FernUniversität in Hagen
Fakultät für Mathematik und Informatik

Abschlussarbeit
im Studiengang Informatik Bachelor of Science

Analyse von Daten einer Smartwatch im Zusammenhang mit Gewaltverbrechen

von
Jana Vera Villforth
Matrikelnummer: 9539310

Datum der Abgabe: 02.12.2024

Erstgutachterin und Betreuerin: Dr. Carina Heßeling
Zweitgutachter: Dr. Marius Rosenbaum

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Seit 2020 steigt das Volumen der jährlich generierten oder replizierten digitalen Datenmenge signifikant an [1]. Der Grund dafür ist die zunehmende Anzahl an Geräten, die Daten erzeugen und speichern können. Beispielsweise ist es Stand 2024 erst seit knapp zehn Jahren möglich eine Uhr am Handgelenk zu tragen, die sich mit dem Mobiltelefon verbinden lässt und Vitaldaten misst und aufzeichnet. Eine Uhr mit diesen intelligenten Funktionen nennt man Smartwatch. Nutzenden ist es durch das Tragen einer Smartwatch unter anderem möglich ihr Training nachzuverfolgen oder unabhängig vom Mobiltelefon zu telefonieren. Die so erzeugten Daten sind jedoch nicht nur für die Nutzenden von Interesse, sondern auch wichtig für die Nachverfolgung von Spuren in Kriminalfällen. Die Aufnahme von Beweismitteln in Strafverfahren hat sich durch die technologische Entwicklung maßgeblich verändert.
Stellvertretend wird die Bedeutung digitaler Spuren an einem Fall des Landgericht Karlsruhe deutlich gemacht. Im Rahmen der Beweismittelaufnahme zur Feststellung der Täterschaft eines Angeklagten ging es überwiegend um die Analyse und die Gewinnung von Erkenntnissen aus digitalen Spuren. Der Anteil anderer Indizien war minimal und hätte zur Überführung des Täters nicht ausgereicht. Das Landgericht hat als Sachverständige eine Forensikerin des Landeskriminalamt (LKA) Baden-Württemberg gehört, die Daten aus der Smartwatch des Opfers und des Smartphones des Täters extrahiert und ausgewertet hat [2]. Die von ihr geleisteten Untersuchungen der intelligenten Geräte haben den Angeklagten überführt, der seine Täterschaft in Abrede gestellt hatte und gleichzeitig den Bedarf für zusätzliche Forschung aufgezeigt.
In der vorliegenden Arbeit sollen vorbereitende Maßnahmen getroffen werden, die im Anschluss vom LKA Baden-Württemberg zur Messung von Daten an verstorbenen Personen genutzt werden. Diese Daten sollen Ermittelnden Rückschlüsse auf mögliche Tathergänge bei Gewaltverbrechen ermöglichen. Bevor solche Messungen durchgeführt werden können, muss

eine Topologie der Dateisysteme verwendeter Messgeräte erstellt werden. Außerdem soll die Validität gesammelter Daten geprüft werden. Ausblickend soll es durch die geleisteten Untersuchungen möglich gemacht werden Messwerte von lebenden und verstorbenen Personen zu unterscheiden und so gerichtsmedizinische Vorgänge durch digitale Spuren unterstützen zu können.
Hierbei ergeben sich die im Folgenden aufgezeigten Probleme.

1.1. Problem

Eine Schwierigkeit der digitalen Forensik ist die Menge der vorliegenden Daten, die von Nutzenden über Jahre auf ihrem Endgerät gesammelt werden. Durch Speicherkapazitäten von bis zu 1TB und Cloud Dienste wächst diese Menge stetig an. Für die Ermittlungsarbeit in Strafverfahren ist es demnach essentiell beweisträchtige Daten effizient identifizieren zu können.
Ein zentrales Problem forensischer Untersuchungen ist der Konflikt zwischen dem Schutz firmeninterner Software und der gründlichen Untersuchung von Geräten. Um Daten analysieren zu können, ist es zunächst wichtig zu wissen in welcher Form die Daten verarbeitet und wo diese gespeichert werden.
Zusätzlich besteht bei der Beweisaufnahme in Strafverfahren die Schwierigkeit, dass nicht immer sämtliche Datenquellen zugreifbar sind. Auch wenn eine Smartwatch mit einem Mobiltelefon gekoppelt sein kann, bedeutet das nicht, dass beide Datenträger für eine Analyse zur Verfügung stehen.
Neben der aktuellen Schnelllebigkeit der Verbraucherelektronik stellt die Feststellung der Validität vorliegender Daten eine Herausforderung dar. Ein Beispiel hierfür ist die Aufzeichnung von Daten an Gegenständen. Es ist möglich eine Smartwatch an einem Gegenstand wie einer Banane anzubringen, die dann beispielsweise einen Puls auf dem Display der Smartwatch anzeigt [3]. Es ist unklar, welche Daten zeigen, ob sich eine Smartwatch tatsächlich an einem Handgelenk befunden hat oder nicht. Recherchemaßnahmen und Vergleichstests kosten viel Zeit, ebenso wie die zuverlässige Deutung erhobener Daten. Die Forschung im Bereich der digitalen Forensik ist aufgrund der noch jungen Entwicklung von Smartwatches bislang nur wenig fortgeschritten.
Aus diesen Problemen ergibt sich die im Folgenden ausgeführte Zielstellung

der Arbeit.

1.2. Zielstellung

Die zentralen Themen der Arbeit sind die Aufnahme der Topologie der Dateisysteme verwendeter Geräte, sowie die Prüfung der Validität aufgezeichneter Daten. Für diese Arbeit von Relevanz sind hierbei Ordner und Dateien, die im Zusamenhang mit Vitaldaten stehen und Ermittelnden ermöglichen Rückschlüsse auf wahrscheinliche Tathergänge zu ziehen.
Grundlage für die Ermittlung beider Ziele ist hierbei die protokollierte Aufnahme von Messdaten. Anschließend werden die Dateisysteme mittels verschiedener Methoden gesichert und einer Analyse unterzogen. Diese soll zum Ergebnis haben, welche Datenelemente gefunden werden können und wo diese abgelegt werden. Es ist möglich, dass Datenträger zum Zeitpunkt der Aufnahme von Beweismitteln in Strafverfahren nicht zugreifbar sind. Es wird geprüft, ob auf allen untersuchten Geräten die gleiche Datenmenge vorhanden ist oder ob Daten außerhalb der Schnittmenge beider Objekte existieren.
Eine Aussage über die Validität der Daten soll durch einen Abgleich der extrahierten Daten mit den Messprotokollen getroffen werden.

1.3. Struktur der Thesis

Die vorliegende Arbeit ist in fünf Kapitel gegliedert.
Das zweite Kapitel gibt einen Einblick in den aktuellen Stand der Forschung und vermittelt Grundlagen zu den behandelten Geräten sowie der verwendeten Software. Kapitel drei geht auf die verwendete Methodik und die in der Arbeit untersuchten Geräte ein. In Kapitel vier werden die Ergebnisse dargestellt und diskutiert. Das letzte Kapitel fasst die Arbeit zusammen und gibt anschließend einen Ausblick auf weitere darauf aufbauende Forschung.

2. Grundlagen

2.1. Internet der Dinge

Laut Uckelmann et al. [4] gibt es für den Begriff Internet of Things (Internet der Dinge) (IoT) keine einheitliche Definition. Er steht als Sammelbegriff über physischen Dingen, die um Funktionen eines Computers erweitert wurden, sich mit dem Internet oder untereinander verknüpfen lassen und in der Lage sind, Daten auszutauschen. Geräte wie beispielsweise Heizungen, Lampen und Kühlschränke lassen sich steuern oder überwachen und senden, empfangen oder speichern Daten. Eine direkte Verbindung entsteht durch einen Anschluss an ein Wireless Local Area Network (WLAN) bzw. Ethernet oder ein Mobilfunknetz. Eine indirekte Verbindung zum Internet wird durch die Bluetooth Verknüpfung zu einem anderen Gerät hergestellt. Eine Bluetooth Verbindung wird für kurze Distanzen gewählt und ist im Vergleich zu einer Verbindung von Geräten über WLAN weniger stabil. Während WLAN Verbindungen auch für den Austausch großer Datenmengen geeignet sind, hat Bluetooth eine vergleichsweise geringe Datenübertragungsrate. Während per Bluetooth maximal 24 Megabits pro Sekunde übertragen werden können, sind es bei WLAN durchschnittlich bis zu 200 Megabits pro Sekunde [5]. In der Praxis kommen vor allem für tragbare Geräte des IoT hybride Verbindungen zum Einsatz. Innerhalb der Reichweite eines bekannten WLANs befinden, sind IoT-Geräte mit diesem Netzwerk verbunden. Werden sie außerhalb dieser Reichweite gebracht, erfolgt der Wechsel zum Mobilfunknetz.
Ein Beispiel für ein tragbares Gerät aus dem IoT Bereich ist das Smartphone. Hierbei handelt es sich um ein tragbares elektronisches Gerät, das die Funktionen eines Mobiltelefons mit Funktionen eines Computers kombiniert. Durch die Möglichkeit des Verbindungsaufbaus zum Mobilfunknetz oder einem WLAN können Nutzende telefonieren und auf das Internet sowie verschiedene Anwendungen zugreifen [6]. Der Zugriff auf das Internet ermöglicht, dass Smartphones auf sogenannte Cloud-Dienste zugreifen

können. Eine Cloud bezeichnet ein Netzwerk von Servern, das Nutzenden Anwendungen, Daten oder Speicherplatz bereitstellt. Der Vorteil hierbei ist, dass dafür notwendige Kapazitäten nicht lokal von einem Endgerät bereitgestellt werden müssen. Häufig werden in der Cloud Sicherungsdateien eines Smartphones hinterlegt, um Geräte im Fall eines Verlusts oder Neuerwerbs mit minimalem Aufwand wiederherstellen zu können. Ein weiteres Gerät des IoT ist die Smartwatch. Sie ist ein mobiles Endgerät, das am Handgelenk getragen wird und sowohl die Funktionen einer Uhr als auch weitere intelligente Funktionen besitzen kann. Je nach Hersteller und Preisklasse variieren die verfügbaren Funktionen stark. Diese werden durch den Einsatz von Sensoren realisiert. Sensoren verwenden spezifische physikalische Effekte, um die Messgröße in ein elektrisches Signal umzuwandeln [7]. Diese Signale werden anschließend so verarbeitet, dass diese Daten auf dem Bildschirm der Smartwatch ausgelesen werden können. Eine nähere Beschreibung der durch die Sensoren ermöglichten Funktionen sind in Kapitel 3.2 und Kapitel 4 für die in der Arbeit verwendete Smartwatch dargestellt.

Smartwatches können, wie in Abbildung 2.1 dargestellt, in verschiedenen Modi verwendet werden.

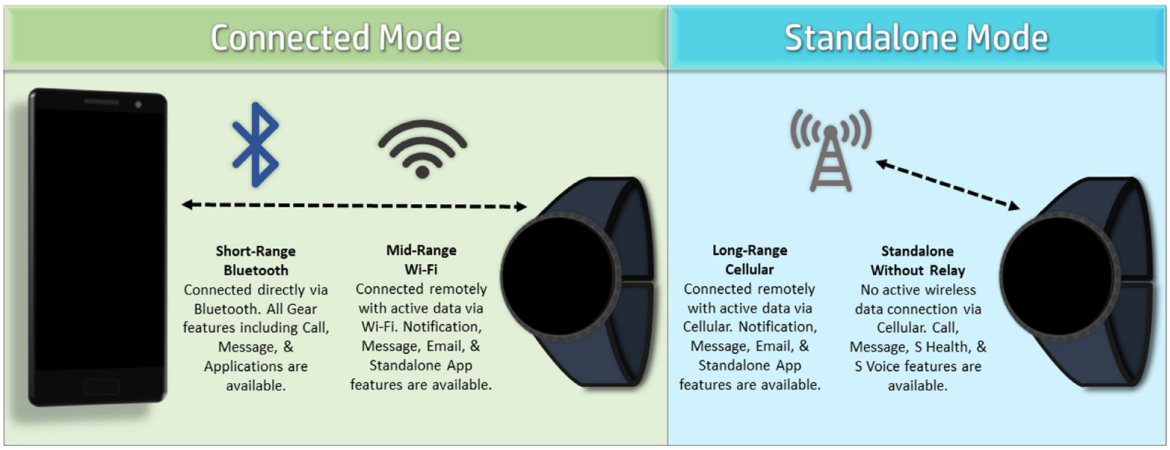

Abbildung 2.1.: Darstellung von verbundenem und eigenständigem Modus nach Odom et al. [8].

Am häufigsten wird eine Smartwatch im verbundenen Modus, in 2.1 links dargestellt, genutzt. Hierbei wird via Bluetooth oder WLAN eine Verbindung zwischen Smartwatch und Smartphone hergestellt. Auch wenn die Reichweite zwischen den beiden Geräten unterbrochen wird, können

netzunabhängige Dienste der Smartwatch weiter genutzt werden. Netzabhängige Dienste sind alle Dienste, die die Anbindung an ein mobiles oder lokales Netz erfordern. Beispiele hierfür sind das Telefonieren oder das Senden und Empfangen von Nachrichten über Short Message Service (SMS) oder E-Mail Dienste. Es gibt Smartwatches, die die Verbindung zu einer Subscriber Identity Module (SIM)- oder embedded SIM-Karte erlauben. In ein Gerät eingesetzt, dient eine SIM-Karte der Authentifikation von Nutzenden und erlaubt die Verbindung zum Mobilfunknetz durch einen Mobilfunkanbieter. In einer Smartwatch erfüllt sie denselben Zweck, was den Einsatz der Smartwatch unabhängig vom Smartphone erlaubt. Diese Form der Nutzung ist in Abbildung 2.1 rechts dargestellt und wird im Folgenden als eigenständiger Modus bezeichnet.
Ein weiterer wichtiger Modus, der die Funktionen der intelligenten Geräte beeinflusst, ist der Energiesparmodus. Schalten Nutzende diesen Modus ein, so werden Dienste, die die Laufzeit der Batterie maßgeblich beeinflussen, reduziert oder deaktiviert. So wird der Energieverbrauch minimiert und die Nutzungsdauer verlängert [9]. Darüber hinaus ist es möglich nur das Ziffernblatt zu aktivieren und alle anderen Dienste vollständig zu deaktivieren. Dann befindet sich die Smartwatch im sogenannten „Nur Uhrfunktion“ Modus.
Um IoT-Geräte betreiben zu können, ist, wie bei einem Computer, ein Betriebssystem notwendig. Während die gängigsten Betriebssysteme für stationäre Computer Varianten von Windows, macOS und Linux sind [10], sind bei mobilen Endgeräten die Betriebssysteme Android und iOS am häufigsten vertreten [11]. Während iOS ein proprietäres Betriebssystem der Firma Apple® ist, ist Android ein Open Source Projekt, das nach seiner Entwicklung von Google LLC gekauft wurde und nach der Umfirmierung Teil der Alphabet Incorporated (Inc.) ist. Der zu Android gehörige Quellcode ist öffentlich einsehbar und wird von einem großen Netzwerk von Nutzenden stets weiterentwickelt. Für Smartwatches entwickelte Google das Betriebssystem Wear OS, Apple® führt sein Betriebssystem für Smartwatches unter dem Namen watchOS.
Die Möglichkeiten der Untersuchung von Android Geräten ist aufgrund der im Internet frei verfügbaren Android Erweiterungen deutlich einfacher als bei Geräten der Firma Apple® . Eine Methode, um die interne Struktur eines Geräts genauer zu analysieren, besteht darin, es zu rooten. Beim Rooten handelt es sich um einen Prozess, bei dem Nutzende

erweiterte Berechtigungen, sogenannte Root-Rechte, erhalten können. So ist es möglich, vollumfänglich auf das Betriebssystem zuzugreifen und ein Gerät in größerem Maße anzupassen, als es bei ungerooteten Geräten der Fall ist. Viele Systemordner sind Nutzenden ungerooteter Geräte nicht zugänglich. Ein weiteres Hindernis, das durch das Rooten von Geräten umgangen wird, sind Sicherheitsvorkehrungen wie das sichere Betriebssystem Trusty, das von Android entwickelt wurde [12]. Hierbei handelt es sich um die Implementierung isolierter Umgebungen zum Schutz sensibler Informationen. Trusty dient den Herstellern als Grundlage, die für ihre gerätespezifische Software und Anwendungen angepasst werden kann. Ein Rootvorgang ist nicht nur auf Smartphones beschränkt, sondern kann auch auf Smartwatches angewendet werden. Die Anforderungen an das Rooten mobiler Endgeräte sind in Kapitel 3.3 aufgeführt. Da es Smartphones und das zugehörige Betriebssystem Android schon länger gibt (2007) als Smartwatches und das dazugehörige Betriebssystem WearOS (2014) [13], ist die Menge an frei verfügbarer Software für Smartphones deutlich größer. Das Rooten eines Smartphones erfordert die Installation einer neuen Firmware. Diese besteht aus dem Betriebssystem, sowie spezifischen Treibern und Software Komponenten, die die Hardware des Geräts ansteuern. Die Software Komponenten beinhalten unter Anderem den Bootloader, der für das Laden des Betriebssystems zuständig ist und die Integrität der Firmware prüft. Der Vorgang, bei dem die neue Firmware auf ein Gerät geladen wird, wird als flashen bezeichnet [14].

2.2. Digitale Forensik

Die Daten, die durch elektronische Geräte aufgezeichnet werden, haben die Forensik grundlegend verändert und sie um die digitale Forensik erweitert. Die Forensik umfasst laut Labudde und Spranger [15] sämtliche Tätigkeitsbereiche, die strafrechtlich und zivilrechtlich relevante Handlungen identifizieren, ausschließen, analysieren und rekonstruieren. Ein Verstoß gegen die Rechtsordnung wird dabei als Straftat bezeichnet. Vor dem Aufkommen des Internets war eine Straftat eindeutig einem Tatort zuzuordnen. Der Begriff „Tatort“ umfasst dabei alle Orte, die im Zusammenhang mit der Straftat stehen. Alle diese Orte sind potentielle Fundorte von Spuren,

durch die eine Tat rekonstruiert werden kann und Ermittelnden entscheidende Beweise liefern können, um Täter zu überführen. Durch elektronische Geräte und das Internet wurde diese Quelle von Spuren vom physischen in den virtuellen Raum verlagert. Digitale Spuren sind Spuren, die auf Daten basieren, welche in Computersystemen gespeichert oder übertragen worden sind [16]. Daten können hierbei auf mehreren Wegen akquiriert werden:

- Bei einer Live Sicherung werden Daten vom laufenden System extrahiert, bevor es abgeschaltet oder verändert wird. Das erlaubt die Sicherung flüchtiger Daten wie beispielsweise den Inhalt des Arbeitsspeichers oder laufender Prozesse.
- Bei einer Post Mortem Akquise werden Daten von Systemen oder Geräten ausgelesen, die nicht mehr in Betrieb sind.
- Bei der Carving Methode werden Dateien oder Datenfragmente aus einem Datenträger wiederhergestellt. Dieser muss dafür nicht intakt sein. Häufig wird diese Methode auch bei formatierten Datenträgern angewendet, um gelöschte Inhalte wiederherzustellen [15].

Zudem gibt es verschiedene Grade der Sicherung von Daten:

- Bei der logischen Datensicherung werden Daten dateiweise kopiert oder durch eine einfache Sichtung der Geräte erfasst.
- Eine physikalische Sicherung ist eine bitweise Kopie des Speichers, die neben der Post Mortem Akquise auch die Carving Methode erlaubt [6].

Ein besonderes Augenmerk der Forensik liegt dabei auf der gerichtsfesten Akquisition von Beweisen. Eine Manipulation der Daten muss hierbei nachweislich ausgeschlossen sein und Daten müssen integer sein, sich also in ihrem ursprünglichen Zustand befinden [16].
Da die Datenverwaltung der untersuchten Geräte in Datenbanken stattfindet, ist es für die Analyse wichtig, einige Begriffe kurz zu definieren. Die Idee relationaler Datenbanken besteht laut Kofler [17] darin, sämtliche Daten in Tabellen zu organisieren. Eine Relation ist dabei eine Tabelle innerhalb einer Datenbank. Sie besteht aus Zeilen, die als Tupel bezeichnet werden, und Spalten, die die Attribute der Tupel beschreiben. Attribute haben Eigenschaften, die das Attribut weiter konkretisieren.

2.3. Stand der Forschung

Seit der Veröffentlichung der ersten Smartwatch fanden Entwicklungen in mehreren Bereichen statt. Beispiele hierfür sind die verbesserte Energieeffizienz, um die Größe der Batterie bei gesteigerter Leistung zu reduzieren. Durch die Optimierung von Sensoren konnte die Genauigkeit von Messergebnissen verbessert werden. Neue Sensoren wurden in die Smartwatch integriert, um zusätzliche Funktionen bereitzustellen. Einen großen Mehrwert bietet eine Smartwatch für den Fitness- und Gesundheitssektor. Forschung im Bereich Bewegungserkennung wurde beispielsweise zur automatischen Trainingserkennung genutzt. Zhuang et al. [18] haben ein Modell entwickelt, das davon abweicht immer dieselben Zeitintervalle zu betrachten und stattdessen zwischen periodisch wiederkehrenden Bewegungen und unregelmäßigen Bewegungen unterscheidet. So ist es möglich, dass die Trainingserkennung einer Smartwatch Bewegungsabläufe spezifischer zuordnen kann. Vilarinho et al. [19] haben ein System zur automatischen Erkennung von Stürzen entwickelt, um Senioren oder kleinen Kindern im Notfall schnell Hilfe zukommen zu lassen. Hierzu wurden Daten der in der Smartwatch verbauten Beschleunigungssensoren ausgelesen. Um die Daten auslesen zu können, ist es notwendig, ein Programm auf den zu untersuchenden Geräten zu installieren. Das Programm fängt die Daten des Sensors ab und speichert diese in einem verwertbaren Dateiformat. Eine Möglichkeit die Daten nachträglich und ohne ein solches Programm auszulesen wird nicht beschrieben. Viel Forschung gibt es außerdem im Bereich der Früherkennung von Herzerkrankungen [20], bei der Messungen zur Herzfrequenz auf Muster untersucht werden, die Hinweise auf mögliche Probleme wie Arrhythmien oder ein erhöhtes Risiko für Herzinfarkte liefern können.
Forschungsarbeit auf dem Gebiet der digitalen Forensik bezieht sich häufig allgemein auf Forensik von IoT-Geräten und nicht direkt auf Smartwatches wie in der Arbeit von Alharbi et al. [21], in der intelligente Geräte einer Wohnung untersucht werden. Alabdulsalam et al. [22] gehen zudem auf die Herausforderungen ein, die mit der Untersuchung von IoT-Geräten verbunden sind. Janarthanan et al. [23] beschreiben Einschränkungen, die sich durch Anforderungen an Privatsphäre, Authentifizierung, den heterogenen Markt an Geräten und Nutzungsbedingungen ergeben und vergleichen aktuelle Literatur der digitalen Forensik. Wenige Arbeiten haben sich mit

der Erfassung von Daten über eine Smartwatch beschäftigt. In [24] hat Adebayo ein Smartphone gerootet und über verschiedene Tools Daten zu Kalenderevents, Kontakten und verbundenen Geräten extrahiert. Da die Datenbank zur Anwendung Samsung™ Health verschlüsselt ist, wurden hierzu keine weiteren Analysen vorgenommen. Odom et al. [8] haben das Augenmerk ihrer Arbeit auf die Erfassung allgemeiner Daten aus Kontakten, sozialen Netzwerken und Kalenderevents nach der Methode des National Institute of Standards and Technology (NIST) [25] gelegt. Da in der vorliegenden Arbeit Rückschlüsse auf Gewaltverbrechen gezogen werden sollen, fehlt bisher eine Erhebung und Analyse von Vitaldaten sowie eine Bestimmung der Speicherorte. Keine der bisher durchgeführten Forschungen kann als Grundlage für zukünftige Messungen an verstorbenen Personen dienen.
Aufgrund der frühen Entwicklungsphase, ist die Forschung auf dem Gebiet der IoT-Geräte wenig fortgeschritten. Dokumente zur Erfassung von Daten auf IoT-Geräten beziehen sich auf die forensische Untersuchung der Geräte wie zum Beispiel die ISO/IEC 27037:2012 [26] oder auf eine sichere Konfiguration des Endgeräts nach RFC 8572 [27]. Nur das auch von Odom et al. [8] verwendete Dokument 800-202 des NIST gibt Richtlinien zur Erfassung von Daten vor [25]. Da in der vorliegenden Arbeit Gesundheitsdaten, die von der Smartwatch gemessen wurden, von Interesse sind, wird Tabelle 20 aus Anhang B der Richtlinien für die Erstellung von Messprotokollen verwendet. Siehe hierzu Kapitel 3.1.
Darüber hinaus sollen in der vorliegenden Arbeit Log-Dateien untersucht werden. In [24] konnten durch Adebayo aus den Log-Dateien nur Erkenntnisse zu verbundenen Geräten, Kalendereinträgen und Suchanfragen gewonnen werden. Weitere Studien zur Analyse von Log-Dateien verfolgen verschiedene Ziele. Eine der Zielsetzungen besteht darin, Rückschlüsse auf die Handynutzung während des Fahrens zu ziehen. Bortnik et al. [28] haben dabei Ereignisse untersucht, die auf einen abgelenkten Fahrer hinweisen. Timko et al. [29] analysierten Bluetooth-Daten zwischen verschiedenen Smartphones und Smartwatches hinsichtlich möglicher Sicherheitslücken. Eine spezifische Analyse auf Android- oder insbesondere Samsung™-Geräten liegt jedoch nicht vor.

3. Methoden

3.1. Vorgehen

Im ersten Schritt der Projektplanung wurden Messpläne zur Erfassung von Vitaldaten entwickelt. Um die Forschungsfragen erfolgreich zu beantworten, mussten diese Pläne die im Folgenden genannten Kriterien erfüllen.
Da die Messungen nach Abschluss der Arbeit an verstorbenen Personen wiederholt werden, ist es sinnvoll sich unter anderem an den Abläufen des Universitätsklinikums Heidelberg zu orientieren. Dort werden verstorbene Personen in Aufzügen transportiert, für Untersuchungen umgelagert oder wechseln auf Autopsietischen gelagert ihren Standort. Nachgestellt werden diese Prozesse durch Aufzugfahren, das Umlagern einer liegenden Person sowie das Fahren in einem Rollstuhl. Gleichzeitig sollen die Messungen Referenzwerte für mögliche Tathergänge liefern. Im Eingangs vorgestellten Delikt war die Überwindung von Etagen ein wichtiger Hinweis für ermittelnde Kräfte der Polizei [2]. Stockwerke werden in den nachfolgend beschriebenen Versuchen mittels eines Aufzugs und durch Treppensteigen überwunden. Dem Handbuch der Smartwatch [9] und aktuellen Studien, wie der von Zhuang und Xue [18], konnte entnommen werden, dass die Messgenauigkeit durch die Position von Smartwatch und Smartphone beeinflusst werden kann. Aus diesem Grund wurde die Position der Geräte bei den Messungen, wie in Abbildung 3.1 dargestellt, einheitlich festgelegt. Hierbei wird zwischen dem links dargestellten verbundenen Modus und dem eigenständigen Modus, in der Abbildung rechts, unterschieden. In beiden Fällen wird die Smartwatch wie im Benutzerhandbuch empfohlen angebracht. Es wird empfohlen, dass Rechtshänder die Smartwatch am linken Handgelenk und Linkshänder sie am rechten Handgelenk tragen. Außerdem ist zu beachten, dass die Smartwatch am Handgelenk auf trockener Haut anliegt, ohne dabei zu fest zu sitzen [9]. Für den verbundenen Modus wird das Smartphone in der Nähe der Smartwatch am Körper getragen. Für Messungen im eigenständigen Modus wird die Smartwatch so weit

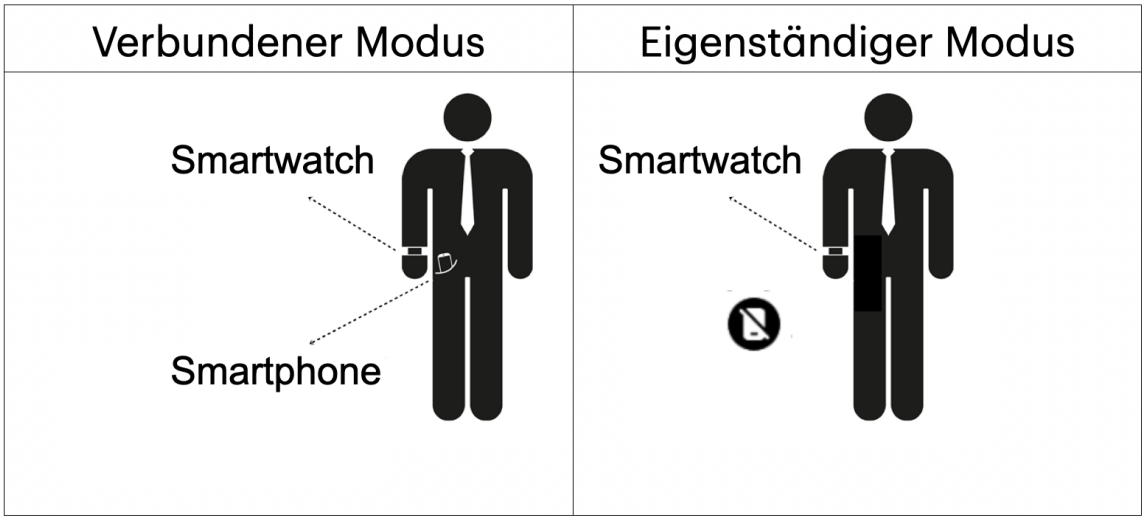

Abbildung 3.1.: Darstellung Aufnahme von Messdaten.

vom Smartphone entfernt, bis das Zeichen auf dem Display erscheint, das die Trennung von Smartwatch und Smartphone anzeigt. In Abbildung 3.1 ist dies im Feld des eigenständigen Modus abgebildet.
Um die gemessenen Daten validieren und anschließend interpretieren zu können, müssen die Messungen protokolliert werden. Dabei wird das von NIST [25] standardisierte Verfahren angewendet, um Daten auf IoT-Geräten zu generieren. In einem ersten Schritt werden Marke und Modell des Endgeräts aufgenommen. Zur eindeutigen Identifikation des Geräts werden auch International Mobile Equipment Identity (IMEI), Mobile Equipment Identifier (MEID) und Electronic Serial Number (ESN) dokumentiert, siehe hierzu die Tabellen A.1, A.3 und A.5 im Anhang. Laut NIST sind ebenfalls Mobile Station Integrated Services Digital Network (MSISDN) oder Mobile Identification Number (MIN) zu dokumentieren. Da in dieser Arbeit keine SIM-Karte verwendet wird, entfällt dies. Um Unterschiede durch Software Updates in weiterführenden Versuchen nachvollziehen zu können, werden nach dem Vorgehen von Alabdulsalam et al. [22] Daten zur aktuellen Software der Geräte dokumentiert. Siehe hierzu die Tabellen A.2, A.4 und A.6 im Anhang. Im nächsten Schritt werden die Ziele aus den Messplänen, wie in Anhang A definiert, in Tabelle 20 des Leitfadens zur Generierung von Daten auf mobilen Testgeräten [25] übertragen. Die Tabelle verfügt über drei Spalten mit den Überschriften Datenobjekte, Eigenschaften und Werte. In der ersten Spalte wird eingetragen, auf welche Anwendung sich das Objekt bezieht, das generiert werden soll. In der

Spalte Eigenschaften wird allgemein beschrieben, welche Art von Daten generiert werden soll. In der Spalte Wert sind die konkreten Eigenschaften der Daten aufgeführt, die generiert werden sollen. Die zum Zwecke der vorliegenden Arbeit ausgefüllten Tabellen sind in den Kapiteln 4.3.1 und 4.3.2 zu finden.
Die Messungen werden mit zwei Smartphones und einer Smartwatch, wie im Kapitel 3.2.1 beschrieben, durchgeführt. Das Samsung™ Galaxy S22 (S22) wird mit Mitteln des LKA Baden-Württemberg gesichert, während Messdaten aus dem Samsung™ Galaxy S8 (S8) mit öffentlich verfügbaren und in Kapitel 3.2.2 erläuterten Mitteln extrahiert werden. Um zusätzliche Hindernisse zu vermeiden, wird bei der Ersteinrichtung der Smartphones darauf verzichtet, Sicherheitsvorkehrungen wie eine persönliche Geheimzahl oder ein Entsperrmuster festzulegen.
Anschließend wird das S8 wie in Kapitel 3.2.3 beschrieben gerootet, um uneingeschränkten Zugriff auf das Dateisystem zu erlangen.
Nachfolgend werden Smartphone und Smartwatch nach der Anleitung des Handbuchs verbunden und Messungen anhand der zuvor erstellten Messpläne durchgeführt. Hierbei erfolgten je zwei Durchgänge. Beim ersten Durchgang werden die Messungen im verbundenen Modus durchgeführt, im zweiten Durchgang wird die Verbindung zwischen Smartphone und Smartwatch getrennt, sodass diese sich im eingenständigen Modus befindet.
Anschließend wurden die Dateisysteme der Smartphones untersucht. Grundlage hierfür bilden die Sicherungsdateien des LKA Baden-Württemberg, die aus dem S22 entnommen werden konnten, sowie die Möglichkeiten, die das Programm Android Debug Bridge (adb) in Verbindung mit dem gerooteten S8 bieten. Die Ergebnisse dieser Untersuchungen werden in Kapitel 4.2 präsentiert. Die Ergebnisse des Abgleichs von Daten und Protokollen sind in Kapitel 4.3 zu finden. Abschließend wird der Datenverkehr zwischen Smartwatch und Smartphone ausgelesen. Hierfür werden die Log-Dateien des Bluetooth-Datenverkehrs mitgeschnitten, extrahiert und untersucht. Die Ergebnisse hiervon werden in Kapitel 4.4 dargestellt.

3.2. Material

3.2.1. Geräte

Für die nachfolgend beschriebenen Versuche werden mehrere Geräte verwendet. Es liegen ein S8 mit der Modellnummer SM-G950F, ein S22, mit der Modellnummer SM-S901B/DS, sowie eine Smartwatch des Modells Samsung™ Galaxy Watch5 LTE mit der Modellnummer SM-R915F vor. Das S8 verfügt über die Android-Version 9 (8/2018) , das S22 verfügt über die Android-Version 14 (10/2023) und die Galaxy Watch5 hat das Betriebssystem Google Wear OS installiert. Eine Übersicht zu allen Versionen von Android und deren Veröffentlichungen ist auf der Internetseite von Android Developers [30] zu finden unter dem Pfad /about/versions.
Laut Statista [31] ist Android das am häufigsten genutzte Betriebssystem auf Smartphones. Die Marke Samsung™ ist unter allen genutzten Geräten, Stand Oktober 2023, am häufigsten vertreten [32]. Die Geräte werden stellvertretend für die Galaxy Reihe der Marke Samsung™ untersucht, wobei das S8 (04/2017) ausgewählt wurde, um im Falle irreversibler Schäden am Gerät, Ressourcen zu sparen. Das S22 (02/2022) vertritt die neueren Modelle der Reihe.
Die Smartwatch ist kompatibel mit mobilen Endgeräten die mindestens über die Android Version 8.0 verfügen. Ihr Arbeitsspeicher ist 1,5GB groß, der interne Gerätespeicher hat 16GB, wovon ca. 7,5GB frei verfügbar sind. Tabelle 3.1 enthält eine Aufstellung der in der Smartwatch enthaltenen Sensoren und deren Funktionen. Wenn die Ergebnisse eines Sensors logisch in einer Anwendung ausgelesen werden können, ist dies in der jeweiligen Spalte durch ein x gekennzeichnet. Die Abkürzungen SH und SHM stehen hierbei für Samsung™ Health und Samsung™ Health Monitor. Die Bioelektrische Impedanzanalyse (BIA) bezeichnet die Messung der Körperzusammensetzung anhand des elektrischen Widerstands, den verschiedene Gewebearten wie Muskeln, Fett und Wasser dem Strom entgegensetzen. So kann die Verteilung von Körperfett, Muskelmasse und Wasseranteilen bestimmt werden [33].

Tabelle 3.1.: Sensoren Samsung™ Galaxy Watch5 und Funktionen, vgl. [9].

Sensor	Funktion	SH	SHM
Barometer	Messungen des Luftdrucks werden genutzt, um die Höhe über dem Meeresspigel zu ermitteln	x	
Beschleunigungssensor	Verfolgen von Bewegungen und Aktivitäten (z.B. Schrittzähler, Messen von Entfernungen,..)	x	
Blutdrucksensor	Misst Blutdruck		x
Elektrokardiogramm (EKG)	Empfängt EKG Signale		x
Kompass	Geomagnetischer Sensor ermöglicht die Bestimmung der Himmelsrichtungen		
Lagesensor (Gyrosensor)	Feststellen der Winkelgeschwindigkeit, Einheit: Grad/ Sekunde [34], Verfolgt Orientierung der Uhr im Raum, bestimmt anhand dieser Daten Häufigkeit, Dauer, Intensität und Muster von Bewegungen	(x)	
Photoplethysmographie Sensor	Lichtsensor (Photodiode) und LED, bestimmt Blutvolumen und Sauerstoffsättigung im Blut	x	
Samsung™ BioActive Sensor	optische Herzfrequenzmessung, elektrische Herzsignalverfolgung, BIA		x
Umgebungslicht-Sensor	Automatisches Anpassen der Bildschirmhelligkeit		
Temperatursensor	Misst Hauttemperatur	x	

Das in der Spalte SH geklammerte x bedeutet, dass die verarbeiteten und nicht die konkreten Werte aus x-, y- und z-Achse des Lagesensors ausgelesen werden können. Die so gewonnenen Werte werden zur automatischen Trainingserkennung oder zur Schrittzählung genutzt. Das Verfahren zur Bestimmung eines Puls wird Photoplethysmographie genannt, der zugehörige Sensor heißt Photoplethysmographie Sensor. Um einen Puls zu bestimmen, werden zwei LEDs auf der Rückseite der Smartwatch eingesetzt. Durch die rote LED wird bestimmt, ob die Smartwatch am Handgelenk getragen wird. Die grüne LED blinkt und gibt dabei Licht ab. Das Blutvolumen in den

Gefäßen ändert sich periodisch mit jedem Herzschlagen. Die grüne LED blinkt mehrmals pro Minute. Je nach Blutvolumen in den Gefäßen, wird unterschiedliche Mengen Licht reflektiert. Die Werte, die die Photodiode misst, dienen als Grundlage für die Berechnung der Herzfrequenz und der Sauerstoffsättigung [35].

3.2.2. Software zur Analyse der Daten

In der vorliegenden Arbeit werden die Daten sowohl logisch als auch physikalisch in der Post Mortem und der Carving Methode gesichert. Folgende Tools dienen hierbei der Datenanalyse:

- adb: Die von Android bereitgestellte Software adb ist ein Kommandozeilenprogramm, das die Interaktion zwischen einem Computer und einem Android Gerät erlaubt. Beispielsweise können Anwendungen installiert oder deinstalliert werden oder es wird der Shell-Zugriff genutzt, um Befehle auf dem Android Gerät auszuführen. Eine genaue Beschreibung des Tools und aller Anwendungsfälle sind auf der Internetseite von Android Developers [30] zu finden unter dem Pfad /tools/adb.

- Tool zur Extraktion von Daten des LKA Baden-Württemberg: Daten aus der physikalischen Sicherung des Endgeräts werden in diesem Programm aufbereitet, teilweise entschlüsselt und zur Analyse bereitgestellt.

3.2.3. Konfiguration und Rootvorgang

Bevor auf das Dateisystem der Smartphones zugegriffen wird, werden diese auf ihre Werkseinstellungen zurückgesetzt, um zuvor gespeicherte Daten vollständig zu löschen. So kann die Anzahl der auf dem Gerät vorhandenen Dateien reduziert werden. Alle Verzeichnisse sind entweder bereits werksseitig installiert oder stehen im Zusammenhang mit der Nutzung einer Smartwatch. Verzeichnisse, die persönliche Daten wie Bilder oder Kalenderdaten speichern, sind für die in der Arbeit durchgeführten Untersuchungen nicht relevant, da sie bereits von Odom et al. [8] eingehend untersucht wurden.

Um ein Smartphone rooten zu können, ist es notwendig die Entwickleroptionen des Smartphones zu aktivieren. Hierzu wird auf dem Smartphone in das Verzeichnis Einstellungen > Telefoninfo > Softwareinformationen navigiert. Im Anschluss wird der Listenpunkt Buildnummer so oft angeklickt, bis das Feld „Entwickleroptionen aktiviert“ zu sehen ist. Das Verzeichnis Einstellungen ist nun um den Listenpunkt „Entwickleroptionen“ erweitert. Um Zugriff über adb auf das Smartphone zu erhalten, muss USB-Debugging (Universal Serial Bus (USB)) erlaubt werden. Erforderlich ist desweiteren die Entsperrung des Bootloaders durch Aktivieren des Listenpunkts „OEM-Entsperrung “ (Original Equipment Manufacturer (OEM)). Dann können die im weiteren Verlauf genannten Anwendungen installiert werden, um das Gerät zu rooten und die Prüfung des Knox Warranty Bits, wie in Kapitel 3.3 beschrieben, zu umgehen:

- Odin [1]: Odin ist ein Tool, um Firmware auf Samsung™ Geräte zu flashen. Auf der offiziellen Webseite sind eine Schritt-für-Schritt Anleitung für Root Vorgänge, sowie weitere wichtige Hinweise zu finden. Die Firmware, die Odin auf Samsung™ Geräte flashen kann, besteht aus den Teilen:

  AP Das Android Package (AP) besteht aus der Benutzeroberfläche und der Android Software.

  BL Da der Bootloader (BL) die Integrität der Firmware prüft, muss dieser ebenfalls ersetzt werden, wenn eine neue Firmware auf dem Gerät installiert wird.

  CS Die Datei mit dem Namen Carrier Specific (CS) enthält spezifische Einstellungen, die je nach Mobilfunkanbieter abweichen.

  CSC In der Datei Consumer Software Customization (CSC) sind länderspezifische Anpassungen und Einstellungen enthalten.

- Magisk [2]: Die Anwendung ermöglicht es Nutzenden, Root-Zugriff auf ihr Endgerät zu erhalten. Wird das Smartphone mithilfe der Anwendung gerootet, ist es außerdem möglich Module zu installieren, die helfen Probleme von gerooteten Geräte, wie in Kapitel 3.3 be-

[1] Zu finden unter https://odindownload.com, Zugriff am 05. Oktober 2024
[2] Zu finden unter https://github.com/topjohnwu/magisk/releases, Zugriff am 10. Oktober 2024

schrieben, zu beheben. Magisk bietet außerdem die Möglichkeit einen Rootvorgang rückgängig zu machen.

- LSPosed[3]: Ein Open Source Framework, über das das Programm KnoxPatch installiert werden kann. Hier wird das sogenannte Android Runtime (ART) Hooking eingesetzt, bei dem zur Laufzeit einer Anwendung in den Programmablauf eingegriffen und dieser verändert wird.

- KnoxPatch[4]: In der Version KnoxPatch v 0.4.5 ist die Anwendung in der Lage die Prüfung des Knox Warranty Bits der Samsung™ Anwendungen Samsung™ Health und Samsung™ Health Monitor zu umgehen.

Folgende Anwendungen werden in der vorliegenden Arbeit untersucht:

- Samsung™ Health: Eine Gesundheits- und Fitnessanwendung der Firma Samsung™. Nutzende können hier ihre Aktivitäten verfolgen und Daten zu verschiedenen Ereignissen erfassen. In Tabelle 3.2 folgt eine Übersicht darüber, welche Daten automatisch aufgezeichnet werden und welche von Nutzenden selbst eingefügt werden können.

Tabelle 3.2.: Daten aus Samsung™ Health laut Benutzerhandbuch [9].

Automatische Erfassung	**Manuelle Erfassung**
- Training	- Training (in der Anwendung
- Anzahl gelaufener Schritte	unter Workout aufgeführt)
- Aktive Zeit	- Menstruationszyklus
- Aktivitätskalorien	- Essen, Wasser
- Daten zur Bewegung bestehend aus:	- EKG
Etagen, Aktive Stunden, Traningszeit,	- BIA
Aktivitätszeit mit erhöhtem Puls,	- Blutdruck
Trainingskalorien	
- Stress (im Tagesverlauf)	
- Schlaf	
- Puls	

[3]Zu finden unter https://github.com/LSPosed/LSPosed, Zugriff am 10. Oktober 2024
[4]Zu finden unter https://github.com/salvogiangri/KnoxPatch, Zugriff am 10. Oktober 2024

Trainings können aktiv von Nutzenden gestartet werden oder durch die automatische Trainingserkennung der Smartwatch angestoßen werden.

- Samsung™ Health Monitor: Die Anwendung erweitert die Funktionen der Samsung™ Health Anwendung und bietet Nutzenden genauere Einblicke in ihre physiologischen Daten. Sie bietet die Möglichkeit die Messungen von EKG, Blutdruck und Körperzusammensetzung anzustoßen.

Zur Analyse der ausgetauschten Bluetooth Daten wird das S8 gerootet und zur Durchführung der Messungen mit der Smartwatch verbunden. Die Daten werden im Anschluss durch entsprechende Befehle in adb ausgelesen. Hierfür muss in den Entwickleroptionen des Smartphones die Option „Bluetooth HCI-Snoop-Protokoll aktivieren“ eingeschaltet sein. Das Host Controller Interface (HCI) bildet hierbei die Schnittstelle zwischen Smartphone und Bluetooth-Controller. Um eine Log-Datei zu erzeugen, ist ein Neustart des Geräts erforderlich.
Zur Analyse der Daten mit dem hausinternen Programm des LKA Baden-Württemberg wird die Smartwatch mit dem S22 verbunden. Nach Aufnahme der Daten wird eine physikalische Sicherung des Geräts vorgenommen.

3.3. Anforderungen

Im Bereich der digitalen Forensik gibt es eine Vielzahl an Herausforderungen, die für eine Untersuchung der Gerätefunktionen überwunden werden müssen.
Herstellerseitig wird kein Einblick in die Implementierung der Software ins jeweilige Endgerät gewährt. Daten werden häufig nur verschlüsselt abgelegt und können auch auf gerooteten Geräten nicht ungehindert ausgelesen werden. Zwar ist das Betriebssystem Android ein Open Source Projekt, dessen Quellcode öffentlich einsehbar ist, allerdings ist Android Teil der Firma Alphabet Inc. und ein erheblicher Teil der Anwendungen wird von proprietärer Software des Konzerns begleitet. Ebenso sind die für diese Arbeit zu untersuchenden Anwendungen Samsung™ Health und Samsung™ Health Monitor proprietäre Software der Firma Samsung™ . Die Verarbeitung der durch die Sensoren gewonnen Daten im Quellcode ist nicht einsehbar und

muss durch Analysen ausgewertet werden. Hierbei ist zu beachten, dass nicht alle Daten problemlos durch das Programm adb ausgelesen werden können, da ein erheblicher Teil zum Schutz der Privatsphäre verschlüsselt abgelegt ist. Auch das Programm des LKA Baden-Württemberg stößt bei der Verarbeitung bestimmter Datenbanken an technische Grenzen. Komplex bei der Betrachtung von Android Geräten ist außerdem die Vielzahl an Herstellern, die auf das Betriebssystem zurückgreifen. Diese Vielzahl führt dazu, dass die extrahierten Dateiformate nicht einheitlich sind und sich auch der Ort, an dem die Daten zu finden sind, von Hersteller zu Hersteller unterscheiden kann [36]. Da nicht alle Geräte aller Hersteller problemlos durch polizeiliche Ressourcen gesichert werden können, ist es von großer Bedeutung zu wissen, wo Daten abgelegt werden. Ist ein Gerät beispielsweise nicht zugänglich, kann überprüft werden, ob Daten auch in der Cloud abrufbar sind. Je nach Synchronisationsintervall können hier Daten zu finden sein, die wichtige Hinweise liefern. Die Größe der Dateisysteme erschwert es, eine abschließende Aussage zur Vollständigkeit der untersuchten Daten zu treffen.
Um Verbraucher vor Hacking Angriffen zu schützen, werden diverse Sicherheitsmechanismen implementiert, die bei der Untersuchung der Geräte zu umgehen sind. Samsung™ nutzt die Sicherheitsplattform Samsung™ Knox zum Schutz sensibler Daten, zur Bereitstellung sicherer Container und zur Wartungs- und Integritätsprüfung der Software [37]. Samsung™ Knox integriert dabei das durch Android bereitgestellte sichere Betriebssystem Trusty [12]. Laut Atamli-Reineh et al. [38] existieren mehrere Angriffsvektoren, um die Sicherheitsprüfung von Knox zu umgehen. In Trusty gibt es das Secure-Flag, um sensible Daten zu schützen. Ist das Secure-Flag aktiviert, können keine Screenshots gemacht werden. In nicht sicheren Bildschirmen wird die aktuelle Ansicht der Anwendung nicht geteilt sondern durch einen Sperrbildschirm versteckt. Nicht sicher sind Bildschirme beispielsweise, wenn die Anwendungen in den Hintergrund treten, um geschlossen zu werden. Nutzende mit Root-Zugriff können das Secure-Flag deaktivieren. So können Sicherheitsvorkehrungen, die durch Trusty in Knox implementiert sind, teilweise ausgehebelt werden, was den Zugriff auf sensible Daten freigibt. In der vorliegenden Arbeit prüfen die erforderlichen Anwendungen das Knox Warranty Bit. Wird das Bit gesetzt, ist die Garantie des Geräts ungültig und ein Start der Anwendung wird abgebrochen. Durch die Installation der in Kapitel 3.2 beschriebenen Software können Samsung™

Health und Samsung™ Health Monitor ungehindert starten.
Das Rooten eines Smartphones ist mit vielen Risiken verbunden. Durch das Deaktivieren der Sicherheitsmaßnahmen, die durch Samsung™ Knox implementiert sind, ist das Smartphone anfälliger für Malware und Sicherheitslücken. Anwendungen erhalten tiefere Systemzugriffe und können so auf sensible Daten zugreifen. Viele Anwendungen, die beim Rooten eines Smartphones unterstützen sollen, können schadhaften Code enthalten und so Informationen unbemerkt weitergeben. Im schlimmsten Fall können fehlerhafte Modifikationen am Gerät dazu führen, dass die Funktionalität des Smartphones eingeschränkt ist oder vollständig verloren geht [38]. Kann ein Smartphone nicht mehr gestartet werden, ist eine komplexe und zeitaufwändige Wiederherstellung notwendig.

4. Ergebnisse

4.1. Untersuchung der Verzeichnisstruktur

Zunächst wurde eine Sicherung aller Daten vorgenommen. Für das S8 wurden die Daten mittels adb extrahiert. Datenbanken der zu untersuchenden Anwendungen liegen nur verschlüsselt vor, weshalb die Auswertung der Messungen auf Grundlage der physikalischen Sicherung des S22 vom LKA Baden-Württemberg erfolgt. So liegen Daten teilweise entschlüsselt vor und auch auf bereits gelöschte Elemente kann teilweise zugegriffen werden. Um das Dateisystem des Geräts mittels adb zu untersuchen, muss das Smartphone mit einem Computer verbunden werden. Im Terminal wird dann das Programm mit dem Befehl *adb start-server* gestartet. Daraufhin wird die Verbindung zum Gerät mit *adb devices* überprüft. Sind mehrere Geräte mit dem Computer verbunden, muss die hier angezeigte Identifikationsnummer in den Befehl eingefügt werden. Da in diesem Fall nur ein Gerät verbunden ist, wird im weiteren Verlauf darauf verzichtet. Um innerhalb des Verzeichnisses des Geräts navigieren zu können, muss zunächst in die Shell-Umgebung gewechselt werden. Hierfür wird der Befehl *adb shell* verwendet. Um uneingeschränkten Zugriff auf die Verzeichnisse zu erlangen, müssen Root Rechte innerhalb der Shell gewährt werden. Hierfür ist der Befehl *su*, für SuperUser, notwendig. Darüber hinaus muss die Shell als Superuser in der Anwendung Magisk hinterlegt sein, wie in Abbildung 4.1 zu sehen.

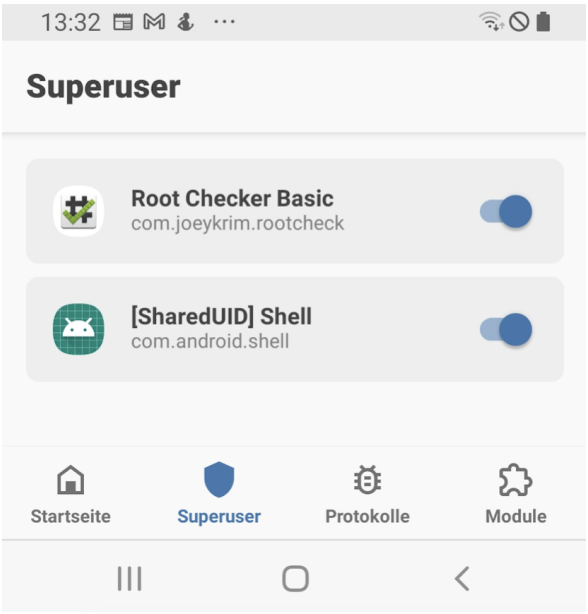

Abbildung 4.1.: Shell als Superuser.

Um alle Verzeichnisse anzuzeigen, wird der Befehl *ls* verwendet. In Tabelle A.7 ist eine Aufstellung der gefundenen Verzeichnisse der ersten Ebene.

Tabelle 4.1.: Verzeichnisse der ersten Ebene, Ausführen des Befehls ls in adb.

acct	atrace.rc	audit_filter _table	bin	bugreports
cache	charger	config	cpefs	d
data	default.prop	dev	efs	etc
factory	fstab.samsung- exynos8895	init	init.baseband.rc	init.carrier.rc
init.container.rc	init.environ.rc	init.rc	init.rilmtcp.rc	init.samsung- exynos8895.rc
init.samsung- exynos8895.usb .rc	init.usb. configfs.rc	init.usb.rc	init.zygote32.rc	init.zygote64 _32.rc
keydata	keyrefuge	lib	mnt	odm
oem	omr	overlay.d	plat_file _contexts	plat_hwservice _contexts

plat_property _contexts	plat_seapp _contexts	plat_service _contexts	preload	proc
product	publiccert.pem	root	sbin	sdcard
sepolicy	sepolicy _version	storage	sys	system
ueventd.rc	ueventdx _version exynos8895.rc	vendor _contexts	vendor_file	vendor _hwservice _contexts
vendor _property _contexts	vendor_seapp _contexts	vendor_service _contexts	vndservice _contexts	

Tabelle A.7 enthält eine Auflistung aller in den Verzeichnissen enthaltenen Funktionen. Verzeichnisse mit System- und Konfigurationsdateien werden von weiteren Analysen ausgeschlossen, da kein Zusammenhang mit den Daten der Smartwatch vermutet wird. Lediglich die Verzeichnisse *sdcard* und *data* werden weiter untersucht. Voller Zugriff wird bei ungerooteten Smartphones nur auf das Verzeichnis sdcard gewährt. Hier werden heruntergeladene Dateien, Alarme, Filme, Benachrichtigungen, Wiedergabelisten, Podcasts und Bilder der Benutzenden gespeichert. Der Einsatz von adb für die Dateien in diesem Verzeichnis ist nicht notwendig, da auch direkt über den Desktop darauf zugegriffen werden kann. Wie in Abbildung 4.2 zu sehen ist, unterscheiden sich die einsehbaren Verzeichnisse nicht.

Abbildung 4.2.: Gegenüberstellung des Zugriffs über adb (links) bzw. Navigation über den Desktop ins Verzeichnis sdcard (rechts).

Der Ordner log enthält keine aktuellen Log-Dateien, sondern wird nur im Verzeichnis sdcard abgelegt, wenn dies manuell angestoßen wird. Hierfür

wird in der Telefonanwendung im Eingabefeld ∗♯ *9900* ♯ eingegeben. So wird ein Menüfenster mit dem Namen SysDump sichtbar. Anschließend wird die in Abbildung 4.3 rot gekennzeichnete Option *Copy to sdcard(include CP Ramdump)* ausgewählt, siehe hierzu Abbildung 4.3. Das Menü SysDump unterscheidet sich vom Kommandozeilentool dumpsys in der Form, dass dumpsys nur ausgewählte Log-Dateien erstellt, während SysDump Log-Dateien zum gesamten System erstellt.

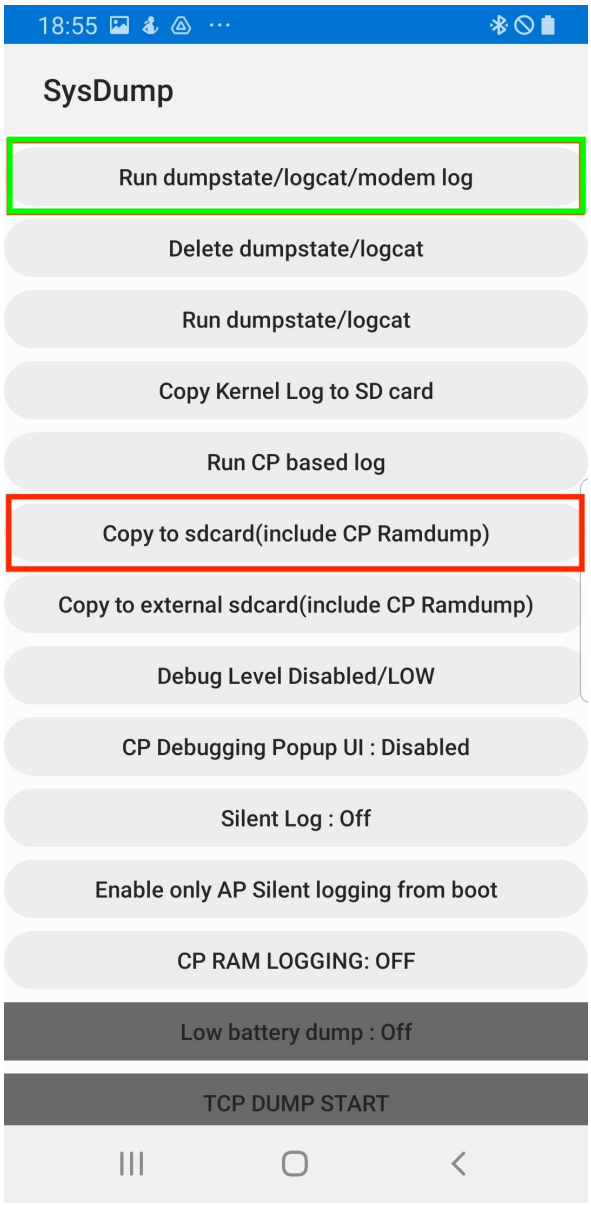

Abbildung 4.3.: SysDump Menü.

Soll ein umfassendes Log-Verzeichnis erstellt werden, wird der in Abbil-

dung 4.3 grün dargestellte Listenpunkt *Run dumpstate/logcat/modem log* ausgewählt und anschließend über adb ausgelesen. Zu finden sind die so erstellten Dateien unter dem Pfad *data/data/logs*.

4.1.1. Das Verzeichnis sdcard

Wie in Abbildung 4.2 zu sehen, sind im Verzeichnis sdcard nur wenige Ordner enthalten. Die Funktionen der Verzeichnisse sind im Anhang in Tabelle A.8 zu finden. Der Ordner Android enthält zwar ein data Verzeichnis, jedoch sind alle enthaltenen Ordner sind leer. Alle Verzeichnisse enthalten persönliche Inhalte der Nutzenden wie Bilder oder Musik. Im Folgenden relevant ist der Ordner log, auf dessen Inhalt in Kapitel 4.4 weiter eingegangen wird.

4.1.2. Das Verzeichnis data

Der Ordner data enthält alle Benutzerdaten und anwendungsspezifischen Informationen. Jede Anwendung hat einen eigenen Ordner.

Tabelle 4.2.: Verzeichnisse der zweiten Ebene.

DownFilters	adb	anr	app	app-asec
app-ephemeral	app-lib	app-private	app_fonts	backup
bio	bootchart	cache	camera	clipboard
custom_image	dalvik-cache	data	drm	enc_user
firmware	fota	hostapd	knox	local
log	lost+found	lxd	media	mediadrm
misc	misc_ce	misc_de	mptcp	nfc
nfc_log	ota	ota_package	overlays	pdb_bkup
property	resource-cache	sec	security	snap
snd	ss	ss_conn _deamon.pid	system	system_ce
system_de	tad	tombstones	user	user_de
vendor	vendor_ce	vendor_de	wifi	

Im Anhang in Tabelle A.9 ist eine vollständige Liste aller Verzeichnisse

aus data und deren Funktionen zu finden. Daten aus Anwendungen sind im Verzeichnis data zu finden. Dieses Verzeichnis enthält 261 Dateien, die nicht einzeln aufgeführt werden. Hier werden zu jeder Anwendung, die auf dem Smartphone installiert ist, Daten gehalten. Daten aus der Verwendung einer Smartwatch sind im Verzeichnis com.sec.android.app.shealth und com.sec.android.app.shealthmonitor zu finden. In beiden Verzeichnissen sind folgende Dateien enthalten.

Tabelle 4.3.: Verzeichnisse der vierten Ebene.

cache	code_cache	databases
files	no_backup	shared_prefs

Der Ordner databases enthält sämtliche Datenbanken, die zur Speicherung der von der Smartwatch aufgezeichneten Daten verwendet werden. Hierbei liegt eine Datenbank in unterschiedlichen Formaten vor. Dateien vom Format .db-shm sorgen für konsistente Daten innerhalb einer Datenbank. Die Abkürzung shm bedeutet Shared Memory und steht bezeichnend dafür, dass Daten hier verändert werden, bevor sie gespeichert werden. Dateien mit der Endung .db-wal enthalten Änderungen an der Datenbank, bevor diese in die Hauptdatenbank mit der Dateiendung .db geschrieben werden. Die Abkürzung wal steht dabei für write-ahaed-logging. Dateien mit der Endung .db-journal protokollieren die Übertragung von Daten in die Hauptdatenbank. Bei Systemabstürzen werden Datenbanken durch Dateien mit dieser Endung wiederhergestellt. Zur Auswertung der Daten werden die Hauptdatenbanken herangezogen. Datenbanken mit Inhalten zu Nutzungsbedingungen und Einverständniserklärungen zu Richtlinien werden nicht weiter untersucht.

4.2. Datentopologie

Daten, die durch Messungen der Smartwatch aufgenommen wurden, sind in den Datenbanken mit dem Namen *SecureHealthData.db*, *SHealthMonitor.db* und *sport.db* zu finden. Da alle Datenbanken verschlüsselt abgelegt und übertragen werden, werden im Folgenden Daten aus der Sicherung des hausinternen Programm des LKA Baden-Württemberg als Referenz

verwendet.
Daten, die manuell in der Anwendung Samsung™ Health Monitor aufgezeichnet wurden, sind auch in der Datenbank *SecureHealthData.db* zu finden. Die Datenbank *sport.db* enthält Aufzeichnungen zu geleisteten Trainings. Hierbei wird nicht unterschieden, ob das Training manuell angestoßen wurde oder ob die automatische Trainingserkennung die Aufzeichnung gestartet hat.
In der Datei *SecureHealthData.db* konnten Daten zu den in der Tabelle 4.4 aufgelisteten Datenbanken generiert werden. Unterschieden wird hier, wie in Kapitel 3.2.3, danach, ob eine manuelle Erfassung notwendig ist oder ob Daten automatisch aufgezeichnet werden. Außerdem sind die Stichpunkte in absteigender Reihenfolge danach sortiert, wie viele Tupel in den jeweiligen Tabellen generiert werden konnten.
Bis auf die Tabellen *android_metadata, datasource, delete_info, delete_info_flag, phd_delete_info* und *sqlite_sequence* haben alle Dateien den Präfix com_samsung_health oder com_samsung_shealth, auf den zur verbesserten Lesbarkeit im Folgenden verzichtet wird. Werden mehrere Dateien einem gemeinsamen Präfix zugeordnet, wird dies durch ein Stern-Symbol gekennzeichnet, um so den Unix- und SQLite-Konventionen [17] zu folgen. Die Unterscheidungen der tieferen Strukturen wird anschließend in Klammern aufgezählt.

Tabelle 4.4.: Daten aus SecureHealthData.db.

Automatische Erfassung	**Manuelle Erfassung**
- step_count	- user_profile
- heart_rate	- blood_pressure
- sleep ∗ (_stage, _combined)	- body ∗ (_fat, _muscle)
- stress ∗ (_daily, _histogram)	- caffeine_intake
- activity ∗ (_level,_day_summary)	- cycle ∗
- calories_burned	- electrocardiogram
- step_daily_trend	- food ∗ (_info, _intake, _nutrition,
- pedometer_day_summary	_favorite, _frequent)
- floors_climbed	- height
- badge	- oxygen_saturation
- best_records	- water_intake
- exercise ∗ (_recovery, _heart_rate)	- weight
	- preferences

Zeitstempel sind Tabellen als Unix Zeit[1] im Format Coordinated Universal Time (UTC) +0 hinterlegt. Um den tatsächlichen Zeitstempel zu berechnen müssen die Zeitzone und Zeitanpassung berücksichtigt werden. Die vorliegenden Daten wurden zur Sommerzeit generiert, was zu UTC+2 übersetzt werden kann. Zu den umgerechneten Stunden müssen folglich zwei Stunden addiert werden.
Eine weitere Gemeinsamkeit der Tabellen ergibt sich aus den Attributen Identifier (ID), Zeit der letzten Änderung, Startzeit der Messung, Aktualisierung der Messung, Erstellungszeit und Endzeit der Messung. Diese Attribute sind in allen Tabellen mit Zeitstempeln zu finden. Die ID ist hierbei eine fortlaufende Nummer, die jeder Messung vorangestellt wird. Auf die Bedeutung der verschiedenen Zeitstempel wird in Kapitel 4.3 weiter eingegangen.
Die meisten Tupel sind in der Tabelle *step_count* aufgezeichnet. Hier werden Daten zur gemessenen Anzahl an Schritten gespeichert. Ein Tupel aus der Tabelle *step_count* enthält die Attribute Schrittzahl, Geschwindigkeit, Strecke, Kalorienzahl, Endzeit der Messung, gerannte Schritte und gelaufene Schritte.

[1]Die Unix Zeit beschreibt die Anzahl der abgelaufenen Millisekunden seit dem 1. Januar 1970

In der Tabelle *heart_rate* werden Daten zur Herzfrequenz festgehalten. Ein Tupel besteht aus den Attributen Herzfrequenz max, Herzfrequenz min und durchschnittliche Herzfrequenz.
Daten zu aufgezeichneten Schlafphasen werden in der Tabelle *sleep_stage* verwaltet. Die konkreten Schlafphasen sind nicht im Klartext abgelegt. Die Zuordnung zu den Schlüsseln erfolgt im folgenden Kapitel.
In der Tabelle *stress* werden Daten zum gemessenen Stresslevel hinterlegt. Ein Tupel enthält die zusätzlichen Attribute Stress max, Stress min und Stresslevel. Bewertet wird Stress hierbei zwischen „niedrig“ und „hoch“ . Bewertet wird das Stresslevel über eine Messung der Herzfrequenzvariabilität, die Schwankungen im Zeitabstand zwischen aufeinanderfolgenden Herzschlägen angibt.
Die Anzahl der Tupel in den Tabellen *activity_day_summary* und *calories_burned* ist gleich, was auf einen Zusammenhang schließen lässt. In *activity_day_summary* sind die Attribute Zusammenfassung Trainingszeit, Zusammenfassung Schrittzählung, Ziel verbrannter Kalorien, Zusammenfassung aktive Zeit, Ziel überwundene Stockwerke, Zusammenfassung Zeitpunkt der Messung, Zusammenfassung überwundene Stockwerke und Zusammenfassung aktive Zeit hinterlegt. Hier sind folglich Daten zu täglichen Zusammenfassungen und Ziele zu finden. In *calories_burned* werden Daten zu täglich verbrannten Kalorien verwaltet durch die Attribute verbrannte Kalorien in aktiver Zeit, sonstige verbrannte Kalorien und verbrannte Kalorien in Trainingszeit.
Auf einen Zusammenhang kann auch bei den Tabellen *step_daily_trend* und *pedometer* geschlossen werden. In beiden Tabellen geht es um die Anzahl an Schritten, die durch die Smartwatch gemessen wurde. Die Tabelle *floors_climbed* verwaltet Informationen zu den überwundenen Stockwerken. In den Tabellen *badge* und *best_records* werden Erfolge bezüglich der Aktivität von Nutzenden verwaltet. Die Daten der Tabelle *exercise* verweisen auf die Herzfrequenz während aktiven Trainings.
Manuell können Daten in folgenden Tabellen erfasst werden. Die Tabelle *user_profile* verwaltet die Daten zum Benutzerprofil wie Größe, Gewicht, Name und E-Mail Adresse des verbundenen Nutzerkontos. Die Daten aus Messungen zu den Tabellen *blood_pressure*, *body* $*$*_fat* und *_muscle* und *electrocardiogram* können in der Anwendung Samsung™ Health Monitor angestoßen werden. Daten zur Flüssigkeits- und Nahrungsaufnahme, die in der Anwendung Samsung™ Health hinterlegt werden können, sind in

den Tabellen *caffeine_intake*, *water_intake* und *food* ∗ zu finden. Zudem können Daten zum Menstruationszyklus hinterlegt werden. Ist die Zykluserfassung eingeschaltet, werden automatisch Daten zur Körpertemperatur erfasst. Die Attribute Gewicht und Größe sind in den Tabellen *height* und *weight* hinterlegt. Benutzerspezifische Einstellungen sind in der Tabelle *preferences* abgelegt.
In weiteren Tabellen konnten keine Daten generiert werden.

4.3. Datenanalyse

Wie bereits im Kapitel 4.2 erwähnt, gibt es Attribute, die in mehreren Tabellen zu finden sind. Ein Beispiel hierfür sind die Attribute Zeit der letzten Änderung (last_modified), Startzeit der Messung, Aktualisierung der Messung(update_time), Erstellungszeit (create_time) und Endzeit der Messung. Obwohl die Namen der Bezeichner sehr sprechend gewählt wurden, bleibt Interpretationsspielraum für die Bedeutung der Attribute Zeit der letzten Änderung, Aktualisierung der Messung und Erstellungszeit. Es ist davon auszugehen, dass das Attribut Erstellungszeit auf die Erstellung des Tupels in der Datenbank schließen lässt. Da die Daten der Attribute Zeit der letzten Änderung und Aktualisierung der Messung nicht immer gleich sind, ist davon auszugehen, dass die Zeitstempel sich auf unterschiedliche Änderungen beziehen. Es ist wahrscheinlich, dass das Attribut Aktualisierung der Messung sich auf die tatsächlichen Messwerte bezieht, während das Attribut Zeit der letzten Änderung auf eine Änderung der Metadaten der Messung hinweist. Ändert sich beispielsweise der Status der Synchronisation der Messung, wird der Zeitstempel angepasst. Dies wird dadurch unterstützt, dass der Zeitstempel Erstellungszeit immer kleiner oder gleich dem Zeitstempel Aktualisierungszeit ist und dieser wiederum immer kleiner oder gleich dem Zeitstempel der letzten Änderung.
Bevor eine Smartwatch mit einem neuen Smartphone verbunden werden kann, muss sie zurückgesetzt werden. Der Datenspeicher der Smartwatch wird dabei geleert und alle Konfigurationen werden auf die ursprünglichen Werkseinstellungen zurückgesetzt. Wenn die Verbindung getrennt wurde, bleiben die gesammelten Daten auf dem Smartphone gespeichert. Wird eine Verbindung erneut aufgebaut, können Daten, die vor der Unterbrechung der Verbindung gesammelt wurden, wiederhergestellt werden. Diese sind

dann logisch auf dem Smartphone und der Smartwatch einzusehen. In der Datensicherung des Smartphones werden diese Daten in der Datei *phd_delete_info* abgelegt. So können Daten aus den Tabellen *step_count* und *sleep* wiederhergestellt werden.
Im Folgenden wird zunächst analysiert welche Struktur Daten aufweisen, die auf konkrete Objekte zurückzuführen sind. Darauffolgend wird aufgezeigt welche Daten auf konkretes Nutzerverhalten hinweisen, wenn mehrere Objekte in Kombination zur Analyse herangezogen werden.

4.3.1. Daten mit konkreten Objekten

Tabelle 4.5.: Erfassung von Daten mit konkreten Objekten.

Datenobjekt	Eigenschaft	Wert
exercise	Joggen mit abwechselnden Tempi	10 min. schnell, 10 min. langsam
step_count	Schritte schnell normales Gehen Schritte langsam	je 1 Minute gehen, >110 Schritte 90<x<110 Schritte <90 Schritte
floors_climbed	Überwinden von Stockwerken	Überwinden von 1,2,3,4 Stockwerken zu Fuß / im Aufzug

In der Tabelle exercise werden Daten zu Trainings aufgezeichnet. Die Aufzeichnung eines Trainings kann manuell oder automatisch angestoßen werden. Laut Benutzerhandbuch der Samsung™ Galaxy Watch5 wird die automatische Trainingserkennung nach 10 Minuten kontinuierlichem Training oder 3 Minuten kontinuierlichem Laufen aktiviert. Einzusehen ist dies in der Spalte des Attributs source_type. Wird das Training automatisch aktiviert, ist eine vier hinterlegt. Wurde das Training manuell angestoßen wird das durch eine eins vermerkt. Die Aufzeichnung erfolgt hierbei ab Start des Trainings. Das Attribut exercise_type bezeichnet die Art des Trainings. Diese werden nicht im Klartext, sondern codiert hinterlegt. Die Nummern 1001, 1002 und 10007 können hierbei auf die Trainingsarten Gehen, Joggen und Zirkeltraining zurückgeführt werden. Vermerkt wird au-

ßerdem die minimale, maximale und durchschnittliche Herzfrequenz. Eine Rückverfolgung der Herzfrequenz über den Verlauf des Trainings ist nicht möglich. Auch nicht in der Tabelle *heart_ rate*, da Daten zur Herzfrequenz hier nur ca. stündlich gespeichert werden. Für Trainings aus der Kategorie Joggen wird eine Zusammenfassung der geleisteten Schritte hinterlegt, sowie die maximale Geschwindigkeit, die dabei erreicht wurde. Genauere Informationen erhält man, wenn man die Tabelle *step_ count* betrachtet. Hier sind minütliche Aufnahmen von Schrittgeschwindigkeit und -anzahl zu finden. Die Attribute step_count_run_step und step_count_walk_step ermöglichen eine Unterscheidung zwischen den Bewegungsarten Gehen und schnelles Laufen bzw. Joggen.
Um zu überprüfen wie gut die Schrittzahl mit real getätigten Schritten überein stimmt, wurden Schritte in unterschiedlichen Geschwindigkeiten getätigt. Bei den Versuchen fiel auf, dass die Armhaltung für die Genauigkeit der Messung entscheidend ist. Wird die Smartwatch weniger bewegt, weil die Hände nah am Körper gehalten werden und nicht, wie bei einer Gehbewegung, mit freien Händen neben dem Körper schwingen, nimmt die Genauigkeit der Werte stark ab. Betrachtet werden Werte aus der Tabelle *step_ count*. Um die Messwerte genau zuordnen zu können, wurden Pausen von einer Minute eingelegt. Der Intervall wurde so gewählt, da Werte in diesem Rhythmus bei aktiver Messung hinterlegt werden. Je weniger schnell die Schritte zurückgelegt werden, desto ungenauer ist das Ergebnis der Messung. Bei einer Geschwindigkeit von weniger als 90 Schritten pro Minute weicht das Messergebnis vom Protokoll um bis zu 20 Prozent ab. Je schneller die Schritte getätigt werden, desto genauer ist das Messergebnis. Gemessen wurde bis zu einer Anzahl von 140 Schritten pro Minute. Ein weiterer Faktor, der die Messgenauigkeit beeinflusst, ist die Tragedauer der Smartwatch am Handgelenk. Die Smartwatch beginnt erst nach etwa zwei Minuten mit der Aufzeichnung von Schritten. Dieser Vorgang entspricht den Empfehlungen im Benutzerhandbuch, das rät, den Beginn einer Messung abzuwarten, bis der Puls auf dem Display der Smartwatch angezeigt wird.
Um ein Tupel in der Tabelle *floors_ climbed* zu generieren, wurde ein Gebäude mit vier Stockwerken gewählt. Nacheinander wurden zwischen einem und vier Stockwerke überwunden. Zwischen Auf- und Abstieg wurden Pausen eingelegt. Die Stockwerke des Gebäudes sind sowohl über einen Aufzug zu erreichen, als auch über ein Treppenhaus. Über die vier Stockwerke

verteilt sind 76 Treppenstufen mit einer durchschnittlichen Treppenhöhe von 17cm. Es ergibt sich eine Gesamthöhe von 12,92m, pro Stockwerk folglich ein Höhenunterschied von je 3,23m. Laut der Internetseite von Samsung™ wird ein Stockwerk ab einer Überwindung von drei Metern von der Smartwatch aufgezeichnet. Tatsächlich liegt hier eine deutliche Abweichung der Messwerte zum Protokoll vor, wie Tabelle 4.6.

Tabelle 4.6.: Vergleich der Messwerte für *floors_climbed* zum Protokoll.

ID	Unix Zeit	Datum	Stockwerke	Protokoll
50	1729063717860	Mi, 16.10.24, 9:28:37Uhr	4	3
51	1729065047000	Mi, 16.10.24, 9:50:47Uhr	2	4
52	1729065422000	Mi, 16.10.24, 9:57:02Uhr	1	3
53	1729065759000	Mi, 16.10.24, 10:02:39Uhr	2	2
54	1729066022000	Mi, 16.10.24, 10:07:02Uhr	1	1

Stockwerksüberwindungen werden nur in positiver Richtung von der Smartwatch aufgezeichnet. Eine Abwärtsbewegung wird nicht aufgenommen. Zuerst wurden die Stockwerke zwischen 9:28 Uhr und 9:47 Uhr mit dem Aufzug überwunden, anschließend erfolgte ab 9:49 Uhr die Messung durch Treppenlaufen. Für ein Überwinden der Stockwerke durch Fahren mit einem Aufzug gibt es nur einen Wert. Bei den Messungen 51 bis 54 wurde gelaufen. Bei dem Messwert mit der Nummer 50 sind vier Stockwerke eingetragen, laut Protokoll wurden zu dieser Zeit nur drei Etagen überwunden. Für alle anderen Fahrten mit dem Aufzug wurden keine Messwerte aufgezeichnet. In der Tabelle *step_count* wurden durch das Betreten und Verlassen des Aufzugs weiterhin Werte aufgezeichnet. Eine Fehlfunktion der Smartwatch kann folglich ausgeschlossen werden. Zwischen 9:28 Uhr und 9:50 Uhr wurden nur 74 Schritte in vier Tupeln aufgezeichnet. Ab 9:50 Uhr werden in *step_count* minütlich Messwerte aufgezeichnet, was auf eine deutlich erhöhte Aktivität zurückzuführen ist. Insgesamt ergibt sich für diese Zeit, in der die Treppen auf- und abgegangen wurden, eine aufgezeichnete Schrittzahl von 818 Schritten. Es kann also daraus geschlossen werden, dass die Aufzeichnungen überwundener Stockwerke deutlich genauer ist je höher die Aktivität des Nutzenden, desto valider sind die hinterlegten Messwerte für überwundene Etagen. Dies gilt sowohl für Fahrten mit dem Aufzug als auch für das Überwinden durch Treppensteigen. Der Messung

mit ID 50 aus Tabelle *floors_climbed* gehen in der Tabelle *step_count* 13 Tupel voraus mit einer gesamt gemessenen Schrittzahl von 676 Schritten. Daraus kann geschlossen werden, dass die Messung durch Schritte aktiviert werden muss, um eine Bewegung im Aufzug feststellen zu können.

4.3.2. Daten ohne konkrete Objekte

Tabelle 4.7.: Erfassung von Daten ohne Objekte, mit Zeitstempel.

Eigenschaft	Wert
Distanz überwinden ohne Schritte	10 min. schieben in einem Rollstuhl
Smartwatch wird angelegt / abgenommen	Anlegen /Abnehmen in Intervallen zwischen 1-10 min.
Uhr wird in eigenständigem / verbundenem Modus betrieben	Smartphone ist außerhalb / innerhalb der Reichweite des Smartphones bei Messungen aus Tabelle 4.5.

Um herauszufinden, welche Objekte erzeugt werden, wenn eine Person eine Distanz überwindet, ohne selbst Schritte zu tätigen, werden die Inhalte mehrerer Tabellen überprüft. Zuerst wurde eine Strecke zu Fuß zurückgelegt, um die Messung der Smartwatch anzustoßen. Anschließend wurde die Testperson mit Smartwatch am Handgelenk für eine Dauer von zehn Minuten in einem Rollstuhl bewegt. In keiner der Tabellen konnten Werte mit den entsprechenden Zeitstemepeln gefunden werden. Der Versuch wurde wiederholt, indem die Testperson zunächst lief und anschließend für eine Dauer von zehn Minuten auf einem Stuhl sitzend verblieb. Die Ergebnisse können nicht voneinander unterschieden werden. Wird eine Person folglich verschoben, ohne dass hierbei eine Laufbewegung ausgeführt wird, kann das aus den Daten der Smartwatch nicht abgeleitet werden. Eine Möglichkeit für zukünftige Messungen ist die Aktivierung des GPS-Sensors. Dieser erfasst Geodaten zur Strecke, die ein mobiles Endgerät zurücklegt. Da in der vorliegenden Arbeit keine SIM-Karte verwendet wurde, liegen hierzu keine Daten vor. Dies trifft sowohl für den eigenständigen, als auch für den verbundenen Modus zu.

Ein Anlegen oder Abnehmen der Smartwatch kann aus Daten der Tebellen nicht eindeutig nachvollzogen werden. Wird die Smartwatch über Nacht abgelegt, fehlen Daten für diesen Zeitraum in der Tabelle *sleep*. Die Abwesenheit von Daten in der Tabelle *step_count* muss nicht darauf hindeuten, dass die Smartwatch nicht getragen wurde, sondern kann genauso gut auf eine ruhige Tätigkeit hinweisen. Werte zur Herzfrequenz werden in der Tabelle *heart_rate* nur über den Verlauf einer Stunde zusammengefasst, eine minütliche Nachverfolgung ist nicht möglich. Startzeitpunkt der Messung ist der Beginn einer vollen Stunde. Der letzte Zeitpunkt zu dem gemessen wird, ist vor Ablauf der Stunde. Auszulesen sind dann die minimale, maximale und durchschnittliche Herzfrequenz innerhalb dieses Intervalls. Weicht der Startzeitpunkt der Messung vom Format hh:00:00 ab, weist das auf ein Anlegen der Uhr nach einem Intervall von mindestens einer Stunde hin. Wurde die Smartwatch in kürzeren Abständen abgelegt und wieder angelegt, ist das aus den vorliegenden Tabellen nicht abzuleiten.
Wurde die Verbindung zwischen Smartwatch und Smartphone getrennt, bleiben die Intervalle der Messungen unverändert. Der Übergang von eigenständigem zu verbundenem Modus kann durch eine Analyse der Zeitstempel identifiziert werden.

Tabelle 4.8.: Messwerte im eigenständigen Modus, Tabelle *heart_rate*.

ID	letzte Änderung	Startzeit der Messung	Aktualisierung der Messung	Erstellungszeit
28	Mo, 05.08.24, 8:41:58Uhr	Sa, 03.08.24, 15:00:00Uhr	Sa, 03.08.24, 16:04:26Uhr	Sa, 03.08.24, 15:02:26Uhr
29	Mo, 05.08.24, 8:41:58Uhr	Sa, 03.08.24, 16:00:00Uhr	Sa, 03.08.24, 17:03:35Uhr	Sa, 03.08.24, 16:04:27Uhr
30	Mo, 05.08.24, 8:41:58Uhr	Sa, 03.08.24, 17:00:00Uhr	Sa, 03.08.24, 18:01:07Uhr	Sa, 03.08.24, 17:03:35Uhr
31	Mo, 05.08.24, 8:41:58Uhr	Sa, 03.08.24, 18:00:00Uhr	Sa, 03.08.24, 19:02:24Uhr	Sa, 03.08.24, 18:01:07Uhr
32	Mo, 05.08.24, 8:41:58Uhr	Sa, 03.08.24, 19:00:00Uhr	Sa, 03.08.24, 19:02:25Uhr	Sa, 03.08.24, 19:02:25Uhr

Beispielhaft wird der Übergang vom eigenständigen in den verbundenen Modus anhand von Daten aus der Tabelle *heart_rate* dargestellt. In Tabelle

4.8 ist zu sehen, dass die Attribute Startzeit der Messung, Aktualisierung der Messung und Erstellungszeit alle vom Samstag, den 03. August stammen. Die letzte Änderung der Daten ist vom Montag, den 05. August. Der Akku der Smartwatch war leer und hat eine letzte Sicherung der Daten um 19:02:25 angestoßen. Als der Akku der Smartwatch geladen war, wurde die Verbindung erneut aufgenommen und die Daten wurden von der Smartwatch auf das Smartphone übertragen. Dies ist zu sehen an dem Zeitstempel der letzten Änderung der für alle IDs aus Tabelle 4.8 gleich ist.

4.4. Untersuchungen des Datenverkehrs

Wie zuvor beschrieben wurde der Listenpunkt *HCI-Snoop Protokoll aktivieren* in den Entwickleroptionen der verwendeten Geräte eingeschaltet. Im Verzeichnis log ist der Ordner GearLog zu finden, der erzeugt wird, wenn ein Smartphone mit einer Smartwatch bei aktivem Logging verbunden ist. In Tabelle A.10 ist eine Aufstellung der darin enthaltenen Log-Dateien und deren Inhalte zu finden. Interessant sind hier vor allem zwei Dateien. In der Datei Heart_dumpState-WatchDump wird der Status der Smartwatch verwaltet. Hier kann genau nachverfolgt werden, wann eine Smartwatch angelegt oder abgenommen wird. Dies wird durch den Infrarotsensor auf der Rückseite der Smartwatch gemessen.

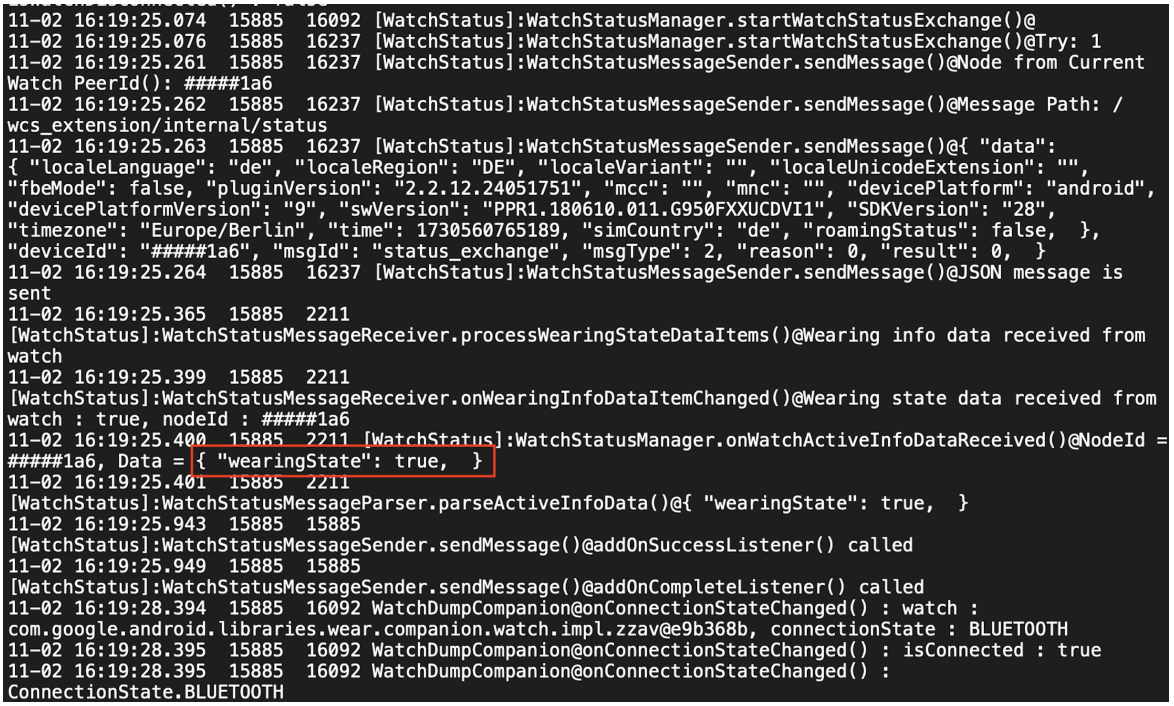

Abbildung 4.4.: Log-Datei zum Verbindungsaufbau zwischen Smartphone und Smartwatch.

In Abbildung 4.4 ist eine log Datei zum Verbindungsaufbau zwischen Smartphone und Smartwatch zu sehen. Die ersten fünf Zeichen einer Zeile bezeichnen das Datum im Format mm-dd, anschließend folgt ein Zeitstempel. Die fünfstellige Nummer der dritten Spalte bezeichnet die Identifikationsnummer des Prozesses, die darauffolgende Nummer stellt die Identifikationsnummer des Threads dar, in dem der Prozess ausgeführt wird. Anschließend ist der Datenverkehr gelistet. Es ist zu erkennen, dass Nachrichten im JSON[2] Format übertragen werden. Daten wie die IMEI Nummer des Geräts liegen nur teilweise im Klartext vor. Wichtig ist hier die Variable wearingState, die anzeigt, ob eine Smartwatch getragen wird oder nicht. In Abbildung 4.4 ist diese mit true gekennzeichnet. Wird eine Smarwtatch abgenommen, wird die Variable auf false gesetzt. Siehe hierzu folgende Abbildung 4.5.

[2]JavaScript Object Notation, beschreibt ein textbasiertes Datenformat, das zum Austauscg von Daten zwischen Anwendungen verwendet wird. Siehe RFC 8259, https://datatracker.ietf.org/doc/html/rfc8259, Zugriff am 09.11.2024

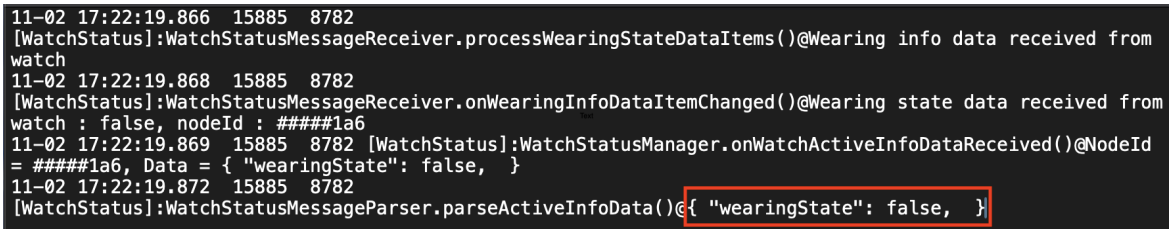

Abbildung 4.5.: Log-Datei zum Verbindungsabbau zwischen Smartphone und Smartwatch.

In derselben Datei kann auch nachverfolgt werden, ob die Smartwatch im eigenständigen oder verbundenen Modus operiert. Hierfür steht die Variable ConnectionState. Wird die Verbindung zwischen Smartwatch und Smartphone getrennt, so wird die Variable auf ConnectionState.Disconnected gesetzt. Wird die Verbindung erneut aufgenommen, wird eine neue Prozess ID erzeugt und der Status der Verbindung geändert. Außerdem wird bei jeder Änderung des Verbindungsstatus auch die Variable wearingState geprüft. Eine Smartwatch kann mit dem Smartphone erneut eine Verbindung aufnehmen, ohne dabei am Handgelenk getragen zu werden. Dieser Ablauf wurde mehrfach wiederholt. In allen Fällen wurde korrekt erkannt, ob die Smartwatch am Handgelenk getragen wurde oder nicht.
Aus der Log-Datei NS_Bridge, wobei die Abkürzung für NotificationBridge steht, können weitere Informationen zum Status der Verbindung zwischen Smartwatch und Smartphone entnommen werden. Wird eine Smartwatch vom Handgelenk abgenommen, ohne dass die Verbindung zum Smartphone unterbrochen wird, überprüft die Smartwatch den Status des Smartphones.

Abbildung 4.6.: Prüfen des Telefonstatus.

In der Abbildung 4.6 ist zu sehen, dass Daten zum Bildschirmzustand mit der Variablen isScreenOn, zur Bildschirmsperre mit der Variablen isKeyLocked und zur Hülle mit der Variablen isCoverOpen abgefragt werden. Zum Schutz vor Kratzern oder Stößen werden viele Smartphones in einer Hülle aufbewahrt. Durch Sensoren wird erkannt, ob die Hülle das Display des Smartphones bedeckt oder nicht. Das Öffnen der Hülle kann den Bildschirm des Smartphones aktivieren, was den Nutzungskomfort erhöht. Da in der vorliegenden Arbeit keine Hülle verwendet wird, ist die Variable immer

auf true. Wird eine Smartwatch abgelegt und die Verbindung bleibt erhalten, wird der Status des Smartphones nach 100 Minuten erneut abgefragt. Bei unveränderten Variablen wird die Verbindung zwischen Smartwatch und Smartphone mit der Nachricht *WATCH_ CONNECTED:false* unterbrochen.
Neue Einträge in den Log-Dateien konnten erst wieder gefunden werden, nachdem die Smartwatch wieder angelegt wurde. Das lässt darauf schließen, dass erst durch die Aktivierung der Smartwatch und den dadurch angestoßenen Verbindungsaufbau zum Smartphone wieder Nachrichten zwischen den Geräten ausgetauscht werden.
Zum Austausch konkreter Daten konnten keine Log-Dateien gefunden werden, da diese verschlüsselt übertragen werden. Weitere Informationen sind den Log-Dateien der Sicherung des LKA Baden-Württemberg zu entnehmen. Unter dem Pfad *data/data/com.sec.android.app.shealth/files/logs* sind Log-Dateien abgelegt, die über den Zeitraum der Messungen für die vorliegende Arbeit täglich automatisch angelegt wurden. Geht eine Smartwatch in den eigenständigen Modus über, wird dies in der Log-Datei mit STATE_DISCONNECTED angezeigt. Wird die Verbindung wieder aufgenommen, wird dies durch einen veränderten Verbindungsstatus angezeigt. JSON Objekte, die die Übertragung von Daten anzeigen, konnten nicht gefunden werden.

5. Fazit und Ausblick

Die Entwicklungen im Bereich des Internets der Dinge sowie der aufstrebenden Fitnesstrends haben dazu geführt, dass Fitness-Anwendungen auf Smartphones einen großen Markt bedienen. Zunehmend werden diese Anwendungen durch Daten unterstützt, die durch das Tragen einer Smartwatch aufgezeichnet werden können. Die Integration dieser Geräte und Anwendungen trägt nicht nur zur Verbesserung des persönlichen Fitnessmanagements bei, sondern eröffnet auch neue Perspektiven für forensische Untersuchungen, da die gesammelten Daten wertvolle Hinweise für die Ermittlungsarbeit in Strafverfahren liefern können.
In der Arbeit konnte gezeigt werden, dass Datenbanken im Zusammenhang mit diesen Anwendungen im Verzeichnis mit dem Pfad data/data/-com.sec.android.app.shealth gespeichert werden. Darüber hinaus konnten wichtige Informationen durch die Untersuchung von Log-Dateien im Ordner data/data/logs gewonnen werden. Extrahiert wurden diese Daten durch die Verwendung des öffentlichen Programms adb sowie durch ein Programm des LKA Baden-Württemberg. Durch die Aufnahme von Daten anhand von Messprotokollen wurde die Validität der gemessenen Daten überprüft. Eine wichtige Erkenntnis der Untersuchung ist, dass die Genauigkeit der Messung stark von einer aktiven Messung, also einer Bewegung der Smartwatch, abhängt. So konnte gezeigt werden, dass die erfasste Schrittzahl in Abhängigkeit von der Laufgeschwindigkeit sowie der Kontinuität der Messdauer von der tatsächlichen Schrittzahl abweichen kann. Es wurde jedoch festgestellt, dass keine Schritte aufgezeichnet werden, wenn keine tatsächliche Bewegung stattgefunden hat. Darüber hinaus konnte gezeigt werden, dass die Erkennung der Überwindung von Stockwerken durch die Nutzung eines Aufzugs nur dann erfolgt, wenn die Messung der Smartwatch zuvor durch eine Bewegung aktiviert wurde. Im Vergleich dazu erweist sich die Erkennung von Stockwerküberquerungen durch Treppensteigen als zuverlässiger, da diese Aktivität direkt und kontinuierlich erfasst wird. Die Untersuchungen haben außerdem gezeigt, dass die Smartwatch weiterhin

kontinuierlich Daten aufzeichnet, auch wenn die Verbindung zum Smartphone unterbrochen wurde. Eine vollständige Synchronisation der Daten erfolgt nach der Wiederaufnahme der Verbindung. Durch die Untersuchung der Log-Dateien konnten Erkenntnisse zur Kommunikation zwischen Smartwatch und Smartphone gewonnen werden. Während aus den Datenbanken keine direkten Rückschlüsse darauf gezogen werden können, ob sich eine Smartwatch tatsächlich am Handgelenk befand, ermöglichen spezifische Variablen in den Log-Dateien eine eindeutige Indikation hierfür.
In der vorliegenden Arbeit wurden die Daten vorwiegend bei vollem Akkustand aufgezeichnet. Weitere Analysen könnten sich darauf beziehen, welchen Einfluss der Energiesparmodus eines oder beider Geräte auf die Datenerfassung hat. Weitere Analysen könnten außerdem die Schlüssel nutzen, die durch Sicherungen des LKA gewonnen werden, um verschlüsselte Log-Dateien weiter zu untersuchen.
Im weiteren Verlauf können die Ergebnisse der Arbeit durch Messungen an verstorbenen Personen durch das LKA Baden-Württemberg erweitert werden. Ein Abgleich von Datenbanken und Log-Dateien zwischen lebenden und verstorbenen Personen kann im Fall von Gewaltverbrechen den Zeitpunkt des Todes eingrenzen und so zusätzlich gerichtsmedizinische Untersuchungen unterstützen.

A. Anhang

Anhang A - Protokoll Rooten des Smartphones Samsung Galaxy S8

Voraussetzungen:

- Odin: https://odindownload.com
- Firmware von SamFW: https://samfw.com
- https://github.com/topjohnwu/magisk/releases

Rootvorgang:

- Magisk auf Smartphone installieren
- Eine Datei auswählen und patchen, AP auswählen und patchen lassen

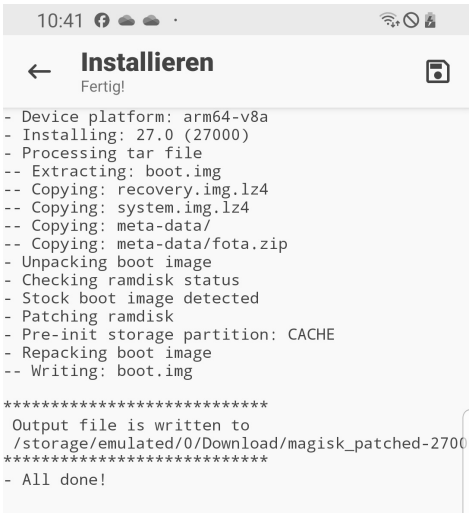

Abbildung A.1.: Screenshot vom 11.Oktober, Patch AP Datei durch Magisk.

- Download gepatchte AP Datei

- Smartphone in Download Modus
- Odin starten, BL, AP(gepatchte Version), CS und CSC Datei einfügen, Start drücken
- SetUp Assistenten des Smartphones durchlaufen
- Magisk neu auf Smartphone installieren

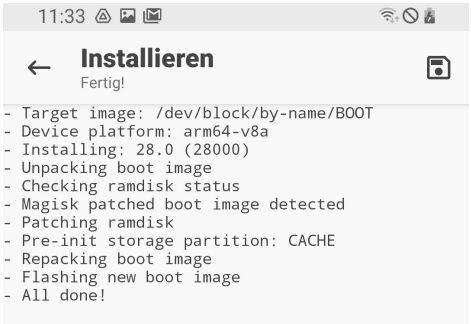

Abbildung A.2.: Screenshot vom 11.Oktober, AP Datei wird neu geflasht.

- Root bestätigen durch Root Checker App

LSPosed installieren:

- LSPosed: https://github.com/LSPosed/LSPosed/releases
- Zygisk in Magisk App aktivieren
- Magisk - Module - Module aus Speicher installieren

In LSPosed Knox Patch installieren

- LSposed Module: Knox Patch installieren
- Modul aktivieren
- Reboot
- Samsung Health in Knox Patch Modul aktivieren

Samsung Health App kann normal gestartet werden.

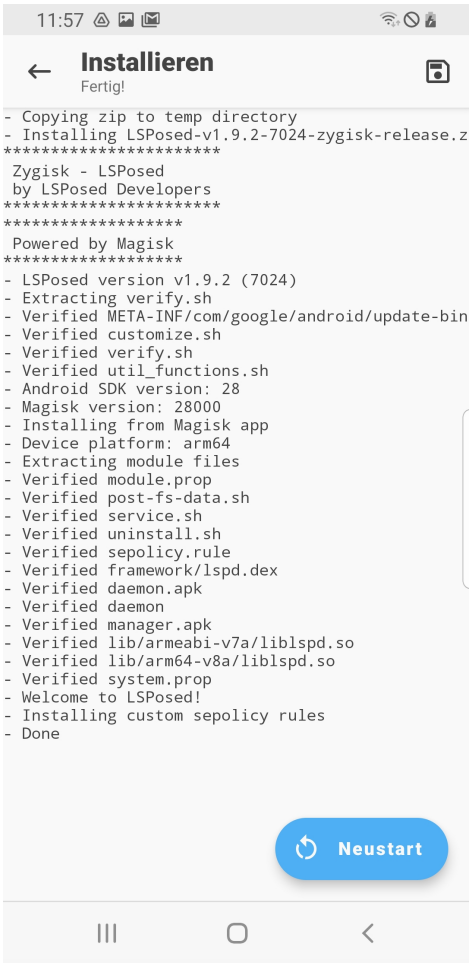

Abbildung A.3.: Screenshot vom 11.Oktober, LSPosed log.

Anhang B - Messpläne

Was soll gemessen werden?

Mit konkreten Objekten Objekte sind aus Inspektion der vorliegenden Tabelle bekannt. Tabellen exercise, step_count, floors_climbed.

Ohne konkrete Objekte Können Rückschlüsse gezogen werden auf Bewegungen des Nutzenden durch Kombination mehrerer Objekte? Welche Erkenntnisse können hier gewonnen werden? Was soll untersucht werden? Eine Distanz wird überwunden ohne, dass die Person mit Smartwatch selbst läuft (keine Armbewegung, wichtig für Gyroskop) durch Schieben

im Rollstuhl, simuliert Schieben einer verstorbenen Person auf Autopsietisch, Vergleichsweise Ruhiges Sitzen im Rollstuhl. Kann ein Unterschied festgestellt werden?
Wann gilt eine Smartwatch als angelegt bzw. abgelegt? Wo können Messwerte des Infrarot Sensors ausgelesen werden? Datenbank shealth.db, aus Metadaten oder Log-Dateien?
Kann festgestell werden, ob die Smartwatch in verbundenem oder eigenständigen Modus operiert hat? Kann Rückschlüsse darauf liefern, ob Smartwatch und Smartphone im selben Raum waren, oder ob Verbindung getrennt wurde durch Bewegen der Person weg vom Smartphone/ Wurde Smartphone außerhalb der Reichweite der verstorbenen Person gebracht, evtl. um Spuren zu verschleiern

- exercise: Tempounterscheidung, schnelles und langsames Joggen im 10min Wechsel Schrittzahl ist hier größer als 140Schritte pro Minute
- step_count: Schrittgeschwindigkeit anpassen, Schrittzahl wird durch Zählen ermittelt und protokolliert mit Zeitstempel
- floors_climbed: Aufzugfahren/ Treppensteigen, 4 Stcokwerke
- Distanz überwinden: Messung aktivieren durch Laufen, dann 10min fahren im Rollstuhl
- Vergleichsmessung zum Fahren im Rollstuhl: 10min Laufen, dann 10min ruhiges Sitzen, keine Schreibtischarbeit oä.
- Uhr anlegen/ablegen: Mit Zeitstempel, werden Daten in einer Tabelle erzeugt? (Außer regelmäßigen Daten wie Herzfreuqnez, alle Stunde, Schrittzahl wird nur bei aktiven Schritten aufgezeichnet,..) Nullschritte?
- verbundener/ eigenständiger Modus: Messungen werden in einem Durchlauf mit aufrechter Verbindung durchgeführt, im zweiten Durchlauf werden die Messungen ohne Smartphone durchgeführt.

Anhang C - Material

Auflistung nach Anhang B der Vorgaben zur Erstellung von Nutzerdaten nach [25]. Erfasst werden Marke und Modell des mobilen Endgeräts, IMEI, MEID, ESN, MSISDN, MIN. Darüber hinaus werden Daten zur Software der Endgeräte erhoben nach dem Vorgehen von Alabdulsalam et al. [22].

Tabelle A.1.: Dokumentation für mobile Endgeräte, Hardware S22.

Element	Wert
Endgerät: Marke, Modell	Samsung, Galaxy S22, SM-S901B/DS
IMEI/ MEID/ ESN	352539591107272/ 89043051202200005222008033608453/ R5CTA2SM14P
MSISDN / MIN	-

Tabelle A.2.: Dokumentation für mobile Endgeräte, Software S22.

Element	Wert
One UI-Version	6.0
Android Version	14
Software Version	SAOMC_SM-S901B_OXM_EUX_14_0062EUX/-/EUX/EUX

Tabelle A.3.: Dokumentation für mobile Endgeräte, Hardware S8.

Element	Wert
Endgerät: Marke, Modell	Samsung, Galaxy S8, SM-G950F
IMEI/ MEID/ ESN	358054080436442/ -/ R38J707D64K
MSISDN/ MIN	-

Tabelle A.4.: Dokumentation für mobile Endgeräte, Software S8.

Element	Wert
One UI-Version	1.0
Android Version	9
Software Version	SAOMC_SM-G950F_OXM_DBT_PP_0009 ce061716a19dd8360d DBT/DBT/DBT

Tabelle A.5.: Dokumentation für mobile Endgeräte, Hardware Watch5.

Element	Wert
Endgerät: Marke, Modell	Samsung, Watch5, SM-R915F
IMEI/ MEID/ ESN	351652591169803/ 89043051202200006222004134894004/ RFAT824DC7D
MSISDN/ MIN	-

Tabelle A.6.: Dokumentation für mobile Endgeräte, Software Watch5.

Element	Wert
One UI-Uhrersion	4.5
Systemversion, Wear-OS Version	11, 3.5
Sicherheitssoftware-Version	ASKS v4.0 Release 20230119

Anhang D - Datentopologie

Tabelle A.7.: Verzeichnisse und Funktionen der ersten Ebene.

Verzeichnis	Funktion
acct	
atrace.rc	Konfigurationsdatei, Performance-Analyse
audit_filter_table	Tabelle zur Aufzeichnung sicherheitsrelevanter Ereignisse unter Linux
bin	Enthält Systembefehle und -dateien
bugreports	Protokolldatei, Zustand des Geräts zu bestimmten Zeitpunkten
cache	Zugriff auf häufig benötigte Informationen
charger	Verwaltet Ladevorgang des Geräts
config	Configuration, Sämtliche Systemeinstellungen
cpefs	Common Platform Encryption File System, Verschlüsselung von Daten auf Android Geräten
d	Dalvik Cache, erhöht Ausführungsgeschwindigkeit von Apps (Java Bytecode -> Dalvik Byte Code -> Anwendung)
data	**Speichert alle Benutzerdaten und App-spezifischen Informationen**
default.prop	Steuert Verhalten des Android Betriebssystems
dev	Device, Schnittstelle zu Hardwarekomponenten
efs	Encrypting File System, Netzwerkverbindungen und Identifikation des Android Geräts
etc	Zusätzliche Konfigurationsdateien
factory	Werkseinstellungen für Zurücksetzen des Geräts hinterlegt
fstab.samsungexynos8895	Unterstützt gerätespezifische Funktionen, Exynos 8895 ist dabei der Prozessor im Gerät
init	Steuert Systemstart und -ablauf
init.baseband.rc	Initialisierung und Betrieb von Basisbanddiensten (Mobilfunkkommunikation)
init.carrier.rc	Netzwerkspezifische Einstellungen

Verzeichnis	Funktion
init.container.rc	Initialisierung und Verwaltung von Containern
init.environ.rc	Zuständig für Umgebungsvariablen
init.rc	Erstes Skript, das bei Systemstart ausgeführt wird
init.rilmptcp.rc	Steuert TCP Verbindungen
init.samsungexynos8895.rc	Unterstützt gerätespezifische Funktionen
init.samsungexynos8895.usb.rc	Unterstützt USB Verbindungen von Geräten mit Exynos 9985 Prozessor
init.usb.configfs.rc	USB-Konfiguration
init.usb.rc	Start USB-Verbindungen
init.zygote32.rc	Hook bei gerooteten Geräten für Laden der neuen Image Datei
init.zygote64_32.rc	Für Geräte mit 64 Bit Architektur die 32 Bit Anwendungen ausführen
keydata	Authentifizierungsdaten von Nutzern, Sicherheitsinformationen
keyrefuge	Integrität und Vertraulichkeit von Schlüsseln und sicherheitsrelevanten Daten
lib	Bibliotheken, die von Anwendungen und Prozessen genutzt werden
mnt	Mount, Zugriff auf Dateisysteme wird hier verwaltet
odm	Overlay Device Manager, Funktionen oder Anpassungen zur Benutzeroberfläche und zum Verhalten des Systems
oem	Original Equipment Manufacturer, spezifische Anpassungen, Konfigurationen und Dateien, vom Hersteller bereitgestellt
omr	Overlay Manager, benutzerdefinierte Anpassungen an der Benutzeroberfläche für Entwickler
overlay.d	Anpassen der Bernutzeroberfläche
plat_file_contexts	Verwaltung von Zugriffsrechten
plat_hwservice_contexts	Verwaltung von Zugriffsrechten
plat_property_contexts	Verwaltung von Zugriffsrechten
plat_seapp_contexts	Verwaltung von Zugriffsrechten
plat_service_contexts	Verwaltung von Zugriffsrechten

Verzeichnis	Funktion
preload	Vorinstallierte Anwendungen und Daten für den ersten Start des Geräts
proc	Prozessinformationen (laufende Prozesse, aktueller Zustand des Kernels)
product	Produktdateien (gerätespezifisch), wichtig bei Installation von Treibern
publiccert.pem	Enthält öffentliche Zertifikate und Schlüssel zur Überprüfung digitaler Signaturen
root	Enthält gesamtes System und dessen Partitionen
sbin	Superuser binary, Systemdateien und -anwendungen
sdcard	**Zugriff auf externe Speicherressourcen, Verwaltung von Mediendateien und App-Daten**
sepolicy	Schutz vor unbefugtem Zugriff und Sicherheitsverletzungen, SELinux
sepolicy_version	Versionsnummer der aktuellen Sicherheitsrichtlinien des Geräts SELinux
storage	Zugriff auf interne und externe Speicherressourcen des Geräts
sys	Informationen über den aktuellen Status des Kernels und die Hardware des Geräts
system	enthält grundlegende Systemdateien und -anwendungen, die für den Betrieb des Geräts erforderlich sind
ueventd.rc	Konfigurationsdatei zur Verwaltung von Geräteereignissen (Zuweisung von Treibern)
ueventd.samsungexynos8895.rc	Konfigurationsdatei zur Verwaltung von Geräteereignissen (Zuweisung von Treibern), speziell für diesen Prozessor
vendor	Unterstützt gerätespezifische Hardware
vendor_file_contexts	Gewährleistung der Sicherheit und Integrität des Android-Betriebssystems
vendor_hwservice_contexts	Sicherheitskontext für in vendor hinterlegte Hardware
vendor_property_contexts	Berechtigungen und Zugriffsrechte

Tabelle A.8.: Verzeichnisse und Funktionen aus sdcard.

vendor_seapp_contexts	Berechtigungen und Zugriffsrechte auf herstellerspezifische Anwendungen
vendor_service_contexts	Zugriff auf Systemressourcen
vndservice_contexts	Zugriff auf Systemressourcen
Verzeichnis	Funktion
Alarms	Speichert Alarme
Android	Enthält die Ordner data, obb (Opaque Binary Blob) zur optimierten Nutzung von Apps bei großen Datenmengen, media
DCIM	Digital Camera Images, Verwaltet Aufnahmen
Downloads	Heruntergeladene Dateien
log	Log-Dateien
Movies	Verwaltet Filmdateien
Music	Verwaltet Musikdateien
Notifications	Verwaltet Meldungen
Pictures	Verwaltet manuell gespeicherte oder heruntergeladene Bilddateien
Playlists	Verwaltung von Wiedergabelisten
Podcasts	Verwaltung von Podcasts
Ringtones	Verwaltung von Klingeltönen
Samsung	Enthält einen Musik Ordner und vorinstallierte Musik

Tabelle A.9.: Verzeichnisse und Funktionen aus data.

Verzeichnis	Funktion
DownFilters	Verwaltung und Speicherung von Dateien, die innerhalb einer App heruntergeladen werden
adb	Speichert Konfigurationsdateien und Log-Dateien bei der Nutzung von adb
anr	Application not responding, Speichert Log-Dateien, wenn eine Anwendung nicht reagiert. Dient der Fehlerbehebung
app	Speichert .apk Dateien der installierten Anwendungen (Android Package) intern
app-asec	Speichert .apk Dateien der installierten Anwendungen (Android Package) extern
app-ephemeral	Speichert temporäre .apk Dateien
app-lib	Anwendungen benötigen Zugrif auf diese Bibliotheken, um zu funktionieren (Schnittstelle Hardware/-Software)
app-private	Hier sind sensible Daten von Nutzenden gespeichert, die anderen Anwendungen nicht zugänglich sein sollen
app_fonts	Speichert benutzerdefinierte Schriftarten
backup	Speichert Sicherungskopien von Anwendungsdaten
bio	Speichert Daten im Zusammenhang mit biometrischen Informationen von Nutzenden
bootchart	Daten zu vergangenen Bootvorgängen, dient der Optimierung von Bootvorgängen
cache	Wird von Anwendungen als Zwischenspeicher genutzt
camera	Einstellungen der Kamera Anwendung eines Smartphones
clipboard	Zwischenspeicher
custom_image	Hier ist die image Datei gespeichert, die von der Anwendung Magisk verwendet wird
dalvik_cache	Speichert optimierte Bytecode-Dateien, um die Ladezeit von Anwendungen zu verkürzen

Verzeichnis	Funktion
data	**Hält Daten für jede installierte App in verschiedenen Datenformaten**
drm	Digital Rights Management, Lizenz- und Schlüsseldateien
enc_user	Speichert verschlüsselte Benutzerdaten
firmware	Speichert Firmware Daten und Ressourcen (Aktualisierungen, Support für Treiber, gerätespezifische Informationen)
fota	Firmware Over -The-Air, ermöglicht Over-The-Air Updates ohne Interaktion von Nutzenden
hostapd	Host Access Point Deamon, Netzwerkkonfigurationen, Verwaltung vonb Verbindungen mit Netzwerken
knox	Speichert Daten im Zusammenhang mit Samsung Knox
local	Speichert temporäre Dateien, notwendig für Betrieb von Anwendungen
log	**Enthält System- und Anwendungsprotokolle**
lost+found	Dateien aus unvollständigen Schreibvorgängen/nach Dateisystemfehlern
lxd	Linux Container Deamon, Verwaltung der Container basierten Virtualisierung
media	Media-Dateien, die von Anwendungen verwendet werden
mediadrm	DigitalRightsManagement, Speichert Informationen, die für den Zugriff auf geschützte Dateien notwendig ist
misc	miscellaneous (engl. verschiedenes/sonstiges), verschiedene/nicht kategorisierte Daten
misc_cs	configuration service
misc_de	device, gerätespezifische Daten
mptcp	Multipath TCP, Verwaltet erweiterte Funktionen des TCP-Protokolls
nfc	Near Field Communication, ermöglicht Kommunikation von Geräten mittels NFC

Verzeichnis	Funktion
nfc_log	Speichert Protokolldateien im Zusammenhang mit NFC
ota	Over-The-Air, Ermöglicht OTA Updates
ota_package	Pakete aus OTA Updates
overlay	Speichert Dateien aus Overlay Paketen, wird verwendet, um das System benutzerspezifisch anzupassen
pdp_bkup	Personal Data Protection, Speichert Daten, die zum Schutz sensibler Nutzerdaten beitragen
property	Notwendig zur Steuerung des Systemablaufs
resource-cache	Dient der Optimierung des Ressourcenverbrauchs des Endgeräts
sec	Speichert Sicherheitsdaten (Schlüssel/Zertifikate) und Authentifizierungsinformationen
security	Speichert Sicherheitsdaten (Schlüssel/Zertifikate) und Authentifizierungsinformationen und Zugrifskontrollen
snap	Snapdragon, Art des Prozessors, Konfigurationsdateien für Prozessor
snd	Sound, Dateien zur Speicherung von Audio- und Sounddateien
ss	System and Security, Systemsicherheit
ss_con_deamon.pid	Security Service Connection Deamon Process ID, Prozessidentifikation und -überwachung
system	Speichert Daten rund um Betriebssystem und Systemdienste ********
system_ce	Credential Encrypted, verschlüsselte System- und Zugangsdaten
system_de	System Data Encrypted, verschlüsselte Benutzereinstellungen und Anwendungsdaten **
tad	Telephony Analytics Data, Anrufprotokolle, Netzwerkstatistiken

Verzeichnis	Funktion
tombstones	Dient der Fehlerdiagnose und -protokollierung bei Abstürzen des Systems
user	Speichert Daten zu Benutzerprofilen
user_de	User Data Encrypted, verschlüsselte Benutzerdaten
vendor	Herstellerspezifische Einstellungen des Geräts
vendor_ce	Vendor Credential Encrypted, Verschlüsselte Daten, die herstellerseitig geladen werden
vendor_de	Vendor Data Encrypted, Verschlüsselte Daten, die herstellerseitig geladen werden
wifi	Speichert Daten zur Nutzung von WLAN Netzwerken

Tabelle A.10.: Log-Dateien und Informationen.

Dateiname	Information
Heart_dumpState-CM	Configuration Manager, verwaltet Einstellungen
Heart_dumpState-CMC	Carrier Management Client log, Aktivierung und Deaktivierung von Mobilfunkprofilen
Heart_dumpState-CS	Call Service, verwaltet Telefonanrufe
Heart_dumpState-ESIM	Aktivierungsprotokolle bei Verwendung einer embedded Subscriber Identity Module (eSIM)
Heart_dumpState-HM	Host Manager, verwaltet Interaktion zwischen den Geräten
Heart_dumpState-NS_AM	Network Service Access Manager, verwaltet Kommunikation mit Netzwerken
Heart_dumpState-NS_BRIDGE	NotificationsBridge, Übertragung von Benachrichtigungen
Heart_dumpState-QuickPanel	Zeigt Notifikationen auf dem „QuickPanel“ also der Anzeige der Smartwatch an
Heart_dumpState-SEARCH	Verwaltet getätigte Suchen
Heart_dumpState-SelfDiagnostic	Systeminformationen, Hardware Status und Fehlerprotokolle
Heart_dumpState-SelloutDir	Aktivierungsdaten der Smartwatch, Status und Profile von Mobilfunkdiensten
Heart_dumpState-SM	Service Manager, verwaltet Einstellungen zwischen Smartphone und Smartwatch
Heart_dumpState-STF	Smartphone Test Framework, Tests und Diagnoselogs für Qualitätsmanagement
Heart_dumpState-WApps	Watch Apps Manager, logs bzgl der Interaktion mit Anwendungen auf der Smartwatch
Heart_dumpState-WatchDump	Verbindungsdaten zwischen Smartwatch und Smartphone
Heart_dumpState-WF	WatchFace verwaltet verfügbare und gewählte Hintergründe und Updates für Design der Anzeige

Abkürzungsverzeichnis

adb	Android Debug Bridge
AP	Android Package
ART	Android Runtime
BIA	Bioelektrische Impedanzanalyse
BL	Bootloader
CS	Carrier Specific
CSC	Consumer Software Customization
EKG	Elektrokardiogramm
eSIM	embedded Subscriber Identity Module
ESN	Electronic Serial Number
HCI	Host Controller Interface
ID	Identifier
IMEI	International Mobile Equipment Identity
Inc.	Incorporated
IoT	Internet of Things (Internet der Dinge)
LKA	Landeskriminalamt
MEID	Mobile Equipment Identifier
MIN	Mobile Identification Number
MSISDN	Mobile Station Integrated Services Digital Network
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
S22	Samsung™ Galaxy S22
S8	Samsung™ Galaxy S8
SIM	Subscriber Identity Module

SMS Short Message Service

USB Universal Serial Bus

UTC Coordinated Universal Time

WLAN Wireless Local Area Network

Literatur

[1] Statista, *Volumen der jährlich generierten/replizierten Datenmenge weltweit von 2010 bis 2022 und Prognose bis 2027 (in Zettabyte)*, Zugriff am 05. Oktober 2024. Adresse: https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/.

[2] F. Sanchez-Hermosilla, "Feststellung der Täterschaft durch digitale Spuren in einem Schwurgerichtsverfahren," *erschienen in Kriminalistik*, S. 274–280, Ausgabe 05/2024.

[3] Techbook, *Fitnesstracker: Puls, Klopapier, Bananen*, Zugriff am: 17. Oktober 2024. Adresse: https://www.techbook.de/mobile-lifestyle/wearables/fitnesstracker-puls-klopapier-bananen.

[4] D. Uckelmann, M. Harrison und F. Michahelles, *Architecting the internet of things*, 1. Auflage. Berlin, Heidelberg: Springer-Verlag GmbH, 2011, ISBN: 978-3-642-19156-5.

[5] N. Golmie, N. Chevrollier und O. Rebala, "Bluetooth and WLAN coexistence: challenges and solutions," *erschienen in IEEE Wireless Communications*, S. 22–29, Ausgabe 10(6) 2003. DOI: 10.1109/MWC.2003.1265849.

[6] Y.-c. Kong, "A forensic analysis approach to smartphones from a criminal investigation perspective," *Masterthesis, University of Hong Kong*, 2015.

[7] H. Bernstein, *Messelektronik und Sensoren: Grundlagen der Messtechnik, Sensoren, analoge und digitale Signalverarbeitung*, 2. Auflage. Wiesbaden: Springer Fachmedien Wiesbaden, 2024, ISBN: 978-3-658-38929-1.

[8] N. R. Odom, J. M. Lindmar, J. Hirt und J. Brunty, "Forensic Inspection of Sensitive User Data and Artifacts from Smartwatch Wearable Devices," *erschienen in Journal of Forensic Sciences*, S. 1673–1686, Ausgabe 64(6) 2019. DOI: 10.1111/1556-4029.14109.

[9] Samsung, *Galaxy Watch5 eine Uhr für deine Ziele*, Zugriff am 05. Oktober 2024. Adresse: https://www.samsung.com/de/watches/galaxy-watch/galaxy-watch5-40mm-silver-lte-sm-r905fzsadbt/.

[10] Statista, *Marktanteile von Betriebssystemen in Deutschland seit 2009*, Zugriff am: 09. Oktober 2024. Adresse: https://de-statista-com.ub-proxy.fernuni-hagen.de/statistik/daten/studie/158102/umfrage/marktanteile-von-betriebssystemen-in-deutschland-seit-2009/.

[11] Statista, *Prognostizierte Marktanteile bei Smartphone-Betriebssystemen*, Zugriff am: 09. Oktober 2024. Adresse: https://de-statista-com.ub-proxy.fernuni-hagen.de/statistik/daten/studie/182363/umfrage/prognostizierte-marktanteile-bei-smartphone-betriebssystemen/.

[12] Android, *Trusty: Secure Execution Environment*, Zugriff am 08. Oktober 2024. Adresse: https://source.android.com/docs/security/features/trusty.

[13] Google, *Android Wear: Moving Forward Like Never Before*, Zugriff am 07. Oktober 2024. Adresse: https://android.googleblog.com/2014/09/android-wear-moving-forward-like.html.

[14] S.-T. Sun, A. Cuadros und K. Beznosov, "Android Rooting: Methods, Detection, and Evasion," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, Ser. SPSM '15, New York, NY, USA: Association for Computing Machinery, 2015, S. 3–14. DOI: 10.1145/2808117.2808126.

[15] D. Labudde und M. Spranger, *Forensik in der digitalen Welt: Moderne Methoden der forensischen Fallarbeit in der digitalen und digitalisierten realen Welt*. Berlin, Heidelberg: Springer Spektrum, 2017, ISBN: 978-3-662-53801-2.

[16] E. Casey, *Digital evidence and computer crime: Forensic science, computers, and the internet*, 3. Auflage. Academic Press, 2011, ISBN: 978-0-12-374268-1.

[17] M. Kofler, *Datenbanksysteme: Das umfassende Lehrbuch*, 1. Auflage. Bonn: Rheinwerk Verlag, 2022, ISBN: 978-3-367-10015-6.

[18] Z. Zhuang und Y. Xue, "Sport-Related Human Activity Detection and Recognition Using a Smartwatch," *Sensors*, Ausgabe 19(22) 2019. DOI: 10.3390/s19225001.

[19] T. Vilarinho, B. Farshchian, D. G. Bajer u. a., "A Combined Smartphone and Smartwatch Fall Detection System," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, S. 1443–1448. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.216.

[20] M. Masoumian Hosseini, S. T. Masoumian Hosseini, K. Qayumi, S. Hosseinzadeh und S. S. Sajadi Tabar, "Smartwatches in healthcare medicine: assistance and monitoring; a scoping review," Ausgabe 23(1) 2023. DOI: 10.1186/s12911-023-02350-w.

[21] R. Alharbi und W. H. Allen, "Collection and Analysis of Digital Forensic Data from Devices in the Internet of Things," in *2019 SoutheastCon*, 2019, S. 1–6. DOI: 10.1109/SoutheastCon42311.2019.9020349.

[22] S. Alabdulsalam, K. Schaefer, T. Kechadi und N.-A. Le-Khac, "Internet of Things Forensics – Challenges and a Case Study," in *Advances in Digital Forensics XIV*, G. Peterson und S. Shenoi, Hrsg., Cham: Springer International Publishing, 2018, S. 35–48.

[23] R. Montasari, H. Jahankhani, R. Hill und S. Parkinson, *Digital Forensic Investigation of Internet of Things (IoT) Devices* (Advanced Sciences and Technologies for Security Applications), 1. Auflage. Springer Cham, 2020, ISBN: 978-3-030-60425-7.

[24] K. O. Adebayo, "Digital Forensic Analysis of Smart Watches," 2020.

[25] R. Ayers, B. Livelsberger und B. Guttman, *Quick Start Guide for Populating Mobile Test Devices*, National Institute of Standards and Technology, Report No.:800-202, Mai 2018. DOI: 10.6028/NIST.SP.800-202.

[26] I. O. for Standardization, *ISO/IEC 27037:2012 - Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence*, Zugriff am 07. Oktober 2024. Adresse: https://www.iso.org/standard/44381.html.

[27] D. T. A. Watsen Networks, *RFC 8572 - Secure Zero Touch Provisioning (SZTP)*, Zugriff am 07. Oktober 2024. Adresse: https://datatracker.ietf.org/doc/html/rfc8572.

[28] L. Bortnik und A. Lavrenovs, "Android Dumpsys Analysis to Indicate Driver Distraction," in *Digital Forensics and Cyber Crime*, S. Goel, P. Gladyshev, D. Johnson, M. Pourzandi und S. Majumdar, Hrsg., Cham: Springer International Publishing, 2021, S. 139–163. DOI: 10.1007/978-3-030-68734-2_8.

[29] D. Timko, M. Sharko und Y. Li, "Security Analysis of Wearable Smart Health Devices and Their Companion Apps," in *2024 IEEE Security and Privacy Workshops (SPW)*, 2024, S. 274–280. DOI: 10.1109/SPW63631.2024.00033.

[30] Android, *Android Developers: Android Mobile App Developer Tools*, Zugriff am 05. Oktober 2024. Adresse: https://developer.android.com.

[31] Statista, *Beliebteste Smartphone-Marken in Deutschland*, Zugriff am 08. Oktober 2024. Adresse: https://de-statista-com.ub-proxy.fernuni-hagen.de/prognosen/999729/deutschland-beliebteste-smartphone-marken.

[32] Statista, *Vergleich der Marktanteile von Android und iOS am Absatz von Smartphones in Deutschland von Januar 2012 bis September 2024*, Zugriff am 10. Oktober 2024. Adresse: https://de-statista-com.ub-proxy.fernuni-hagen.de/statistik/daten/studie/256790/umfrage/marktanteile-von-android-und-ios-am-smartphone-absatz-in-deutschland/.

[33] M. S. Mialich, J. M. F. Sicchieri und A. A. J. Junior, "Analysis of body composition: a critical review of the use of bioelectrical impedance analysis," *International Journal of Clinical Nutrition*, S. 1–10, Ausgabe 2(1) 2014.

[34] I. A. Faisal, T. W. Purboyo und A. S. R. Ansori, "A review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture," *J. Eng. Appl. Sci*, S. 826–829, Ausgabe 15(3) 2019.

[35] P. C. Chang und M. S. Wen, "Atrial fibrillation detection using ambulatory smartwatch photoplethysmography and validation with simultaneous holter recording," *erschienen in American Heart Journal*, S. 55–62, Ausgabe 247/2022.

[36] E. Oriwoh, D. Jazani, G. Epiphaniou und P. Sant, "Internet of Things Forensics: Challenges and approaches," in *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2013, S. 608–615. DOI: 10.4108/icst.collaboratecom.2013.254159.

[37] Samsung, *Über Samsung Knox*, Zugriff am 14. Oktober 2024. Adresse: https://www.samsungknox.com/de/about-knox.

[38] A. Atamli-Reineh, R. Borgaonkar, R. A. Balisane, G. Petracca und A. Martin, "Analysis of Trusted Execution Environment usage in Samsung KNOX," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, Ser. SysTEX '16, Trento, Italy: Association for Computing Machinery, 2016. DOI: 10.1145/3007788.3007795.