkannte Prozessmodelle angelehnte Systematik zur Untersuchung. Die Folge ist im ungünstigsten Fall eine falsche Handhabung der Smartphones, die zum Verlust von essentiellen Beweismitteln führt.

Ziel der Masterarbeit war es, eine systematische und folglich ordnungsgemäße forensische Behandlung der aktuellen auf Android basierenden Smartphones zu ermöglichen. Der Schwerpunkt lag hierbei auf der Datensammlung, da diese die Basis für sämtliche weiterführenden Untersuchungen darstellt. Dabei sollte keine reine Auflistung von Methoden erfolgen, sondern vielmehr ein Leitfaden vergleichbar zum Ansatz „IT-Forensik“ des BSI mit praxisnahen Handlungsanweisungen zum angemessenen Umgang mit den Smartphones bereitgestellt werden. Darüber hinaus war angestrebt, mit Hilfe des praktischen Bezugs ein Nachschlagewerk zur Lösung gegenwärtiger Problemstellungen, vor allem im Hinblick auf die Datensammlung, zu schaffen.

7.2 Ergebnis

Mit dieser Masterarbeit wird ein Leitfaden zur forensischen Untersuchung von Smartphones basierend auf Android zur Verfügung gestellt. Das allgemeine Prozessmodell, welches auf anerkannten Ansätzen von BSI, NIST sowie Alexander Geschonneck beruht, greift die mit den Geräten einhergehenden Besonderheiten bezüglich der IT-Forensik auf. Die übersichtliche Einteilung in sechs Prozessabschnitte, deren Inhalte jeweils einzelne Bausteine bilden, erlaubt eine schnelle Erfassung des gesamten Untersuchungsablaufs. Die Prozessbausteine des Modells, die um die aus Android resultierenden spezifischen forensischen Aspekte ergänzt wurden, beleuchten den aktuellen Stand der Technik sowie elementare Problemstellungen und Lösungsansätze auf diesem Gebiet. Ein besonderes Augenmerk liegt hierbei auf der Datensammlung und der Sicherung des internen Flash-Speichers mit Hilfe von softwarebasierten Methoden. Auf Basis eines praktischen Beispielszenarios werden Funktionsweise sowie Plausibilität der Handlungsanweisungen aus der Datensammlung erfolgreich demonstriert. Die in der Zielsetzung gestellten zentralen Fragestellungen können schließlich mit den erlangten Ergebnissen in dieser Masterarbeit beantwortet werden:

- Wie können Android-Smartphones systematisch einer forensischen Untersuchung unterzogen werden?
- Wie kann eine forensisch korrekte und umfassende Datensammlung erfolgen, welche die Integrität und die gerichtliche Verwertbarkeit der digitalen Spuren gewährleistet?
- Welche Herausforderungen ergeben sich bei der forensischen Untersuchung, im Speziellen bei der Datensammlung, und wie können diese ggf. bewältigt werden?
  - Wie können Daten des Android-Smartphones bezogen werden und welche Auswirkungen, besonders auf die Integrität und gerichtliche Verwertbarkeit, haben notwendige Manipulationen am Gerät zur Folge?
  - Wie können die Daten extrahiert und gesichert werden, wenn das Android-Smartphone durch verschiedene Schutzmaßnahmen, wie z. B. einen Sperrbildschirm oder eine Verschlüsselung, geschützt ist?
- Welche Daten sind auf Android-Smartphones von Interesse und können innerhalb einer Untersuchung ermittelt werden?

- Welche existierenden Werkzeuge sind für den Einsatz zur forensischen Untersuchung, im Speziellen zur Datensammlung, von Android-Smartphones geeignet?

Die im Leitfaden erarbeiteten Handlungsanweisungen ermöglichen die systematische Durchführung einer forensischen Untersuchung von auf Android basierenden Smartphones. Dadurch kann, wie beispielhaft demonstriert, eine ordnungsgemäße Handhabung der Smartphones sichergestellt und so einem Verlust von essentiellen Beweismitteln entgegengewirkt werden. Die Berücksichtigung des Betriebszustands und der aktivierten Schutzmaßnahmen während der Datensammlung erlauben darüber hinaus die Anfertigung von Sicherungen mit einem möglichst geringen Einfluss auf die bestehenden Daten und deren Integrität sowie gerichtliche Verwertbarkeit. Mit Hilfe des praktischen Bezugs wird ebenfalls ein Nachschlagewerk zur Lösung gegenwärtiger Problemstellungen, vor allem in Bezug auf die Datensammlung, zur Verfügung gestellt. Schließlich können IT-Dienstleister aus den im Leitfaden aufgezeigten Methoden Sicherheitsrisiken hinsichtlich Android ableiten, die bei Erstellung eines Sicherheitskonzepts für Kunden zu berücksichtigen sind.

7.3 Ausblick

Im Zuge der Untersuchungen kristallisierten sich weitere Aufgaben heraus, die in Folgearbeiten umgesetzt werden können.

Demnach stellt, neben dem Marktführer Android, das von Apple entworfene iOS ein wichtiges Betriebssystem im Smartphone-Umfeld dar. Auf Basis des erarbeiteten allgemeinen Prozessmodells könnte der Leitfaden um die spezifischen technischen Aspekte von iOS ergänzt werden. Eine Herausforderung stellen in diesem Zusammenhang vor allem die im Vergleich zu Android restriktiveren Schutzmaßnahmen des Betriebssystem dar, welche die Durchführung der Datensammlung deutlich stärker beeinflussen werden.

Ferner fokussiert der Leitfaden „IT-Forensik“ vom BSI bisher ausschließlich IT-Systeme wie traditionelle Computer. Unter Verwendung dieser Masterarbeit kann, vor allem aufgrund der starken Anlehnung des allgemeinen Prozessmodells an die Strukturen des BSI, der veröffentlichte Leitfaden erweitert werden. Durch die Erweiterung des Ansatzes um die besonderen technischen Gegebenheiten von Smartphones, insbesondere mit dem Betriebssystems Android, wird schließlich dessen Anwendungsbereich vergrößert.

Zuletzt finden Smartphones zunehmend Einzug im geschäftlichen Umfeld. Daraus resultieren steigende Kundenanfragen bezüglich geeigneter Konzepte zur Absicherung der Geräte und der darauf befindlichen Daten. Wie bereits angeführt können aus den im Leitfaden enthaltenen Methoden, besonders im Bereich der Datensammlung, potentielle Sicherheitsrisiken abgeleitet werden. Auf Basis dessen ist schließlich die Erstellung eines generischen Sicherheitskonzepts hinsichtlich Android-Smartphones denkbar.

7.4 Fazit

Die forensische Untersuchung von Android-Smartphones stellt ein spannendes und zugleich komplexes Themengebiet dar, welches in Zukunft eine herausragende Rolle bei der Aufklärung von Straftaten einnehmen wird. Hierbei ist zu berücksichtigen, dass die Untersuchung nicht nur ein Umdenken, z. B. bei der Überführung in ein Labor, erfordert. Aufgrund der fehlenden Möglichkeit zum Ausbau der Speichermedien und der damit verbundenen Abhängigkeit zum Betriebssystem ist bei Smartphones eine sehr hohe Variabilität in Bezug auf die Durchführung der Datensammlung gegeben. Beeinflusst wird die Durchführung, wie in der Masterarbeit gezeigt, besonders durch den Betriebszustand und betriebssystemspezifische Schutzmaßnahmen. Unter Android sind, neben einem aktivierten USB-Debugging, vor allem privilegierte Rechte für einen umfassenden Datenzugriff entscheidend. So ist beispielsweise die Sicherung des RAMs ohne entsprechende Rechte ausgeschlossen. Bei dem internen Flash-Speicher kann kein direkter Zugriff auf die Data-Partition, die mitunter die wichtigsten Daten in Android enthält, erfolgen. Die Methoden zur Aktivierung des USB-Debugging und die Erweiterung der Rechte entgegen der Schutzmaßnahmen von Android erfordern weitreichende technische Kenntnisse. Darüber hinaus müssen die Methoden den forensischen Aspekten genügen und folglich die Gewährleistung der Integrität zur Aufrechterhaltung der Beweiskraft sicherstellen. Die Gefahr einer falschen Handhabung der Geräte und infolgedessen eines Verlusts von essentiellen Beweismitteln ist im Vergleich zur forensischen Untersuchung von Computern somit deutlich höher. Umso wichtiger ist ein Leitfaden, der diese Variabilität erfasst und geeignete Handlungsanweisungen aufzeigt. Mit dem in dieser Masterarbeit erarbeiteten Leitfaden kann dem Forensiker zukünftig eine essentielle Basis bereitgestellt werden, die trotz der gegebenen Herausforderungen eine erfolgreiche forensische Untersuchung von Android-Smartphones erlaubt.

Anhang A Ablaufschema „Geschonneck“

Das in Kapitel 3.1.3 aufgeführte Modell von Alexander Geschonneck wurde zur besseren Vergleichbarkeit auf die wesentlichen Inhalte reduziert. Das detaillierte Ablaufschema zur forensischen Untersuchung von Smartphones sieht wie folgt aus (vgl. [Ges11] S. 283):

Abbildung 36: Ablaufschema von Alexander Geschonneck (vgl. [Ges11] S. 283)

Das vorgegebene Ablaufschema weist, wie in Kapitel 3.1.3 „Modell „Geschonneck”“ und Abbildung 36 dargestellt, eine Unstimmigkeit in Bezug auf die Abschirmung des Smartphones auf. Diese ist nach Erhebung, Untersuchung und Analyse des Speichers der SIM-Karte nicht erforderlich. Allerdings kommt es durch das Einschalten des Geräts mit eingesetzter SIM-Karte üblicherweise zum Aufbau von Datenverbindungen, die maßgebliche Inhalte des internen Flash-Speicher modifizieren und schließlich zur Verletzung der Integrität der Daten führen können. Demzufolge sollte immer eine Abschirmung des Geräts, unabhängig von dessen Betriebszustand, erfolgen.

Anhang B Glossar

Android-Manifest Das Android-Manifest enthält essentielle Informationen über die zugehörige Applikation [And13h]. Diese Informationen benötigt das Betriebssystem vor der eigentlichen Installation und Ausführung der Applikation. Beispielsweise werden darin Berechtigungen festgelegt, die angeben, welche Ressourcen außerhalb der Sandbox (z. B. vom Betriebssystem oder anderen Applikationen) benötigt werden.

Beweismittel Mit Hilfe von Beweismitteln wird im Zuge eines Gerichtsverfahrens „der Beweis des Vorliegens oder Nichtvorliegens einer Tatsache geführt“ [BBK+13].

Block Device Bei einem Block Device handelt es sich um ein Speichermedium, welches Daten in Blöcken fester Größe verarbeitet. Typischerweise haben die Blöcke hierbei eine Größe von 512 Byte. Ein weit verbreiteter Vertreter auf diesem Gebiet ist die Festplatte.

Chain of Custody Die Chain of Custody bezieht sich auf die lückenlose Dokumentation „des Verbleibs der Beweismittel und deren Einsichtnahme“ (siehe [Bun11] S. 90). Dadurch soll fortwährend nachvollziehbar sein, welche Person zu welchem Zeitpunkt auf die Beweismittel zugegriffen hat und welche Arbeiten daran ausgeführt wurden. Dies soll insgesamt die Anfechtbarkeit der Beweiskraft von erhobenen Beweismitteln, z. B. in einem Gerichtsverfahren, reduzieren.

Cross-Compiler Ein Cross-Compiler ermöglicht es, Kompilate, d. h. Objektdateien oder ausführbare Dateien, für ein anderes als das Entwicklungssystem zu erzeugen.

	Das Entwicklungssystem kann sich sowohl software- als auch hardwareseitig vom Zielsystem unterscheiden. Auf diese Weise ist es möglich, über einen traditionellen Computer (Entwicklungssystem) eine ausführbare Applikation für ein Smartphone (Zielsystem) zu erstellen.
Custom ROM	Ein Custom ROM ist eine angepasste Betriebssystemversion, beispielsweise von Android, die nicht vom ursprünglichen Gerätehersteller stammt. Ziel ist es, mit einer Custom ROM die Funktionalitäten des Smartphones zu erweitern.
Datenakquise	In der Literatur wird der Begriff „Datenakquise“ häufig nicht eindeutig definiert. In dieser Ausarbeitung wird darunter die Erhebung der zu sichernden Daten, deren Untersuchung zur Extraktion potentieller Spuren und die Analyse der extrahierten Spuren in Form einer Korrelation und Bewertung verstanden.
Flasher Box	Eine Flasher Box ist ein spezielles Gerät, welches das Lesen und Überschreiben der Inhalte eines Flash-Speichers erlaubt. Im Smartphone-Umfeld setzten Hersteller die Geräte zur Fehlersuche, Reparatur und Aktualisierung ihrer bereitgestellten Produkte ein.
Ganzzahlüberlauf	Bei einem Ganzzahlüberlauf wird der gegebene Zahlenraum überschritten. Eine solche Überschreitung entsteht durch eine arithmetische Operation, deren Ergebnis einen Zahlenwert annimmt, der zu groß ist, also mehr Stellen aufweist, als im ganzzahligen Datentyp gespeichert werden kann. Die überzähligen Stellen führen letztlich zu fehlerhaften Zahlenwerten.

International Mobile Equipment Identifier

Der International Mobile Equipment Identifier besteht aus einer 15-stelligen Kennnummer, die eine eindeutige Identifikation des Smartphones erlaubt. In der Regel kann die Kennnummer durch Eingabe des Wertes `*#06#` abgerufen oder über ein Typenschild, das sich unter dem Akku befindet, ermittelt werden (vgl. [Nat07] S. 40).

Mobile Device Management

Ein Mobile Device Management-System dient zur zentralisierten Absicherung, Überwachung und Verwaltung von mobilen Endgeräten im Unternehmensumfeld.

Remote Wipe

Ein Remote Wipe hat die Löschung der Daten und die Wiederherstellung der Werkseinstellungen eines Smartphones zur Folge. Der Vorgang kann durch den Gerätebesitzer bei Verlust oder Diebstahl initiiert werden. Voraussetzung zur erfolgreichen Durchführung ist eine bestehende Internetverbindung.

Sandbox

Bei einer Sandbox handelt es sich um eine dedizierte, von anderen Ressourcen isolierte Laufzeitumgebung für Applikationen (vgl. [BP10] S. 27). Der Ansatz wird als Sicherheitsmechanismus genutzt, um Applikationen voneinander zu separieren.

Smudge Attack

Der Begriff „Smudge Attack“ widmet sich der Rekonstruktion eines zu zeichnenden Musters, welches zur Aufhebung des Sperrbildschirms eines Smartphones benötigt wird (vgl. [AGM+10] S. 1). Die Grundlage der Methode bilden angefertigte Fotos von Fettrückständen auf dem Touchscreen des Geräts. Die Fotos werden mit einer Bildbearbeitungssoftware anschließend ausgewertet. Anhand der nicht reflektierenden Fettrückstände kann das zu zeichnende Muster rekonstruiert werden.

Die Methode wurde von Wissenschaftlern der Universität von Pennsylvania tiefgehend erforscht und im Jahr 2010 auf dem vierten USENIX Workshop vorgestellt.

Spur

Als Spuren werden im kriminalistischen Sinne Gegenstände oder Hinweise in einem Ermittlungsverfahren bezeichnet, die eine Theorie über einen Tatbestand oder eine Täterschaft belegen oder widerlegen können (vgl. [Cas11] S. 7). Digitale Spuren basieren hierbei auf Daten, die in IT-Systemen gespeichert oder zwischen diesen übertragen werden (vgl. [Cas11] S. 7). Die ermittelten Spuren dienen schließlich als potentielle Beweismittel.

Anhang C Literaturverzeichnis

[ACC+13] Auty, Mike; Case, Andrew; Cohen, Michael; et al.: *Volatility* : An advanced memory forensics framework. http://code.google.com/p/volatility/, zuletzt besucht am 1. Juli 2013.

[AGM+10] Aviv, Adam; Gibson, Katherine; Mossop, Evan; et al.: *Smudge Attacks on Smartphone Touch Screens*. In: Proceedings of the 4th USENIX conference on offensive technologies, USENIX Association, WOOT 2010, Berkeley, CA, USA, 2010, S. 1-7.

[And13a] Android Developers: *App Install Location* : API Guides. http://developer.android.com/guide/topics/data/install-location.html, zuletzt besucht am 25. April 2013.

[And13b] Android Developers: *Content Provider Basics* : API Guides. http://developer.android.com/guide/topics/providers/content-provider-basics.html, zuletzt besucht am 26. April 2013.

[And13c] Android Developer: *User Security Features* : Filesystem Encryption. http://source.android.com/devices/tech/security/index.html, zuletzt besucht am 25. Juni 2013.

[And13d] Android Developers: *Android Debug Bridge* : Tools. http://developer.android.com/tools/help/adb.html, zuletzt besucht am 30. April 2013.

[And13e] Android Developers: *Device Monitor* : Tools. http://developer.android.com/tools/help/monitor.html, zuletzt besucht am 30. April 2013.

[And13f] Android Developers: *Developer Tools* : Tools. http://developer.android.com/tools/index.html, zuletzt besucht am 30. April 2013.

[And13g] Android Developers: *Logcat* : API Guides. http://developer.android.com/tools/help/logcat.html, zuletzt besucht am 2. Mai 2013.

[And13h] Android Developers: *The AndroidManifest.xml File* : API Guides. http://developer.android.com/guide/topics/manifest/manifest-intro.html, zuletzt besucht am 24. April 2013.

[BBK+13] Becker, Joachim; Berwanger, Jörg; Krumme, Jan-Hendrik; et al.: *Beweismittel* : Gabler Wirtschaftslexikon. http://wirtschaftslexikon.gabler.de/Archiv/371/beweismittel-v10.html, zuletzt besucht am 29. Mai 2013.

[Bec11] Becker, Arno: *Innenansichten – Die Architektur von Android*. In: c't (2011)4, S. 122-127.

[Ber11] Berkel, Jan: *Why can't I get an HPROF dump from certain devices?* : Stack Overflow. http://stackoverflow.com/questions/5640182/why-cant-i-get-an-hprof-dump-from-certain-devices?rq=1&rq=1, erstellt am 12. April 2011, zuletzt besucht am 2. Juni 2013.

[Blo05] Bloomberg Businessweek: *Google Buys Android for Its Mobile Arsenal* : Bloomberg Businessweek. http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal, erstellt am 16. August 2005, zuletzt besucht am 12. April 2013.

[Blo13] Bloomberg Businessweek: *Company Overview of Android Inc.* : Bloomberg Businessweek. http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=23584687, zuletzt besucht am 12. April 2013.

[BP10] Becker, Arno; Pant, Marcus: *Android 2 – Grundlagen und Programmierung.* Heidelberg, Deutschland: dpunkt.verlag, 2010.

[Bun09] Bundesamt für Sicherheit in der Informationstechnik: *M 6.126 Einführung in die Computer-Forensik.* https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m06/m06126.html, erstellt in 2009, zuletzt besucht am 4. Juli 2013.

[Bun11] Bundesamt für Sicherheit in der Informationstechnik: *Leitfaden „IT-Forensik“.* Bonn, Deutschland, 2011. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Internetsicherheit/Leitfaden_IT-Forensik_pdf.pdf?__blob=publicationFile, zuletzt besucht am 23. Mai 2013.

[BZ13] Beisel, Benjamin; Zillner, Florian: *Weitsichtig – Android-Angriffe erkennen und analysieren*. In: iX (2013)2, S. 126-131.

[Cam10] Camera and Imaging Products Association: *Exchangeable image file format for digital still cameras* : Exif Version 2.3. 2010. http://www.cipa.jp/english/hyoujunka/kikaku/pdf/DC-008-2010_E.pdf, erstellt am 26. April 2010, zuletzt besucht am 5. Juli 2013.

[Can13] Cannon, Thomas: *Screen Lock Bypass Pro* : Google Play. https://play.google.com/store/apps/details?id=net.thomascannon.screenlockbypass.pro, zuletzt besucht am 15. Juni 2013.

[Car05] Carrier, Brian: *File System Forensic Analysis.* Crawfordsville, IN, USA: Pearson Education, 2005.

[Car13] Carrier, Brian: *The Sleuth Kit* : Overview. http://www.sleuthkit.org/sleuthkit/, zuletzt besucht am 3. Juli 2013.

[Cas11] Casey, Eoghan: *Digital Evidence and Computer Crime.* Waltham, MA, USA: Elsevier, 2011.

[Cel13a] Cellebrite: *UFED Applications* : Examine, Expose, Evaluate. http://www.cellebrite.com/mobile-forensic-products/ufed-applications.html, zuletzt besucht am 13. Juni 2013.

[Cel13b] Cellebrite: *About Cellebrite* : Mobile Forensic Division. http://www.cellebrite.com/mobile-forensic-company.html, zuletzt besucht am 24. Juni 2013.

[Cel13c] Cellebrite: *Android Forensics* : Physical Extraction and Decoding from Android Devices. http://www.cellebrite.com/forensic-solutions/android-forensics.html, zuletzt besucht am 24. Juni 2013.

[Fai12] Fairbanks, Kevin: *An analysis of Ext4 for digital forensics*. In: Proceedings of the 12th Annual DFRWS conference, DFRWS 2012, New Orleans, LA, USA, 2012, S. 118-130.

[Gar13] Gartner: *Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012* : Gartner Newsroom. http://www.gartner.com/newsroom/id/2335616, erstellt im Februar 2013, zuletzt besucht am 14. April 2013.

[Ges11] Geschonneck, Alexander: *Computer-Forensik: Computerstraftaten erkennen, ermitteln, aufklären.* Heidelberg, Deutschland: dpunkt.verlag, 2011.

[GTA12] Goel, Archit; Tyagi, Anurag; Agarwal, Ankit: *Smartphone Forensic Investigation Process Model*. In: International Journal of Computer Science and Security (2012)5, S. 322-341.

[Gud10] Gudjonsson, Kristinn: *Mastering the Super Timeline With log2timeline*. 2010. http://www.sans.org/reading_room/whitepapers/logging/mastering-super-timeline-log2timeline_33438, zuletzt besucht am 6. Juli 2013.

[Hei10] Heise: *Android 2.3 mit Ext4-Dateisystem*. http://www.heise.de/open/meldung/Android-2-3-mit-Ext4-Dateisystem-1152964.html, erstellt am 14. Dezember 2010, zuletzt besucht am 13. Juni 2013.

[Hei11] Heise: *Smartphones verraten richtige und falsche Geodaten*. http://www.heise.de/mobil/meldung/Smartphones-verraten-richtige-und-falsche-Geodaten-1232287.html, erstellt am 24. April 2011, zuletzt besucht am 5. Juli 2013.

[Hoo11] Hoog, Andrew: *Android Forensics – Investigation, Analysis and Mobile Security for Google Android.* Waltham, MA, USA: Elsevier, 2011.

[Kle11] Klement, Stefan: *Sicherheitsaspekte der Google Android Plattform*. Bremen, Deutschland, 2011. http://www.informatik.uni-bremen.de/~sohr/papers/DiplomarbeitKlement.pdf, zuletzt besucht am 23. Mai 2013.

[KO10] Kovacik, Steven; O'Day, R. : *A Proposed Methodology for Victim Android Phone Analysis by Law Enforcement Investigators*. 2010. https://viaforensics.com/viaforensics-articles/viaforensics-aflogical-tool-android-forensic-investigations.html, erstellt am 1. Februar 2011, zuletzt besucht am 17. Juli 2013.

[Lar13] Larabel, Michael: *Google Working On Android Based On Linux 3.8* : Phoronix. http://www.phoronix.com/scan.php?page=news_item&px=MTMxMzc, erstellt am 27. Februar 2013, zuletzt besucht am 14. April 2013.

[Mas13] Massachusetts Institute of Technology: *SIMILE Widgets* : Free, Open-Source Data Visualization Web Widgets, and More. http://simile-widgets.org/, zuletzt besucht am 6. Juli 2013.

[Mic06] Microsoft: *Beschreibung des FAT32 File Systems* : Hilfe und Support. http://support.microsoft.com/kb/154997, erstellt am 10. August 2006, zuletzt besucht am 4. Juli 2013.

[Nat07] National Institute for Standards and Technology: *Guidelines on Cell Phone Forensics* : Recommendations of the National Institute of Standards and Technology. Gaithersburg, 2007. http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf, zuletzt besucht am 23. Mai 2013.

[Nat13] National Institute of Standards and Technology: *National Vulnerability Database* : Vulnerability Summary for CVE-2013-2596. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-2596, erstellt am 5. April 2013, zuletzt besucht am 25. Juli 2013.

[Ope13] Open Handset Alliance: *Open Handset Alliance*. http://www.openhandsetalliance.com/index.html, zuletzt besucht am 12. April 2013.

[RB12] Roos, Björn; Baier, Harald: *IT-Forensik im Wandel – Die Aufweichung des Paradigmas der Unveränderbarkeit am Beispiel von Smartphones mit dem Windows Phone Betriebssystem*. In: Proceedings of the D.A.CH Security 2012, D.A.CH Security 2012, Konstanz, Schweiz, 2012, S. 301-313.

[Rei10] Reiber, Lee: *Are You Protected?*. http://blog.mobileforensicsinc.com/are-you-protected/, erstellt am 6. Dezember 2010, zuletzt besucht am 24. Juli 2013.

[Roo11] Roos, Björn: *Forensische Untersuchung und Entwicklung eines Frameworks zur Analyse von mobilen Endgeräten mit dem Android OS*. Darmstadt, Deutschland, 2011.

[SG07] Savoldi, Antonio; Gubian, Paolo: *SIM and USIM Filesystem: a Forensics Perspective*. In: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC 2007, Seoul, Korea, 2007, S. 181-187.

[Sta13] Statista: *Anzahl der verfügbaren Apps im Google Play Store (Android Market) von Dezember 2009 bis April 2013*. http://de.statista.com/statistik/daten/studie/74368/umfrage/anzahl-der-verfuegbaren-apps-im-google-play-store/, zuletzt besucht am 19. Juli 2013.

[Syl12] Sylve, Joe: *Android Mind Reading: Memory Acquisition and Analysis with DMD and Volatility* : ShmoCon 2012. 2012. http://www.shmoocon.org/2012/presentations/Android_Mind_Reading.pdf, erstellt am 2. Februar 2012, zuletzt besucht am 3. Juni 2013.

[Syl13] Sylve, Joe: *LiME – Linux Memory Extractor* : Instructions v1.2. 2013. http://lime-forensics.googlecode.com/files/LiME_Documentation_1.1.pdf, erstellt am 19. März 2013, zuletzt besucht am 03. Juni 2013.

[VZC11] Vidas, Timothy; Zhang, Chengye; Christin, Nicolas: *Toward a general collection methodology for Android devices*. In: Proceedings of the 11th Annual DFRWS conference, DFRWS 2011, New Orleans, LA, USA, 2011, S. 14-24.

[WGP+08] Woodhouse, David; Gleixner, Thomas; Pitre, Nicolas; et al.: *UBIFS* : Raw flash vs. FTL devices. http://www.linux-mtd.infradead.org/doc/ubifs.html#L_raw_vs_ftl, erstellt am 26. Oktober 2008, zuletzt besucht am 14. Juni 2013.

[Wir13] Wireshark: *The Wireshark Wiki* : USB capture setup. http://wiki.wireshark.org/CaptureSetup/USB, zuletzt besucht am 4. Juni 2013.

[XDA13] XDA Developers: *[Root][JB 4.2] Root your Nexus 7 without unlocking bootloader*. http://forum.xda-developers.com/showthread.php?t=2233852, erstellt am 14. April 2013, zuletzt besucht am 25. Juli 2013.

[XDI09] XDIN: *The Android boot process from power on* : Android Blog. http://xdinandroid.com/2009/06/the-android-boot-process-from-power-on.html, erstellt am 11. Juni 2009, zuletzt besucht am 17. April 2013.