Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Bereich Elektrotechnik und Informatik

Master-Thesis

Sicherung und Auswertung von forensischen Artefakten aus Smart Home Systemen

Eingereicht am: 29. September 2022
von: Steffen Dietrich

Aufgabenstellung

Die vorliegende Masterthesis soll zunächst ein Überblick zum Stand der Technik gegeben werden, welche Möglichkeiten zum Umgang mit Artefakten aus Smart Home Anwendungen bestehen. Zudem soll eine einheitliche Methode entwickelt werden, die in den Workflow des Erkennungsdienstes integriert werden kann. Der anschließende Output für die Praxis soll ein wissenschaftlich fundierter Workflow für die Vor-Ort-Arbeit an Tatorten und bei Durchsuchungen werden. Die Masterarbeit richtet sich an IT-Forensiker*innen in Ermittlungsbehörden. Nachgelagertes Ziel ist es, den Qualitätsstandard im Umgang mit forensischen Artefakten aus Smart Home Anwendungen zu erhöhen.

Kurzreferat

Da Smart Home Systeme häufig eine Momentaufnahme der Situation an einem bestimmten Ort zu einem Tatzeitpunkt geben können, stellen deren forensische Artefakte eine Unterstützung bei Ermittlungen von Strafverfolgungsbehörden dar. Um diese Artefakte adäquat sammeln und auswerten zu können, müssen sowohl technische als auch rechtliche Grundlagen geschaffen werden.

Ziel der vorliegenden Masterthesis ist die Entwicklung eines Leitfadens für die Sammlung und Auswertung forensischer Artefakte aus Smart Home Systemen. Die Untersuchungsschritte sollen flexibel auf unterschiedlichste Arten von Smart Home Systemen anwendbar sein. Hierbei wurden zunächst unterschiedliche Vorgehensweisen beleuchtet und zu einem allgemein gültigen fünfstufigen Untersuchungsablauf zusammengeführt, bestehend aus: OSINT-, Netzwerk-, App-, Cloud- und Geräteanalyse. Zum besseren Verständnis der Methoden wurden diese in der Untersuchung des Smart Home Systems der Herstellerfirma Hama praktisch angewendet. Es wurden Daten ermittelt, die Rückschlüsse auf das Nutzungsverhalten sowie auf Zeitpunkte von realen Ereignissen zuließen.

Hierzu zählt beispielsweise die Sammlung von Alarmmeldungen eines Türsensors aus dem Speicherabbild eines Smartphones.

Abstract

Since smart home systems can provide inferences about situations in a particular environment at the time of a crime, their forensic artifacts can be used to assist law enforcement agencies in crime investigations. In order to collect and evaluate these artifacts in a qualified manner, both technical and legal foundations must be created.

The aim of this master thesis is to develop a guideline for the collection and evaluation of forensic artifacts from smart home systems. The methods are to be flexibly applicable to the most diverse types of smart home systems. Different approaches were described and combined into a generally applicable five-step investigation procedure, consisting of: OSINT-, network-, app-, cloud- and device-analysis.

For a better understanding, the methods were applied to collect and examine forensic artefacts from the Hama Smart Home ecosystem. The artifacts found allowed conclusions to be drawn about usage behavior as well as real events. This includes, for example, the collection of alarm messages from a door sensor from a forensic image of a smartphone.

Glossar

API (Application Programming Interface)

Mithilfe von APIs bieten Programme Programmierschnittstellen zur Kommunikation mit Drittsoftware. Über APIs können beispielsweise Cloudlösungen einfach in lokale Applikationen eingebunden werden.

Baudrate

Die Baudrate gibt an, wie viele Symbole innerhalb einer Sekunde in einer Kommunikation übertragen werden. Die Baudrate ist nicht zu verwechseln mit der Bitübertraungsrate, da ein Symbol aus mehreren Bit besteht.

Carving

Das Wiederherstellen nicht überschriebener gelöschter Daten mithilfe der Dateiheader bezeichnet man als Carving. So werden selbst Dateien, die vom Dateisystem nicht mehr gefunden werden, wiederhergestellt. Die Metadaten sind jedoch verloren.

DNS-Request

Ein DNS-Request (Domain Name System) ist eine Anfrage zur Auflösung eines Hostnames in eine IP-Adresse. Diese Anfrage wird von einem Computersystem an einen Nameserver gestellt. Nameserver lösen den Domainnamen mithilfe von Datenbanken in eine IP-Adresse auf. Diese IP-Adresse wird im Anschluss als Antwort auf den DNS-Request versendet.

eMMC-Chip

Ein eMMC-Chip (Embedded Multimedia Card) ist ein häufig in Kleinstgeräten eingesetztes Speichersystem, das aus dem eigentlichen Speichersystem in Form eines NAND-Speichers und einem Speichercontroller besteht.

ISM-Band

Die ISM-Bänder (Industrial, Scientific, Medical) sind Funkfrequenzbereiche, in welchen Funkstationen ohne Lizenz betrieben werden dürfen. Hierzu zählt beispielsweise die 2,4 GHz-Frequenz, auf der unter anderem WLAN-Netzwerke betrieben werden.

ISO/IEC 27043

Der Standard ISO/IEC 27043 definiert die Untersuchung von Vorfällen im Bereich der IT-Sicherheit. Hierbei werden unter anderem Vorgaben und Handlungsanweisungen beschrieben, nach denen nach einem IT-Sicherheitsvorfall die forensische Untersuchung eingeleitet und durchgeführt wird.

Logische und physikalische Datensicherung

Die physikalische Sicherung beschreibt eine Sicherung jedes einzelnen physikalischen Bits eines Datenspeichers. Somit werden auch gelöschte Daten und ungenutzte Blöcke des Datenspeichers mit gesichert.

Bei einer logischen Sicherung eines Datenspeichers werden ausschließlich die im Dateisystem vorhandenen Daten gesichert.

OSI-ISO-Schichtenmodell

Das OSI-ISO-Schichtenmodell (Open Source Interconnection) stellt einen Standard zur Netzwerkkommunikation zwischen Computersystemen dar. In sieben Schichten werden unterschiedliche Ebenen von Verbindungen beschrieben, beginnend mit der physikalischen Bitübertragung bis zur Anwendung.

Parser

Parsing beschreibt das Umstrukturieren von Rohdaten in ein lesbares oder verarbeitbares Format. Beispielsweise können so codierte Einträge aus Datenbanken ausgelesen und forensisch aufbereitet werden.

Penetrationstest

Bei einem Penetrationstest wird durch gezielte Angriffsversuche die Sicherheit von Computersystemen überprüft.

Portscan

Bei einem Portscan wird ein System in einem Netzwerk auf offene Schnittstellen, sogenannte Ports untersucht. Hierbei kann nicht nur ermittelt werden, ob ein Port geöffnet ist, sondern auch ob und welcher Netzwerkservice hinter dem Port läuft.

Smart Home Ökosystem

Als Smart Home Ökosystem bezeichnet man die gemeinsame Verwendung mehrerer smarter Geräte. Meist werden die einzelnen Komponenten im selben Netzwerk betrieben. Die Geräte können entweder eigenständig agieren, oder per App oder Bridge miteinander verknüpft sein. So lassen sich Automatismen erstellen, bei denen die Sensoren einzelner Geräte eine Aktorfunktion anderer Geräte auslösen.

Swapfiles

In Swapfiles wird ein temporär nicht benötigter Inhalt des Arbeitsspeichers in das Dateisystem ausgelagert. So kann der Arbeitsspeicher vergrößert werden.

1 Inhalt

1. Einleitung

Mit der Verbreitung von Computersystemen hat sich auch die Forensik in diesem Bereich entwickelt. Bereits heute sind Beweismittel aus informationstechnischen Geräten eine tragende Säule der Strafermittlung (Hahn, 2017). Die IT-Forensik beschäftigt sich mit der Sammlung, Analyse und Präsentation von digitalen Artefakten, die einen kriminellen Tathergang nachvollziehbar machen. Als digitalforensische Artefakte werden Daten bezeichnet, die Spuren der Nutzung von Informationstechnologie in Datenspeichern abbilden. Digitalforensische Artefakte können Rückschlüsse auf reale Ereignisse bilden, beispielsweise in Form von Bildern.

Das Aufkommen von Smartphones hat den Bereich der klassischen IT-Forensik um den Teilbereich der Mobilforensik erweitert. Die vielseitigen Nutzungsmöglichkeiten von Smartphones als Kommunikationsgerät, Kamera, Notizspeicher, Zugang zum Internet etc., machen sie zu einer bedeutsamen Quelle digitalforensischer Artefakte.

Nach klassischen Computern und Smartphones wurde in den vergangenen Jahren die Rolle des Internet of Things (IoT) immer prägender für den Alltag. Zunächst wurden vernetzte eingebettete Systeme nur im Zuge der Industrie 4.0 eingesetzt (Georgiev & Schlögl, 2018). Mittlerweile ist auch im Privatsektor die Nutzung von smarten Systemen allgegenwärtig. Ihre Aufgabe ist es, Nutzer*innen, möglichst komfortabel und individuell steuerbar, Funktionen von Haushaltsgegenständen zur Verfügung zu stellen. Bedingt wurde die schnelle Verbreitung durch die geringen Kosten für eine smarte Nachrüstung (Zawoad & Hasan, 2015). Smart Home Systeme werden im Haushalt in den Bereichen Entertainment, Haushaltsgeräte, Sicherheit, Energie, Klima sowie Beleuchtung eingesetzt. Forensische Artefakte aus diesen vernetzten Systemen stellen IT-Forensiker*innen vor neue Herausforderungen.

1.1 Motivation

In einer Umfrageuntersuchung des Institutes Bitkom Research aus dem Jahr 2021 gaben 41% der Befragten an, Smart Home Anwendungen in ihrem Haushalt zu nutzen. 2018 lag dieser Anteil noch bei 26%. In der Gruppe der 30- bis 49-Jährigen lag der Anteil bereits bei mehr als der Hälfte der Befragten. In den kommenden Jahren wird die Nutzung von Smart Home Anwendungen voraussichtlich weiter zunehmen (Klöß & Gentemann, 2020). Strafverfolgungsbehörden müssen diesem Trend folgen, um Ermittlungsansätze in diesem neuen Sektor adäquat nutzen zu können.

Durch die Verwendung smarter Alltagsgegenstände und die daraus resultierende Häufigkeit der Interaktionen zwischen Mensch und Maschine entsteht ein umfassendes Abbild des Nutzerverhaltens (Awasthi, Read, Xynos, & Sutherland, 2018). Dieses kann wiederum entscheidende Erkenntnisse über Nutzergewohnheiten und reale Ereignisse geben (Losavio et al., 2018). So war es Azhar & Bate möglich, den Schlafrhythmus von Nutzer*innen mittels WLAN-SCAN zu untersuchen (Hannan Bin Azhar & Bate, 2019). Für die Herstellerfirmen sind Rückschlüsse auf ein Nutzungsverhalten von großem Wert, da die Daten zur Verbesserung des Service und zur Anzeige personalisierter Werbung genutzt werden können. Diese Daten werden entweder selbst genutzt oder als detaillierte Nutzungsprofile gewinnbringend verkauft. Ziel ist es, das Nutzerverhalten zu erkennen, bevor es auftritt (Georgiev & Schlögl, 2018).

Entsprechend ist das Interesse an Artefakten aus Smart Home Systemen auch für Strafverfolgungsbehörden hoch. Chung et. al. bezeichnen Smart Home Systeme als „Always on Human Life Blackbox“, die eine wichtige Unterstützung für die Verbrechensaufklärung bieten können (Chung, Park, & Lee, 2017). Neben Nutzerverhalten sind auch die von Sensoren aufgezeichneten Änderungen von Umgebungszuständen von Bedeutung. Temperaturänderungen können Aufschluss über geöffnete Fenster oder Türen geben, smarte Stromzähler haben möglicherweise den Einschaltzustand eines Haushaltsgerätes aufgezeichnet, Sicherheitseinrichtungen den Zeitpunkt eines Alarms (Sevida & Casey, 2019). Die so gewonnenen Erkenntnisse geben bereits Aufschluss über reale Ereignisse. Kombiniert man die Daten miteinander und mit den bisherigen

Ermittlungsansätzen, so ergibt sich ein detaillierteres Bild. Auf diese Weise werden die Daten nicht nur als mögliche Indizien wichtig, sondern beispielsweise auch für die anschließende Vernehmungstaktik.

Um einen Qualitätsstandard bei der Sammlung und Auswertung von Artefakten aus Smart Home Systemen zu gewährleisten, werden in Zukunft digitale Tatortforensiker*innen als Teil des klassischen Erkennungsdienstes eingesetzt (Stoyanova et al., 2020). Diese Teams kommen neben Tatorten auch bei Durchsuchungen zum Einsatz.

1.2 Problemstellungen der Smart Home Forensik

Etablierte Workflows und Software aus der klassischen IT-Forensik geraten im Bereich des IoT, im Speziellen bei der Smart Home Forensik an ihre Grenzen. Hinzu kommt, dass bisher wenige Tools entwickelt wurden, die Smart Home Systeme auswerten, bzw. auf diese zugreifen können (Al-Masri et al., 2018). Ein Grund hierfür ist die Heterogenität der Systeme unterschiedlicher Herstellerfirmen. Die Art und das Einsatzgebiet einzelner Geräte diversifiziert die Untersuchungsmethoden weiter (Hutchinson et al., 2020). Hinzu kommt die Verwendung unterschiedlicher Protokolle zur Kommunikation. Bisher konnte sich kein Standard zum Umgang mit forensischen Daten aus IoT-Systemen durchsetzen (Stoyanova et al.,2020). Die Verwertbarkeit der durchgeführten Untersuchungen vor Gericht wird daher in Zweifel gezogen. Zur Validierung von forensischer Software im Bereich IoT fehlen wissenschaftliche Methoden, die gerichtlichen Prozessen standhalten (Arshad et al., 2018). Selbst die klassische Digitalforensik wird wegen ihrer ausschließlich empirischen Gültigkeitsüberprüfung zuweilen als Pseudowissenschaft betitelt. Digitalforensiker*innen können vor Gericht nur überzeugend auftreten, wenn sie die Funktionsweise, Kommunikationswege und Datenstruktur des untersuchten Systems lückenlos darlegen können (Arshad et al., 2018). Da dieses detaillierte Verständnis für jedes Gerät neu entwickelt werden muss, ist der Zeitaufwand verglichen mit der Auswertung von Computern oder Smartphones hoch. Zudem

fehlen Schulungsmethoden und Labore zur Aus- und Weiterbildung von Digitalforensiker*innen im Smart Home Sektor (Wu et al., 2019).

Am Tatort stellt sich weiterhin die Frage, wie smarte IT detektiert werden kann. Durch ihre geringe Größe und ihre Ähnlichkeit zu üblichen Haushaltsgeräten ist eine aufwändige Suche notwendig. Die Kurzlebigkeit digitaler Artefakte in den Geräten fügt der Arbeit am Tatort und bei Durchsuchungen eine zeitkritische Komponente hinzu (Hutchinson et al., 2020). Zunehmend rückt der Schutz persönlicher Daten in den Fokus des IT-Sektors. Der Einsatz von Ende-zu-Ende-verschlüsseltem Datenaustausch und verschlüsselter Firmware erschwert die Systemanalyse (Gupta, 2019). Smarte Haushaltssysteme speichern ihre erstellten Daten meist nicht im Gerätespeicher, sondern übermitteln sie an Cloudsysteme (Perumal et al., 2015). Hier ergeben sich weitere Herausforderungen, denen das Kapitel Cloudanalyse gewidmet ist.

1.3 Forschungsfrage und Zielsetzung

Zur Auswertung von Artefakten aus Smart Home Systemen wurden bisher verschiedene Ansätze wissenschaftlich beschrieben, die im Kapitel Aktueller Forschungsstand genauer beleuchtet werden. Die geleistete Forschungsarbeit versucht, der Heterogenität der Systeme gerecht zu werden. Hierfür wurde entweder detailliert an einzelnen Komponenten bestimmter Herstellerfirmen gearbeitet oder hersteller- und systemübergreifend mit geringerem Detailgrad. Beide Methoden haben ihre Vor- und Nachteile.

Das Ziel dieser Masterarbeit ist es, die bisherigen Methoden zu beleuchten, ihre Vor- und Nachteile zu erörtern und einen umfassenden Überblick über die Best Practices zu geben. Die Kenntnis dieser Methoden erlaubt eine signifikante Zeitersparnis und Qualitätsverbesserung, da sie den Analysierenden eine fundierte Entscheidungsbasis für die Wahl der Methode zur Verfügung stellt. Die beschriebenen Vorgehensweisen werden anschließend anhand des Smart Home Systems des deutschen Herstellers Hama erläutert.

1.4 Abgrenzung des Forschungsgebiets

Grundsätzlich unterscheidet die IoT-Forensik zwischen „IoT as a target“, „IoT as a tool“ und „IoT as a witness“ (Salamh, 2020). Hierbei wird differenziert, ob IoT-Geräte als Ziele von Cyberattacken, als Werkzeuge für Cyberattacken oder als digitale Zeugen von Straftaten betrachtet werden. Die vorliegende Arbeit beschäftigt sich ausschließlich mit „IoT as a witness“. Weiterhin wird ausschließlich der Bereich Smart Home betrachtet. Hiervon ausgenommen sind Entertainmentsysteme sowie die in einigen Quellen als Smart Home Geräte angesehenen Wearables (z.B. Smartwatches). Auch die Entschlüsselung verschlüsselter Firmware wird kein Bestandteil der Abschlussarbeit sein.

2 Aktueller Forschungsstand

Das folgende Kapitel befasst sich mit der bisher geleisteten Forschungsarbeit auf dem Gebiet der Smart Home Forensik. Die wissenschaftlichen Veröffentlichungen werden kategorisiert und in einen Kontext zur Forschungsfrage der vorliegenden Masterthesis gesetzt.

Grundsätzlich lassen sich bisherige Forschungsarbeiten in verschiedene Kategorien gliedern. Zum einen werden einzelne Smart Home Ökosysteme untersucht, es werden detailliert die jeweiligen forensischen Artefakte und Methoden zu deren Auswertung beschrieben (z.B. Awasthi et al., 2018). Zum anderen versuchen weitere wissenschaftliche Arbeiten, wie Salamh (2020), forensische Modelle zu schaffen, die allgemeingültig und herstellerübergreifend anwendbar sind. Ein weiterer Ansatz baut realistische Netzwerke aus unterschiedlichen Ökosystemen auf und versucht, die angewendeten Modelle für unterschiedliche Geräte zu validieren (z.B. Kim et al., 2020).

Die Beschreibung einzelner Herstellersysteme bietet einen hohen Detailgrad bei der Sammlung und Auswertung von forensischen Artefakten. Einzelne Datenquellen und Speicherorte werden ebenso dargestellt wie Methoden zur Auswertung. Problematisch stellen sich neue Versionen von Geräten oder deren Betriebssystemen dar. Eine Änderung kann zur Verschiebung von Artefaktquellen führen. Dies erschwert die Auswertung. So gelten die Forschungsergebnisse ausschließlich für das vorliegende Gerät in der beschriebenen Version.

Awasthi et al. (2018) bieten in ihrer Abhandlung über die forensische Analyse des Almond Hubs von einem ganzheitlichen Leitfaden, um Artefakte aus dem System nicht nur zu sammeln, sondern die spätere Präsentation der Ergebnisse mit einem Parser vorzubereiten. Der Almond Hub ist ein System, welches als Brücke zwischen Smart Home Komponenten und einem WLAN-Router dient. Als problematisch erweist sich bei Parsern grundsätzlich, dass diese mit jeder Version des Betriebs- und Dateisystems umgeschrieben werden müssen. Trotzdem vereinfacht ein fundiertes Parsing die spätere Auswertung.

Auch Sevida und Casey (2019) befassen sich mit der forensischen Auswertung mittels Parser. Nach der ausführlichen Beschreibung des Laboraufbaus wurde versucht, ein Logging zu implementieren, um Mitteilungen über den Gerätestatus zu sichern. Hierfür wurde der Netzwerkverkehr analysiert und geparst. Die Arbeit befasst sich ausschließlich mit IoT-Geräten als Ziel von Cyberattacken. Eine Umsetzung für „IoT as a witness“ gestaltet sich schwierig, da die Loggingfunktionen vor der jeweiligen Tat implementiert werden müssten. Die Notwendigkeit solcher Funktionen im Privatsektor ist nicht gegeben.

Chung et al. (2017) stellen das Framework „CIFT“ (Cloud-based IoT Forensic Toolkit) vor. Mithilfe dieses Tools kann das „Amazon Alexa“ Ökosystem forensisch ausgewertet werden. Hierbei werden zwei Ansätze verfolgt, einerseits die Untersuchung des Gerätes selbst, andererseits die Auswertung der in Cloudsystemen gespeicherten Daten. Zur Analyse der Clouddaten wird eine inoffizielle API-Schnittstelle für „Alexa“ genutzt. Zudem konnten verwertbare Artefakte im Browsercache von Chrome gefunden werden. Es wurden Benutzeraccounts, WLAN-Einstellungen und sogar Sprachmitschnitte heruntergeladen, Artefakte und ihre Speicherorte werden ausführlich beschrieben. Die Verwendung einer inoffiziellen API vor Gericht ist problematisch, da das Unternehmen Amazon keine offiziellen Schnittstellen bekannt gibt. Somit kann die Verwendung inoffizieller API leicht angefochten werden.

Das Paper von Li et al. (2015) beschäftigt sich ebenfalls mit dem Amazon Ökosystem. Untersucht wird der Home Assistent „Amazon Echo“. Der beschriebene Untersuchungsprozess orientiert sich am Vorgehen in der klassischen Digitalforensik. Die Autoren unterscheiden bereits zu Beginn einer Untersuchung zwischen „IoT as a target“ und „IoT as a witness“. Zunächst erfolgt eine Identifizierung des Gerätes mit einer anschließenden Sicherung volatiler Daten zu deren Erhaltung. Die gesicherten Dateien werden gesammelt und analysiert. Auch der Arbeitsspeicher und Swapfiles werden hierbei als Artefaktquelle konserviert. Abschließend werden die Funde präsentiert. Die ganzheitlichen Prozesse zur Analyse von Smart Home Systemen bestehend aus Cloudanalyse, Appanalyse, Geräteanalyse und Netzwerkanalyse werden in

mehreren weiteren Veröffentlichungen angewendet. Sie können entweder parallel oder aufeinanderfolgend abgearbeitet werden. Die Prozesse werden auch in dieser Thesis aufgegriffen und im Kapitel Methoden beschrieben.

Hersteller- und geräteunabhängige Workflows können flexibler und breiter angewendet werden, dafür sinkt der Detailgrad der Untersuchung.

Das von Kebande und Ray (2016) vorgestellte DFIF (Digital Forensic Investigation Framework for Internet of Things) besteht aus drei Modulen, die grundsätzlich auf jedes IoT-System angewendet werden können. Im ersten Schritt wird eine forensische Planung durchgeführt, die z.B. die Identifikation möglicher Artefakte und deren Asservierung beinhaltet. Der zweite Schritt besteht aus den bereits in Li et al. (2015) beschriebenen Prozessen Cloudforensik, Netzwerkforensik und Geräteforensik. Abschließend wird ein reaktiver Prozess durchgeführt, in welchem die Asservate gesammelt und untersucht werden. Das Modell beschreibt einen umfassenden Workflow für die forensische Analyse von IoT-Geräten. Die Prozesse müssen jedoch aufwändig für jedes Gerät neu durchgeführt werden.

Salamh (2020) präsentiert das allgemein anwendbare FAHAD-Modell (Forensic Analysis of Home Automation Devices). Validiert wird dieses Modell an der „Kasa Smart Light Bulb“, einer smarten LED-Birne und an der „Eufy Floodlight Camera“, einer smarten IP-Kamera. Das Modell beschäftigt sich mit der gerätenahen Untersuchung von Smart Home Systemen. Die Chip-Off-Methode zum Auslesen der eMMC-Chips wird jedoch nicht genauer beschrieben. Das forensische Image des Speicherinhaltes wird mittels der Forensiktools Autopsy (autopsy.com, 2022) und Binwalk untersucht. Zudem bietet das Paper eine Entscheidungshilfe, inwiefern klassische forensische Methoden für Smart Home Analysen angewendet werden können.

Das von Sadineni et al. (2019) entwickelte Modell orientiert sich am Standard ISO/IEC 27043. Dieser beschreibt Vorfallsreaktionspläne für IT-Sicherheitsvorfälle. Unter anderem wird die mehrstufige forensische Untersuchung nach einem Vorfall dargestellt. Auch hier beginnt der Ablauf nicht mit der Vorfallsanalyse, sondern mit einer proaktiven Vorbereitung. Die

involvierten Geräte werden dokumentiert, mögliche Vorfälle definiert und die Datensammlung vorbereitet. Wird ein Sicherheitsvorfall erkannt, so wird der forensische Prozess initiiert und die Untersuchung eingeleitet. Die letzte Phase beschreibt die forensische Untersuchung, bestehend aus Beweissicherung, -analyse, Vorfallsrekonstruktion und -präsentation.

Das Modell legt keinen Fokus auf „IoT as a witness“. Die Vorbereitungsphase wird benötigt, somit kann der modifizierte Standard ISO/IEC 27043 kaum auf private Smart Home Systeme angewendet werden. Dennoch finden sich wertvolle Impulse für die Strafermittlung. Beispielsweise wird im Schritt der Untersuchungsvorbereitung bereits die Herstellerfirmen der Geräte kontaktiert und um Unterstützung ersucht.

Einige Autoren haben einen umfassenderen Ansatz gewählt. Hier wird eine Auswahl von Geräten und Hubs unterschiedlicher Ökosysteme in einem gemeinsamen Labornetzwerk untersucht.

Kim et al. (2020) beschreiben die Sammlung und Analyse von forensischen Daten aus dem Google Nest Hub als Basis für Samsung Smart Things Geräte und eine Kasa cam (IP-Kamera). Ausführlich werden Artefakte aus dem Google "My Activity" Webinterface dargestellt. Der hier präsentierte Arbeitsablauf ist ausschließlich auf diese Geräte angepasst. Die beschriebenen Datenquellen sind nicht allgemeingültig. Dennoch kann die Kenntnis dieser Quellen Ansätze für andere Geräteuntersuchungen liefern.

Ebenfalls mit Amazon Echo beschäftigen sich Hannan Bin Azhar und Bate (2019). Die untersuchten Artefakte werden in die Kategorien Netzwerk-, Cloud- und Geräteartefakte unterteilt. Zur besseren Übersicht werden Netzwerkartefakte in Cloud, Maschine zu Maschine und Mensch zu Maschine gegliedert. Der Untersuchungsprozess startet mit einer passiven Beobachtung der Geräte im Netzwerk sowie ein Portscan. Es folgen Cloud- und Geräteuntersuchung. Es wird ein artefaktzentrierter Ansatz gewählt, der allgemeine Anwendung finden kann. Die Untersuchung findet nicht im Hinblick auf ein bestimmtes Gerät statt, sondern mit Fokus auf die zu findenden Artefakte.

Auch Hutchinson et al. (2020) analysieren ein Netzwerk aus Komponenten verschiedener Herstellerfirmen. Das System besteht aus einem Amazon Fire TV Stick, einem Google Nest Hub Max, einem Smartlock und einer Smartbell von August sowie einer TP-Link Smart Bulb. In dieser Forschungsarbeit werden Artefakte ebenfalls in verschiedene Kategorien unterteilt: Sensorkommunikation, Kommunikation innerhalb des Netzwerkes und Kommunikation außerhalb des Netzwerkes (z.B. zur Cloud). Ein vierstufiger Untersuchungsprozess wird angewendet, beginnend mit der Initialisierung. Hier wird sich zunächst ein grundlegendes Verständnis über die Funktionen des Systems und das Verhalten im Netzwerk verschafft. Im Folgenden werden die Datenquellen identifiziert und anschließend die dort gespeicherten Daten extrahiert. Im letzten Schritt werden die gefundenen Artefakte hinsichtlich ihrer Relevanz untersucht. Auch wenn sich das Paper auf "IoT as a target" fokussiert, können die gewählten Unterteilungen und Arbeitsschritte für "IoT as a witness" genutzt werden.

Tekeoğlu und Tosun (2015) fokussieren sich auf IP-Kameras. Der hier vorgestellte Arbeitsablauf legt den Schwerpunkt auf gerätespezifische Analyse, um der Andersartigkeit unterschiedlicher Gerätearten gerecht zu werden. So sind z.B. bei IP-Kameras andere Artefakte zu finden als bei einfachen Sensoren, da durch die Kameras Multimediadateien erzeugt werden. Die vorgestellten Methoden werden am Beispiel der Belkin NetCam IP-Kamera erläutert.

Mit der Automation von forensischen Auswertungen beschäftigen sich Dorai et al. (2018). Die Forschenden stellen ein Tool zur automatischen Akquise, Auswertung und Berichterstellung für das Google Nest Ökosystem vor. Das Forensic Evidence Acquisition and Analysis System (FEAAS) beachtet bei der Auswertung besonders die unterschiedlich langen Halbwertszeiten forensischer Artefakte. Zur Überprüfung der Ergebnisse wurden unterschiedliche Szenarien erstellt, deren digitale Spuren anschließend ausgewertet wurden. Forensisch relevante Daten wurden größtenteils aus unverschlüsselten iPhone-Backups gewonnen. Die darin befindlichen Datenbanken wurden automatisiert geparst und aus den so aufbereiteten Ergebnissen ein Bericht mit Angabe von Zeitstempel und Ereignissen generiert.

3 Methoden

3.1 Verwendete Software

3.1.1 aircrack-ng Suite (Version 1.6)

Die aircrack-ng Suite ist eine Open Source Toolsammlung, deren Hauptanwendungsgebiet auf Sicherheitsanalysen und Penetrationstests von WLAN-Netzwerken liegt. Die in dieser Arbeit verwendeten Tools sind airmon-ng und airodump-ng, beide Tools werden zur WLAN-Aufklärung eingesetzt.

Airmon-ng versetzt die Netzwerkkarten in den sogenannten Monitor Mode. In dieser Einstellung werden alle WLAN-Pakete in der Umgebung abgefangen und an das Betriebssystem weitergeleitet.

Das zweite verwendete Tool ist airodump-ng. Dieses Tool zeichnet den im Monitor-Mode abgefangenen WLAN-Verkehr auf. Zudem werden statistische Daten zum mitgeschnittenen WLAN-Verkehr live ausgebeben. Beide Tools werden über die Kommandozeile gesteuert (aircrack-ng.org, 2022).

3.1.2 Android Studio (Version 2021.2.1 Patch 1)

Die Android-IDE Android Studio wird zur Entwicklung von Android-Applikationen genutzt. Unter anderem bietet die Plattform Möglichkeiten zum Debuggen von apk-Dateien. Diese Funktion kann in forensischen Analysen hilfreich sein, wenn einzelne Funktionen und Dateien analysiert werden (developer.android.com, 2022).

3.1.3 Cellebrite UFED (Version 7.58.0.172)

Das Programm UFED (Universal Forensics Extraction Device) des Softwareentwicklers Cellebrite wird angewendet, um Datenspeicherabbilder aus mobilen Geräten (meist Smartphones) zu erstellen. Cellebrite nutzt dabei teilweise Sicherheitslücken der untersuchten Geräte aus. So kann in vielen Fällen auch die Bildschirmsperre umgangen werden. Zudem bietet Cellebrite UFED ein

Tool zur Auswertung der erstellten Speicherabbilder an (cellebrite.com, 2022).

3.1.4 Censys.io

Die IT-Sicherheitsberatungsfirma Censys stellt unter search.censys.io eine Suchmaschine für Netzwerkservices im Internet zur Verfügung. Mittlerweile enthält die Datenbank 215.000.000 IPv4- und über 35.000 IPv6-Hosts. Zu den Hosts werden Informationen zu offenen Ports, laufenden Services, verwendeten Zertifikaten und Betreibern gesammelt. Diese Informationen können in forensischen Netzwerkanalysen genutzt werden (search.censys.io, 2022).

3.1.5 Dex2jar (Version 2.1)

Das Tool dex2jar wird verwendet, um Dalvik Executable Dateien (.dex) in Java Archive (.jar) umzuwandeln (github.com/pxb1988, 2021).

3.1.6 FTK-Imager (Version 4.3.1.1)

Mit der forensischen Software FTK-Imager der Firma AccessData werden forensische Images von Datenträgern erstellt. Ebenfalls bietet es eine Writeblocker-Funktion, mit der forensische Images geladen und analysiert werden können, ohne dass bei diesem Zugriff Daten überschrieben werden (accessdata.com, 2022).

3.1.7 Jd-gui (Version 1.6.6)

Mithilfe dieses Java-Decompilers lassen sich .class-Dateien grafisch darstellen, analysieren und verändern. Das Tool kann auf Windows, Linux und MAC-OS genutzt werden. Zudem gibt es Plugins für IDE-Plattformen wie z.B. Eclipse (java-decompiler.github.io, 2022).

3.1.8 Killerbee (Version 3.0.0-beta)

Killerbee ist eine Toolsammlung, die in Sicherheitsanalysen von ZigBee-Netzwerken Anwendung findet. Mithilfe einer ZigBee-Sniffer-Hardware können Netzwerkpakete abgefangen und analysiert werden. Zudem ist es möglich, aktiv

ZigBee-Pakete in eine Verbindung einzuspeisen. Das Tool zbwireshark wird in der vorliegenden Arbeit im Bereich Funkaufklärung genutzt. Es stellt ZigBee-Netzwerkverkehr in Wireshark dar (github.com/riverloopsec, 2022).

3.1.9 Kismet (Version 2022-08-R1)

Kismet ist ein Open Source Tool zur Aufklärung drahtloser Netzwerke (z.B. WLAN). Das Programm kann auf unterschiedlichen Plattformen betrieben werden. Mit Kismet lässt sich der Funkverkehr in der Umgebung mitschneiden und in Echtzeit darstellen. Die grafische Oberfläche ist eine Webanwendung, die mit dem Browser geöffnet wird (kismetwireless.net, 2022).

3.1.10 Magnet Axiom (Version 5.7.0.27176)

Die forensische Software Axiom der Herstellerfirma Magnet Forensics wird zur Erstellung und Untersuchung forensischer Datenspeicherabbilder verwendet. Axiom kann sowohl Festplatten aus Computersystemen, als auch mobile Geräte als Quelle verarbeiten (magnetforensics.com, 2022).

3.1.11 Nmap (Version 7.80)

Nmap (Network Mapper) ist ein Programm zur Analyse von Netzwerkports und -Services. Besonders in seiner Funktion als Portscanner findet Nmap breite Anwendung. Neben der Suche nach offenen Ports kann Nmap auch die dahinterliegenden Services ermitteln. Zudem verfügt es über eine Script-Engine (nse), mithilfe derer die Nutzung angepasster Skripte zur Serviceanalyse ermöglicht wird (nmap.org, 2022).

3.1.12 Sleuth Kit (Version 4.10.1)

The Sleuth Kit (TSK) ist eine Open Source Toolsammlung für forensische Analysen von Datenträgerabbildern. Die Tools bieten Informationen zu Partitionen, Dateisystemen und einzelnen Dateien. In dieser Thesis wird das Tool fls verwendet, welches Dateien in vorhandenen Dateisystemen analysiert und darstellt (wiki.sleuthkit.org, 2015).

3.1.13 Wireshark (Version 3.6.2-2)

Wireshark ist ein Netzwerkanalyseprogramm, mit dem sich alle gängigen Protokolle untersuchen lassen. Die Software ist dazu in der Lage Netzwerkverkehr an einer Schnittstelle mitzuschneiden, grafisch darzustellen und zu speichern. Die Netzwerkpakete aus den Mitschnitten können einzeln betrachtet werden. Zudem verfügt Wireshark über umfangreiche Funktionen zum Filtern von Paketen. So ist eine detaillierte Analyse des Netzwerkverkehrs möglich (wireshark.org, 2022).

3.2 Auffinden von Smart Home Geräten

Smart Home Geräte an einem Tatort oder während einer Durchsuchung aufzufinden und gleichzeitig die spurenschonende Arbeitsweise des klassischen Erkennungsdienstes anzuwenden ist keine triviale Aufgabe für IT-Forensiker*innen (Hahn, 2017). Hinzukommt, dass smarte Geräte häufig sehr klein oder schwer von normalen Haushaltsgegenständen zu unterscheiden sind (Stoyanova et al., 2020). In diesem Kapitel werden Vorgehensweisen zur digitalen Aufklärung am Zielobjekt beschrieben. Zunächst werden Methoden zur Funkaufklärung beschrieben. Im Anschluss werden verschiedene weitere Möglichkeiten zum Auffinden von Systemen beleuchtet.

3.2.1 Aufklärung von Funkprotokollen

3.2.1.1 WLAN

Um der Anforderung gerecht zu werden, Ethernet-Pakete (Standard IEEE 802.2) per Funk übermitteln zu können, wurde 1997 der Standard IEEE 802.11 entwickelt. Seit 1999 zertifiziert die WiFi-Alliance WLAN-Geräte (Wireless LAN) WLAN basiert auf den Standards IEEE 802.11 a, b, g und n, welche sich hinsichtlich ihrer Datenübertragungsrate und Frequenzbändern unterscheiden (Gessler & Krause, 2015). Die Standards beschreiben die Datenübertragung innerhalb der physikalischen sowie der Sicherungsschicht des ISO/OSI-Modells (tutorial-reports.com, 2022).

Die WLAN-Standards wurden für eine hohe Datenübertragung ausgelegt, hierbei wurde kein Fokus auf niedrigen Energieverbrauch gelegt, weshalb WLAN meist in IoT-Geräten mit dauerhafter Spannungsversorgung oder in Gateways für energiesparendere Funkprotokolle eingesetzt wird (Bertko & Weber, 2017).

Im Juli 2021 wurde der Standard IEEE 802.11ax von der Bundesnetzagentur in Deutschland freigegeben. Das als WLAN 6 bezeichnete Funkprotokoll erreicht bis zu 11 GBit/s und kann sowohl auf den ISM-Bändern der 2,4 GHz als auch auf der 5 GHz-Frequenz betrieben werden (bundesnetzagentur.de, 2021).

Ein WLAN-Netzwerk kann in zwei verschiedenen Modi betrieben werden. Werden zwei Geräte direkt miteinander verbunden, so spricht man vom Ad-Hoc-Mode (tutorial-reports.com, 2022). Dieser kann beispielsweise genutzt werden, um appgesteuerte Drohnen zu kontrollieren. Die Verbindung wird dann zwischen der Drohne und einem Smartphone, das als Fernbedienung dient, aufgebaut. Im Infrastruktur-Modus betrieben wird zum Aufbau eines Netzwerkes ein Access-Point benötigt. Dieser Access-Point fungiert als Zentrale und übernimmt die Steuerung der Netzwerkverbindung angeschlossener Endgeräte (Gessler & Krause, 2015).

Abgesichert wird WLAN heute standardmäßig mit der WPA2-Verschlüsselung. Ein unberechtigtes Mitlesen von Klartextpaketen over-the-air wird auf diese Weise vermieden (tutorial-reports.com, 2022).

Im Smart Home Bereich wird WLAN als Funkstandard vor allem im Infrastrukturmodus eingesetzt. So können mehrere Geräte über einen Access-Point (meistens ein WLAN-Router) sowohl untereinander als auch mit Steuerungsservern und -applikationen kommunizieren.

WLAN kann demnach auch zum Auffinden von Smart Home Geräten verwendet werden. Hierbei wird über die Signalstärke der einzelnen Netzwerkgeräte deren Position lokalisiert. Die Signalstärke wird mithilfe des Leistungspegels angegeben. Dieser definiert sich als logarithmisches Verhältnis zwischen Sendeleistung und Empfangsleistung (Gessler & Krause, 2015).

$$L_p = 10 \lg\left(\frac{P}{P_0}\right) [dB]$$

P ist hierbei die Empfangsleistung, P_0 entspricht der Sendeleistung. Der Leistungspegel wird in Dezibel (dB) angegeben. Dezibel stellt das Verhältnis einer Leistung zu einer Referenzleistung dar.

Die Signalstärke der WLAN-Sendestationen in der Umgebung wird mithilfe

spezieller Aufklärungstools ermittelt. Hierzu wird auf einem mobilen Untersuchungsrechner am Tatort der WLAN-Chip im sogenannten Monitor Mode betrieben, der den gesamten empfangenen Netzwerkverkehr an die Applikationsebene weiterleitet (aircrack-ng.org, 2022). Trotz WPA2-Verschlüsselung kann neben der Signalstärke auch die Art des Gerätes (Access Point oder Client) aus versendeten Paketen ausgewertet werden. Durch Access-Points werden sogenannte Beacon-Frames versendet. Die Aufgabe dieser WLAN-Pakete ist es, WLAN-Geräten im Umfeld die für eine Verbindung benötigten Parameter des Access-Points mitzuteilen.

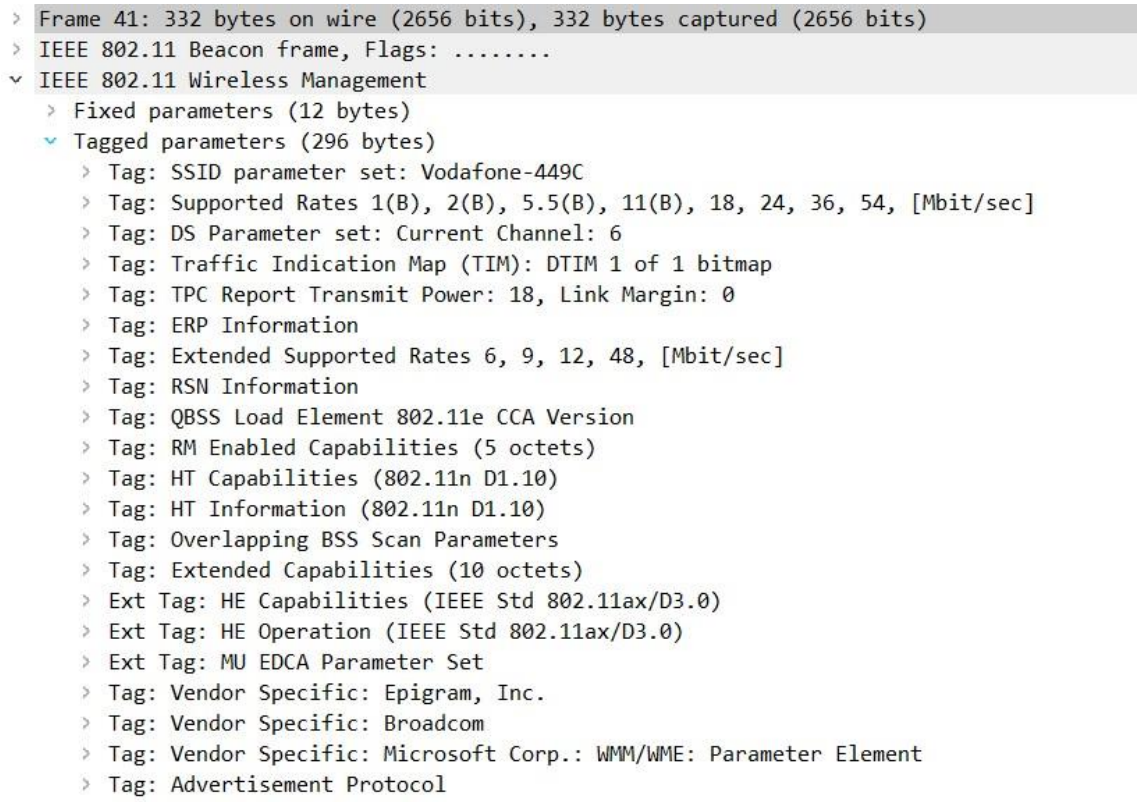


Abbildung 1 – Screenshot: Darstellung eines IEEE 802.11 Beacon frame in Wireshark

Abbildung 1 zeigt einen Beacon-Frame mit den übermittelten Parametern. Beacons werden im Klartext versendet, damit die Informationen allen potenziellen Clients zur Verfügung stehen. Angeschlossene Geräte werden über versendete Datenpakete ermittelt.

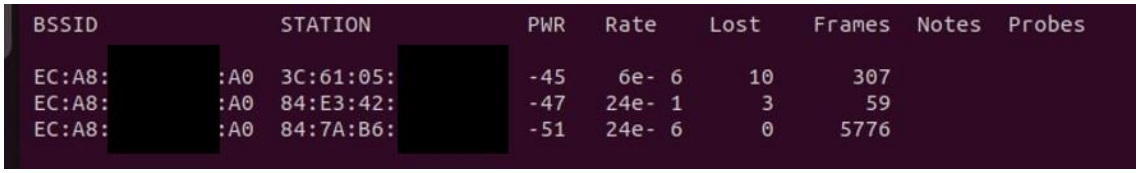


Abbildung 2 – Screenshot: Ausgabe von airodump-ng

Das im Folgenden beschriebene Tool Airodump-ng stellt wie in Abbildung 2 dargestellt diese Geräte in Zusammenhang zu den jeweiligen Accesspoints dar. Die Signalstärke wird ebenfalls angegeben. Um die Aufklärung zu vereinfachen können zwei Untersuchungsrechner, beispielsweise Laptops, mit gleichen WLAN-Chips betrieben werden. Die Positionsbestimmung kann dann mittels Differenz der Signalstärke erfolgen.

Während der WLAN-Aufklärung gibt es einige Aspekte zu beachten.

Um den Störfunk auf einzelnen Frequenzen zu vermindern, können WLAN-Hotspots auf verschiedenen WLAN-Channels arbeiten. Der verwendete Channel wird einem Client zuvor im Beacon-Frame mitgeteilt. Als WLAN-Channels werden überlappungsfreie Frequenzbereiche bezeichnet. Im 2,4 GHz-Spektrum liegen die Kanäle 1-14 (Schemberg et al., 2019). Seit Einführung des Standards IEEE 802.11a senden WLAN-Stationen auch auf 19 Kanälen des 5 GHz-Spektrums. Untersuchungsrechner müssen der Forderung gerecht werden, auf diesen Kanälen den Netzwerkverkehr abzuhören. Zudem sind Heimrouter dazu in der Lage, zur gleichen Zeit im 2,4 und im 5 GHz-Spektrum Hotspots zur Verfügung zu stellen (Gessler & Krause, 2015). Diese unterschiedlichen WLAN-Channels müssen überwacht werden. Da die gleichzeitige Überwachung aller Kanäle nicht möglich ist, führt der WLAN-Chip des Untersuchungsrechners sogenanntes Channel-Hopping durch. Der überwachte Kanal wird so in zeitlich regelmäßigem Abstand gewechselt. Hierbei können wertvolle Informationen verloren gehen. Als Lösung sollte möglichst schnell ermittelt werden, auf welchem Kanal der untersuchte Hotspot betrieben wird.

Eine weitere Schwierigkeit ist die ungenaue Abstrahlung von WLAN-Funk. Auf ihrem Weg werden Funkwellen an Oberflächen reflektiert, gebrochen oder absorbiert. So haben beispielsweise Wände, Türen und Glas einen Einfluss auf

Richtung und Stärke elektromagnetischer Wellen (Schemberg et al., 2019). Das erschwert die Richtungsbestimmung einer Funkquelle.

Auch der Störfunk auf einzelnen Kanälen wird bei der WLAN-Aufklärung zum Problem (Bertko & Weber, 2017). Aufklärungstools bieten zur Lösung eine Filterfunktion an, mithilfe derer sich der Funkverkehr einzelner Hotspots überwachen lässt.

Die zwei WLAN-Aufklärungstools, die in diesem Kapitel beschrieben werden, sind Airodump-ng und Kismet.

Airodump-ng ist Teil der Aircrack-ng Suite. Diese Werkzeugsammlung wurde geschrieben, um Penetrationstests in WLAN-Umgebungen durchzuführen. Nachdem der WLAN-Chip in den Monitor-Mode gesetzt wurde, wird mittels Airodump-ng der umgebende WLAN-Traffic beobachtet. Hierbei können die zu untersuchenden Channels ausgewählt werden. Zudem bietet das Programm Filter zur Untersuchung einzelner Hotspots (https://aircrack-ng.org/doku.php?id=airodump-ng, 2022).

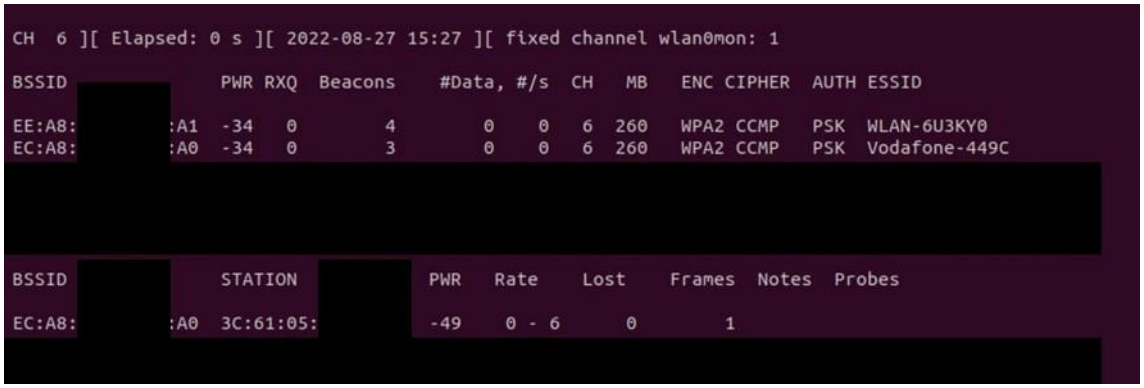


Abbildung 3 - Screenshot: Ausgabe von airodump-ng - Untersuchung Kanal 6

Abbildung 3 zeigt die Ausgabe des Programms Airodump-ng. Zum Zeitpunkt der Aufnahme wird der WLAN-Kanal 6 untersucht (CH 6). Im oberen Teil des Fensters werden neben den MAC-Adressen der Hotspots (BSSID - Basic Service Set Identifier) auch die Signalstärke (PWR - Power) und der Funknetzwerkname (ESSID- Extended Service Set Identifier) angezeigt. Der untere Teil der Abbildung zeigt verbundene Clients. Auch hier wird neben der BSSID des verbundenen Hotspots die MAC-Adresse des Clients (STATION) sowie die

Signalstärke (PWR) angezeigt. Zur Positionsbestimmung von Geräten liegen damit alle benötigten Informationen vor.

Das Funkaufklärungstool Kismet wurde nicht allein für die Analyse von WLAN-Netzwerken geschrieben. Weitere Funkprotokolle, beispielsweise Bluetooth, Bluetooth Low Energy, ZigBee lassen sich ebenfalls mit dem Programm untersuchen. Kismet bietet eine Weboberfläche zur Steuerung an. Hier lassen sich mehrere Datenquellen aktivieren, sodass unterschiedliche Protokolle zur selben Zeit empfangen und untersucht werden können. Auch die Frequenzkanäle der verschiedenen Funkprotokolle werden über das Webinterface ausgewählt (kismetwireless.net, 2022).

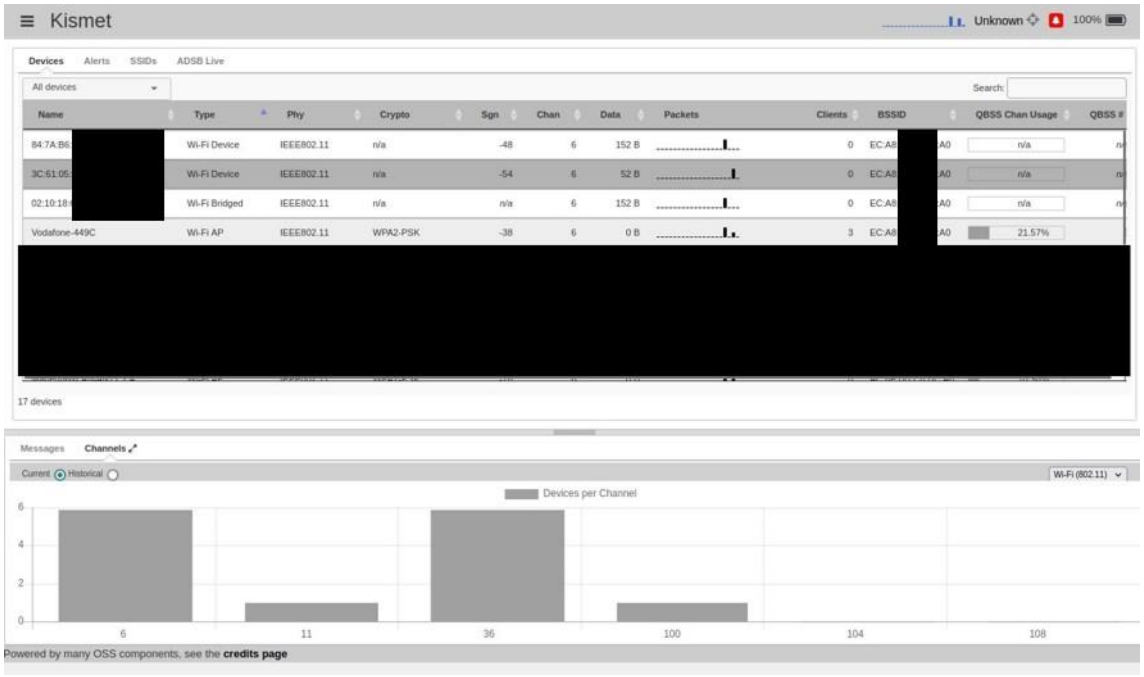


Abbildung 4 - Screenshot: Ausgabe von Kismet, WLAN-Scan

Die Oberfläche von Kismet ist in Abbildung 4 dargestellt. Die aufgelisteten Geräteinformationen zeigen den Gerätenamen (Name), die Signalstärke (Sgn) sowie die MAC-Adresse (BSSID) der Geräte. Auch wird mit dem Gerätetyp (Type) unterschieden, ob es sich um einen Accesspoint oder um einen Client handelt. Die zur Aufklärung benötigten Daten sind damit gegeben. Angeschlossene WLAN-Geräte können auch mithilfe der Benutzeroberfläche von Heimroutern ermittelt werden. Das Vorgehen wird im Kapitel Routeranalyse genauer beschrieben.

3.2.1.2 Bluetooth Low Energy

Das Funkprotokoll Bluetooth wurde zur Verbindung zwischen Geräten im Nahbereich entwickelt. Seit 2001 entwickelt die Bluetooth Special Interest Group den Standard IEEE 802.15.1 weiter. Mit der Veröffentlichung der Version Bluetooth 4.0 im Jahr 2010 wurde der Standard um das Protokoll Bluetooth Low Energy (Bluetooth LE oder BLE) erweitert (Bertko & Weber, 2017). Das Protokoll definiert die Datenübertragung auf der Bitübertragungs- sowie auf der Sicherungsschicht des OSI-ISO-Schichtenmodells (Gessler & Krause, 2015). Bluetooth Low Energy zeichnet sich durch seinen niedrigen Energieverbrauch aus. Dieser wird unter anderem erreicht, indem Geräte abgeschaltet werden, wenn keine Daten übermittelt werden müssen. Der Funkstandard wird daher meist für batteriebetriebene Geräte mit niedrigem Bedarf an Netzwerkbandbreite angewendet. Seit Einführung des Standards Bluetooth 5.0 kann eine Datenübertragungsrate von 2 Mbit/s erreicht werden. Die Reichweite wird mit bis zu 10m angegeben (elektronik-kompendium.de, 2022).

BLE wird auf dem ISM-Band (Industrial Scientific Medical Band) betrieben. In diesem Frequenzspektrum können Funkstationen lizenzfrei betrieben werden. Die Frequenz von Bluetooth Low Energie liegt im Bereich von 2,402 bis 2,478 MHz. Dieser Bereich ist in 40 Kanäle aufgeteilt, die Channel 0-36 dienen dabei zur Datenübertragung, während Channel 37-39 als Advertising Channels genutzt werden. Die Channel sind auf dem Frequenzspektrum nicht in Reihenfolge angeordnet (rfwireless-world.com, 2022). Mit Classic Bluetooth ist BLE nicht kompatibel.

Bluetooth LE kann entweder im verbindungslosen oder im verbindungsorientierten Modus betrieben werden. Im verbindungslosen Modus wird keine feste Verbindung aufgebaut, es wird ausschließlich zur Übertragung von Nutzdaten kommuniziert. Hierbei sendet der sog. Broadcaster in regelmäßigen Zeitabständen Advertising Beacons aus, die von sog. Observern empfangen werden können. Dazu hören Observer die drei Advertising-Kanäle ab. Die Beacons enthalten die zur Datenübertragung benötigten Informationen.

Im verbindungsorientierten Modus werden nicht nur die Nutzdaten übertragen, sondern auch Verbindungsdaten zum Aufbau und Halten der Kommunikation. Hierbei unterscheidet man die Geräte als Periphal (Slave) und Central (Master). Periphals senden sog. Connectable Advertiser Events aus, die ähnlich den Advertising Beacons die nötigen Informationen zum Verbindungsaufbau enthalten. Centrals scannen die drei Advertising-Kanäle nach diesen Events ab. Wird eine Verbindungsanfrage gesendet, so initiiert das als Central fungierende Gerät die Verbindung.

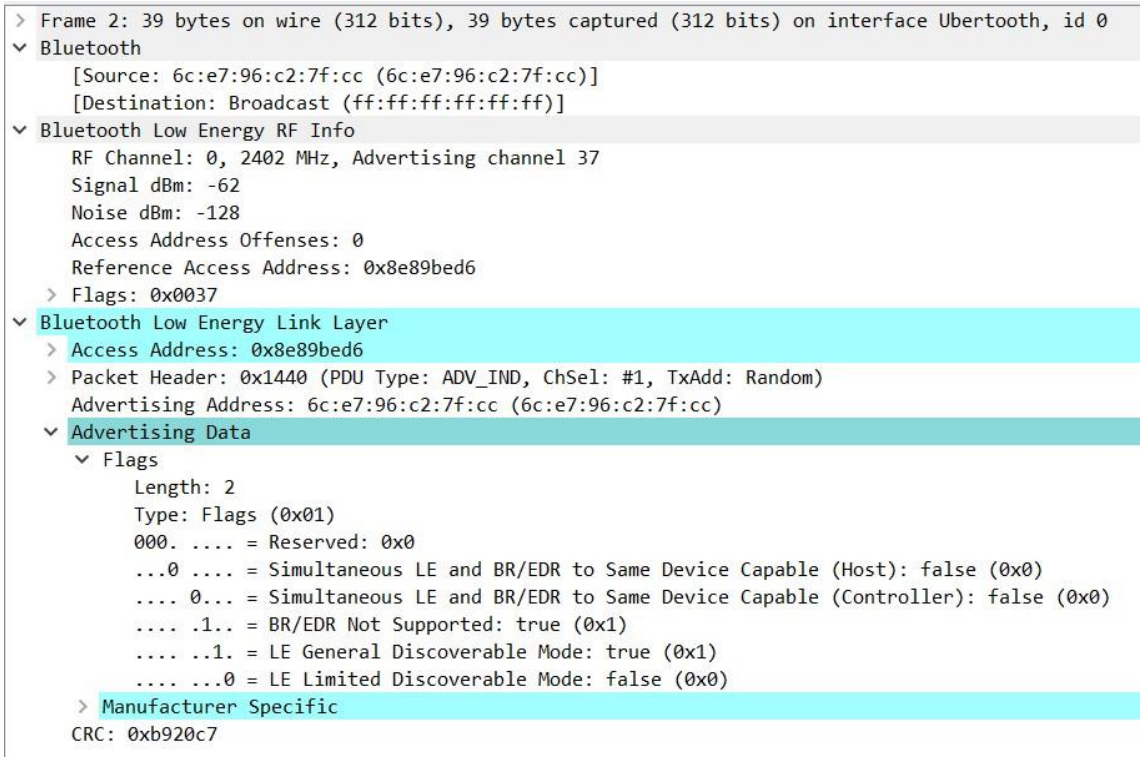


Abbildung 5 - Screenshot: Darstellung eines BLE-Beacon-Frames

Der in Abbildung 5 dargestellte Beacon-Frame ist ein Advertisement Indication Beacon (ADV_IND), mit welchem ein Peripheral-Gerät eine Verbindung zu einem beliebigen Central-Gerät in der Umgebung anfragt.

Bluetooth Low Energy wird in fünf unterschiedlichen States betrieben.

- Advertisment: Broadcaster (verbindungslos) oder Periphal (verbindungsorientiert) sendet auf den Kanälen 37-39 Advertisement events.

- Scanning: Observer (verbindungslos) oder Central (verbindungsorientiert) scannt die Advertising-Kanäle nach Advertisement events ab.

- Initiating: Im verbindungsorientierten Modus wird durch das als Central fungierende Gerät nach Empfang eines Connectable Advertiser Events der Verbindungsaufbau eingeleitet.

- Connection: Im verbindungsorientierten Modus steht eine Verbindung zwischen Central und Periphal. Die Nutzdaten können über diesen Kommunikationsweg transportiert werden.

- Standby: Muss ein Client keine Daten (beispielsweise Sensormesswerte) übertragen, so wird die Netzwerkschnittstelle abgeschaltet, um den Energieverbrauch zu senken.

Die beschriebenen Vorgänge zur Datenübertragung mittels Bluetooth Low Energy können genutzt werden, um BLE-Geräte zu lokalisieren. Mithilfe von BLE-Scannern kann der Netzwerkverkehr in der Umgebung abgefangen werden. Ein solcher Scanner ist der Ubertooth One der Firma Great Scott Gadgets (greatscottgadgets.com, 2021). Mithilfe von Netzwerkanalysesoftware können die Pakete bereits während des Scans analysiert werden.

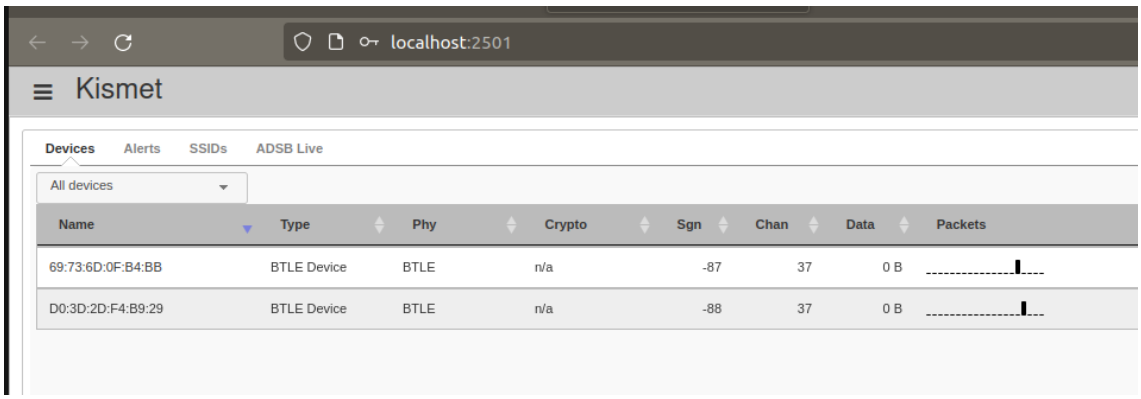


Abbildung 6 - Screenshot: Kismet Bluetooth Low Energy

Das Programm, welches hier als BLE-Scanning-Tool vorgestellt wird, ist Kismet.

Wie bereits im Kapitel WLAN erläutert, kann Kismet auch Bluetooth und Bluetooth-Low-Energy Daten darstellen (siehe Abbildung 6). Bei der Analyse von Bluetooth Low Energy müssen mehrere Aspekte beachtet werden. Geräte im Standby-Modus erzeugen keinerlei Netzwerkverkehr. Auch Centrals und Observer senden im Scanning-Modus keinerlei Daten aus. Daher sind diese Geräte mit den beschriebenen Methoden nicht ohne Weiteres auffindbar. Eine Lösung könnte es an dieser Stelle sein, absichtlich Gerätefunktionen auszulösen. Dazu muss das Gerät jedoch bekannt sein. Des Weiteren kann ein Langzeit-Scan erfolgen, falls Sensordaten nur in größeren Zeitabständen übermittelt werden.

Auch die Funktion Frequency Hopping Spread Spectrum (FHSS) erschwert die Aufklärung des BLE-Protokolls. Hierbei wird zwischen zwei Geräten beim Verbindungsaufbau ein Muster festgelegt, nach welchem der Übertragungskanal nach einer bestimmten Zeit gewechselt wird. So wird das Verfolgen des Datenstroms erschwert (elektroniknet.de, 2022).

3.2.1.3 ZigBee

Im Jahr 2004 wurde mit dem Standard IEEE 802.15.4 das Protokoll ZigBee von der ZigBee Alliance vorgestellt (Bertko & Weber, 2017). Der Standard definiert den Datenaustausch auf der physikalischen Schicht und der Sicherungsschicht des OSI-ISO-Schichtenmodells. ZigBee wurde darauf aufbauend zum Einsatz von Funkverbindungen mit geringstem Stromverbrauch konzipiert. In Europa wurde das Frequenzspektrum 868 MHz für ZigBee freigegeben, welches auf 16 Kanäle aufgeteilt wurde. In diesem Bereich beträgt die Datenübertragungsrate 20kbit/s (Gupta, 2019).

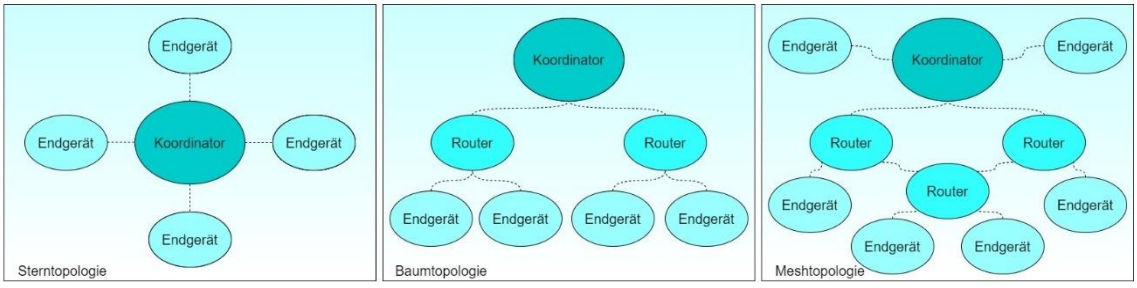


Abbildung 7 - ZigBee-Topologien (angelehnt an informit.com, 2009)

ZigBee kann in unterschiedlichen Netzwerktopologien betrieben werden, eine Anpassung des Netzwerks lässt sich einfach und flexibel umsetzen (siehe Abbildung 7). Zu den gängigen Topologien gehören Meshes, Sterne oder Baumstrukturen (Gessler & Krause, 2015).

Jedes Gerät kann in einem ZigBee-Netzwerk eine der folgenden Rollen übernehmen:

- Koordinator: ZigBee-Netzwerke sind hierarchisch aufgebaut, am Anfang jeder Hierarchie steht der ZigBee-Koordinator. Diese Rolle wird im Netzwerk einmalig vergeben. Der Koordinator ist für den Auf- und Abbau von Verbindungen, sowie für die Verwaltung der Topologie zuständig.

- Router: Das Routing von Datenströmen innerhalb eines Netzwerks, sowie das Auffinden der Route mit dem niedrigsten Energieverbrauch sind Aufgaben, die ZigBee-Router übernehmen. In einem Netzwerk kann es mehrere Router geben, über welche Netzwerkpakete ausgetauscht werden. Die Wegeanfragen, die zum effizienten Routing benötigt werden, werden an eine Broadcastadresse gestellt, sodass jeder Router im Netzwerk diese empfängt.

- Endgerät: Sensoren und Aktoren werden als Endgeräte in ZigBee-Netzwerke eingebunden. Ihre Aufgabe ist die Übermittlung von Sensordaten oder das Ausführen von Kommandos. Um Energie zu sparen, befinden sich diese Geräte die meiste Zeit im Sleep-Modus. Erst durch den Empfang eines Requests werden sie aktiv.

Jedes ZigBee-Gerät verfügt über eine eigene einzigartige 64-Bit-Geräteadresse, jedoch wird bei Zugang in ein Netzwerk eine 16 Bit-Adresse zugewiesen. Die Verwaltung obliegt dem ZigBee-Koordinator (Gessler & Krause, 2015).

Mit einem speziellen Funk-USB-Dongle ist es möglich, IEEE 802.15.4-Datenverkehr zu sniffen. Wieder kann zur Funkaufklärung das Tool Kismet auf

dem Untersuchungsrechner verwendet werden. Wird der Dongle als Datenquelle eingestellt, so können Pakete auf ausgewählten Kanälen empfangen werden.

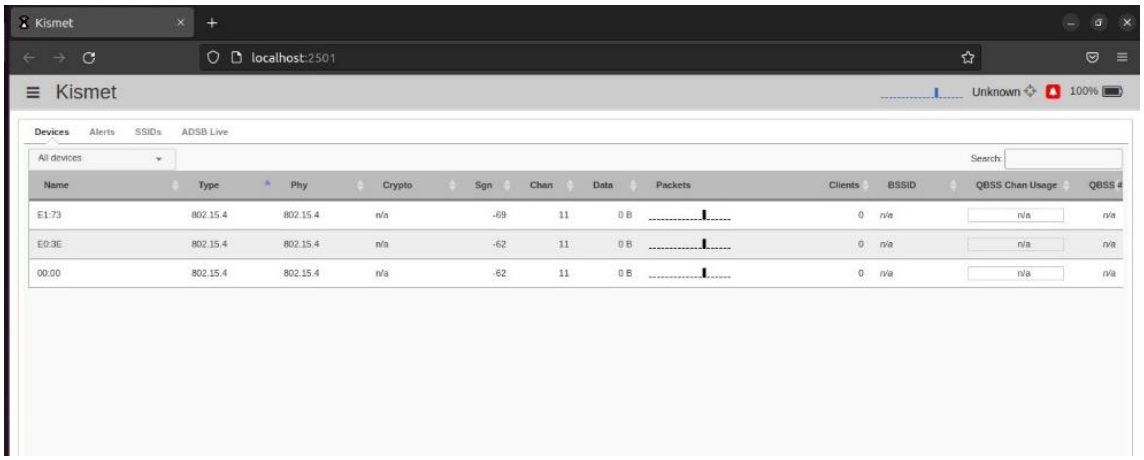


Abbildung 8 - Screenshot: Analyse von ZigBee mit Kismet

Abbildung 8 zeigt die Analyse von ZigBee mithilfe von Kismet. Da sich die ZigBee-Endgeräte im Normalfall im Sleep-Modus befinden, muss entweder auf einen Request des Koordinators gewartet werden, oder absichtlich ein Request ausgelöst werden.

Anschließend kann der Netzwerkverkehr mithilfe des Tools zbwireshark in Wireshark aufgezeichnet werden.

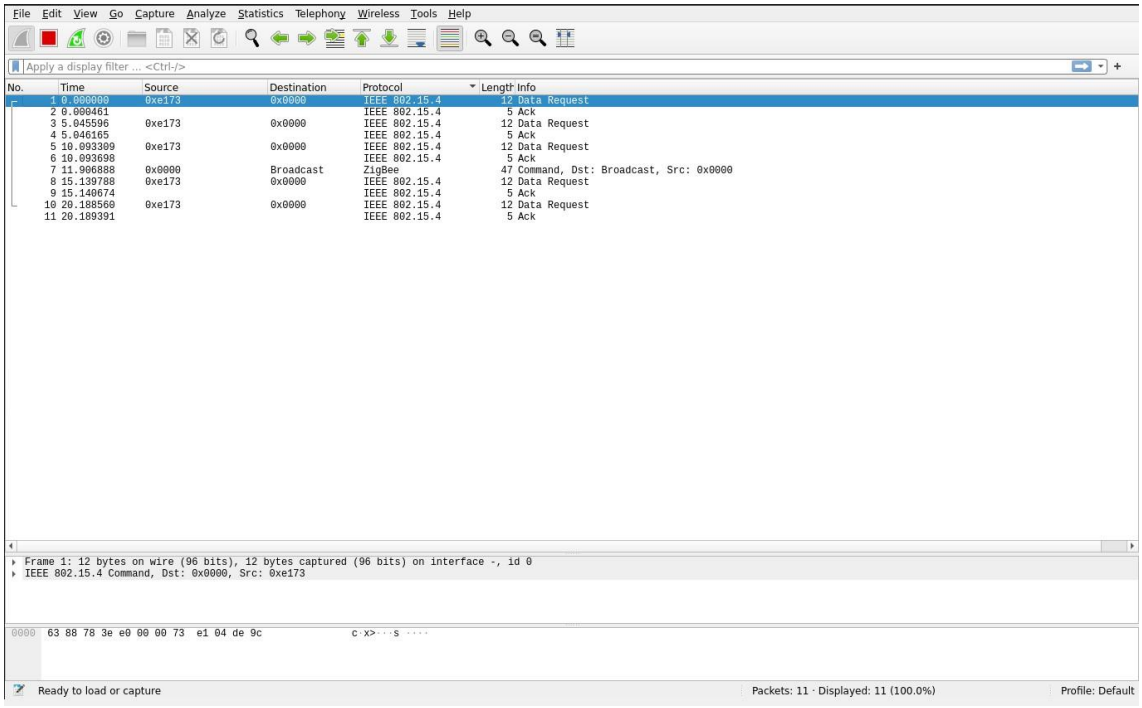


Abbildung 9 - Screenshot: Darstellung IEEE 802.15.4 Netzwerkverkehr in Wireshark

Abbildung 9 zeigt den Output des Kommandos

zbwireshark -c [Channel]

Zu sehen sind mehrere Data-Requests des Endgerätes an den Koordinator. Diesem ist immer die Adresse 0x0000 zugewiesen. Ebenfalls zu sehen ist die 16-Bit-PAN-ID des Endgerätes, in diesem Fall: 0xe173 (ZigBee Alliance, 2015).

Für das Auffinden der Geräte ist der in Kismet festgestellte Wert der Signalstärke (Sgn) ein entscheidender Hinweis, da er einen Rückschluss auf die Entfernung des Gerätes zum Untersuchungsrechner zulässt.

3.2.2 Weitere Aufklärungsmethoden

Grundsätzlich lassen sich Smart Home Geräte auch mit weniger technischen Möglichkeiten finden. Bei Tatortsicherungsmaßnahmen aufgefundene Verpackungen können Hinweise über verwendete Geräte geben. Besteht ein Zugriff auf ein tatorteigenes Smartphone oder ein verknüpftes E-Mail-Konto kann auch hier nach Hinweisen auf verwendete Systeme gesucht werden.

Eine weitere Option wäre der Einsatz von Wärmebildkameras zum Auffinden versteckter elektronischer Geräte. Unter Spannung stehende Platinenteile sondern detektierbare Betriebswärme ab.

Eine aufwändigere Option ist der Einsatz von Datenträgerspürhunden (ESD-Dog - Electronic Storage Detection Dog). Die Hunde können selbst sehr kleine elektronische Speichermedien wie Micro SD-Karten wahrnehmen. Ursprünglich wurden diese Hunde auf das Aufspüren spezieller Platinenkleber trainiert (tasteofthewildpetfood.com, 2021). Es konnte jedoch in weiterführenden Versuchen nicht endgültig ermittelt werden, welche von Platinen abgesonderten Gerüche die Hunde wahrnehmen. Die Suche ist zudem weniger erfolgsversprechend als beispielsweise die Suche nach Drogen, da Platinen weniger geruchsintensiv sind (Burger, 2019).

Die enge Zusammenarbeit mit dem klassischen Erkennungsdienst ist ein entscheidender Faktor für die Qualität der digitalen Tatortarbeit. Der eingespielte Arbeitsablauf der Spurensicherung am Tatort darf nicht gestört oder unterbrochen werden. Gleichzeitig ist eine zügige Sicherung flüchtiger Artefakte aus IT-Systemen notwendig, um keine Spuren zu verlieren. Hierbei wird es notwendig, die verschiedenen Arbeitsschritte beider Bereiche flüssig in einen gemeinsamen Workflow zu integrieren und Synergieeffekte zu nutzen.

3.3 Untersuchung von Smart Home Systemen

Artefakte aus Smart Home Systemen sind unterschiedlich lange haltbar. Bei der Untersuchung sollte der Lebensdauer der Artefakte große Aufmerksamkeit geschenkt werden. Kurzlebige Artefakte wie z.B. Routerlogs sollten grundsätzlich noch im laufenden Betrieb gesichert werden. Die Methoden hierzu werden im Kapitel Routeranalyse vorgestellt. Weniger flüchtige Daten, beispielsweise im Dateisystem abgelegte Zugangsdaten, können im Anschluss im Labor untersucht werden, wo die nötige Hardware und Infrastruktur gegeben sind. Vor Beginn der Analyse wird erörtert, welche Arten von Artefakten zu erwarten sind und wo diese gespeichert sein können. Anschließend wird nach Lebensdauer und Priorität die Reihenfolge festgelegt, nach der die Daten vor Ort und im Labor gesichert werden.

3.3.1 OSINT-Analyse

Bevor mit der eigentlichen Untersuchung begonnen wird, sollten über das System möglichst viele Informationen aus frei verfügbaren Quellen (Open Source Intelligence, OSINT) gesammelt werden. Ziel ist, sich einen Überblick über die Gerätefunktion zu verschaffen. Somit kann bereits abgeschätzt werden, mit welchen Arten von forensischen Daten zu rechnen ist. Zudem lassen sich auf diese Weise die nachfolgenden Analysen konkretisieren.

Webseite Hersteller

Eine erste Quelle ist die Webseite des Geräteherstellers. Hier finden sich Antworten auf Fragen wie:

- Gibt es für das System eine App zur Steuerung?
- Verfügt das System über eine Cloudanbindung?
- Welche weiteren Geräte gibt es aus diesem Ökosystem?

Außerdem finden sich auf der Webseite häufig Bedienungsanleitungen und Datenblätter des Gerätes, die einen Überblick über die Gerätefunktionen und

technische Details enthalten. Aus der Beschreibung der verschiedenen Einsatzbereiche lassen sich Hinweise auf forensische Artefakte ermitteln.

Konformitätserklärungen benennen die verwendeten Funkprotokolle und das verwendete Spektrum. Diese Kenntnisse werden für die Vorbereitung der Netzwerkanalyse genutzt.

FCC-ID

Die Federal Communications Commission (FCC) wurde mit der Aufgabe betraut, die Kommunikation über Funk, Satellit und Kabel in den USA zu regulieren. Jedes Gerät, welches auf einem dieser Kommunikationswege senden und empfangen will, muss von der FCC zertifiziert werden (fcc.gov, 2022). Mit der Zertifizierung erhalten Geräte eine eindeutige Identifikationsnummer, die FCC-ID. Anhand dieser Identifikationsnummer können in der Datenbank der FCC wertvolle Kenntnisse über ein Gerät gewonnen werden. Um die Recherche in der Datenbank zu erläutern, wurde als Beispiel die Base Station der Herstellerfirma Arlo (Modell VMB4500) gewählt. Ist die FCC-ID bekannt, so kann das Gerät auf der Webseite https://fccid.io/ gesucht werden. Unter der FCC-ID 2APLE18300391 können neben anderen folgende relevante Dokumente eingesehen werden (fccid.io, 2022):

- Internal Photos: Die Fotos zeigen das Innere und die Platinen der Base Station. Diese geben Hinweise auf serielle Schnittstellen und die verwendeten Mikochips.

- Bedienungsanleitung

- WiFi Channel Description: Beschreibt die verwendeten WiFi-Kanäle und Frequenzbänder

Google Scholar

Im Jahre 2004 startete Google das Projekt Google Scholar, eine Datenbank für akademische Literatur. Heute ist Google Scholar die größte Datenbank dieser

Art. Durch die einfache Stichwortsuche lassen sich wissenschaftliche Publikationen schnell gezielt finden. Sucht man nach „Arlo Base Station“, dann erhält man als ersten Suchtreffer die wissenschaftliche Arbeit von Sevida und Casey (2019) (scholar.google.com(1), 2022). Auch andere wissenschaftliche Datenbanken, beispielsweise Researchgate, können als offene Quellen genutzt werden (researchgate.net, 2022).

Schwachstellendatenbanken, z.B. die der NIST sind potenzielle Recherchequellen. Für “IoT-as-a-witness“ sind Schwachstellen und deren Ausnutzung jedoch weniger relevant (nvd.nist.gov, 2022).

Es ist zu beachten, dass die Phase der OSINT-Analyse vor Ort nicht zu viel Zeit in Anspruch nehmen sollte, da sonst eventuell schon die ersten Artefakte verloren gehen. Ein Teil der OSINT-Analyse kann während der nachfolgenden Laborarbeit erfolgen.

3.3.2 Netzwerkanalyse

Mit einer gezielten Analyse des durch ein Smart Home Gerät produzierten Netzwerkverkehrs soll das Verhalten des Gerätes eruiert werden. Die gewonnenen Erkenntnisse geben Hinweise auf Verbindungen zu Cloudspeichern und auf die Herkunft der Steuerungsbefehle. Zudem kann ermittelt werden, ob und wie relevante Sensordaten übermittelt werden und ob sich weitere Geräte im Netzwerk befinden. Zudem werden möglicherweise Nutzungsdaten im Klartext übertragen (Tekeoğlu & Tosun, 2015). Dieses Wissen kann sogar genutzt werden, um Nutzerverhalten nachzuvollziehen. So konnten die Forschenden in Hannan Bin Azhar et al. (2019) nur anhand des Netzwerkverkehrs den Schlafrhythmus von Smart Home Nutzer*innen nachvollziehen und Copos et al. (2016) erkennen, ob die Hausbewohner derzeit anwesend sind.

Das ZigBee-Protokoll wird an dieser Stelle nicht weiter betrachtet, da hier meistens Gateways als Koppler eingesetzt werden, die wiederum per WLAN am Heimrouter angeschlossen sind.

Bevor eine Untersuchung beginnt, kann eine Unterteilung der Kommunikationsbeziehungen hilfreich sein. Unterschiedliche Ansätze zum Kategorisieren von Netzwerkdaten sind:

- Passive und aktive Kommunikation: Welche Daten werden durch das Gerät ohne Interaktion erzeugt, welche Daten entweder durch Steuerbefehle oder durch Zustandsänderungen?

- Sensordaten und Steuerungsdaten (Hutchinson et al., 2020): Welche Daten werden durch Sensoren periodisch, durch Abfrage oder durch Auslösen eines Triggers erzeugt? Welche Daten werden benötigt, um ein Gerät zu steuern?

- Cloud, Netzwerk, M2M, Maschine zu Mensch: In Hannan Bin Azhar et al. (2019) wird diese feinere Unterteilung vorgestellt. Die Kommunikationsdaten werden nach dem jeweiligen Kommunikationspartner kategorisiert.

- Extern und intern: Welche Daten werden innerhalb eines Subnetzes erzeugt und verarbeitet? Welche Daten werden in oder aus einem Subnetz heraus gesendet?

- Cloud side Client side (Chung et al., 2017): Welche Kommunikationsdaten werden von Cloudlösungen übermittelt, welche Kommunikationsdaten durch Smart Home Systeme?

Die Kategorisierung der Daten hilft später dabei, die im Netzwerk abgebildeten Funktionen einzelner Geräte und ganzer Systeme nachzuvollziehen.

Eine weitere Entscheidung zur Netzwerkanalyse ist, welche Daten direkt vor Ort und welche Daten später im Netzwerklabor erhoben werden. Die Datenerhebung vor Ort gestaltet sich schwierig, da es wenig Möglichkeiten gibt ohne großen Eingriff den Netzwerkstrom zu sniffen. Die hierfür gegebenen Möglichkeiten wie z.B. ARP-Spoofing stellen eine forensisch relevante Änderung der

Netzwerkkonfiguration dar. Zudem sind die Maßnahmen nur sehr begrenzt erfolgsversprechend. Umgekehrt befinden sich die Geräte im Netzwerklabor nicht mehr in ihrem ursprünglichen Netzwerk, auch hier werden die Konfigurationen teilweise geändert. Eine originalgetreue Nachbildung des Netzwerks am Tatort ist schwierig, da nicht alle Einstellungen aller Geräte zu Beginn bekannt sind. Zudem muss für Labortests das Smartphone vorhanden und entsperrt sein, damit die App genutzt werden kann. Es besteht die Möglichkeit, einen Laborversuch mit Vergleichsgeräten durchzuführen. Diese Vergleichsgeräte haben jedoch nicht dieselbe Gerätekonfiguration der Asservate. Der Transport der Asservate ins Netzwerklabor sollte mithilfe einer mobilen Stromquelle erfolgen, damit die Geräte nicht zu lange spannungsfrei sind. So kann möglicherweise ein Verlust gespeicherter Konfigurationen vermieden werden.

Durch eine gut geplante Netzwerkanalyse können diese Hürden zumindest abgeschwächt werden. Beispielsweise kann in Bedienungsanleitungen von Smart Home Geräten überprüft werden, ob die Geräte ihre Konfiguration bereits bei Spannungsverlust oder durch Auslösen einer Reset-Funktion zurücksetzen. So kann die Entscheidung getroffen werden, ob die Geräte vom Strom getrennt werden, um eine Analyse wie in Kapitel Geräteuntersuchung beschrieben vorzunehmen.

Der Eingriff in ein Netzwerk am Tatort erfolgt in mehreren Stufen. Zunächst werden die Netzwerkpakete mit einem Sniffer, beispielsweise Wireshark, passiv mitgeschnitten (Hannan Bin Azhar & Bate, 2019). Zu diesem Zweck wird ein Untersuchungsrechner, meist ein Laptop, mittels LAN-Kabel an den Heimrouter angeschlossen. Es folgen aktive Maßnahmen wie Portscans mit nmap. Hierbei werden Geräte im Netzwerk durch Versenden modifizierter Anfragen auf offene Ports und die dahinterstehenden Services untersucht. Außerdem kann so ermittelt werden, ob sich weitere, bisher nicht aufgefundene Geräte im Netzwerk befinden, sofern diese angemeldet sind und mit einer Antwort auf die modifizierten Pakete reagieren. Weiterhin lassen sich offene Schnittstellen am Router erkunden. Portscans können, sofern die Anmeldedaten des WLAN-Netzwerks bekannt sind auch im Labor durchgeführt werden. Wurden die

grundlegenden Einstellungen dokumentiert, so können abschließend Trigger von Sensoren ausgelöst werden, um die dazugehörigen Interaktionen im Netzwerk zu erzeugen (Scientific Working Group on Digital Evidence, 2020). Aus den gewonnenen Erkenntnissen wird Netzwerktopologie entwickelt. Auf die Arbeit am Tatort folgen weitere Untersuchungen in einem Netzwerklabor. Ziel ist es, ein detaillierteres Bild über die Kommunikation und Funktion von Smart Home Geräten zu erlangen. So geben beispielsweise DNS-Requests einen Hinweis auf Kommunikation zu Cloudsystemen. Artefakte aus diesen Cloudsystemen werden in einem späteren Schritt analysiert (siehe hierzu Kapitel Cloudanalyse). Durch das Auslösen von Steuerbefehlen per App kann die Herkunft dieser Befehle und der Ablauf der Steuerung ermittelt werden.

Weitere Ziele sind offene Schnittstellen der Smart Home Geräte. So verfügen einige Systeme über einen SSH-Zugang, über den auf das Betriebssystem zugegriffen werden kann. Awasthi et al. (2018) zeigen, dass es sogar möglich ist, ein Image des Dateisystems zu erstellen.

Während der Untersuchung werden Patterns im Netzwerkverkehr sichtbar. Diese müssen anschließend eindeutig und erklärbar dargestellt werden (Stoyanova et al., 2020).

Der Aufbau eines Netzwerklabors kann in verschiedenen Varianten erfolgen. Den Kern des Netzwerks bildet ein WLAN-Hotspot mit Routingfunktionen. Dieser Router ist vollständig steuerbar und lässt sich leicht überwachen. Auch auf die Firewall sollte Zugriff bestehen, um bestimmte Datenströme, z.B. DNS-Anfragen, feingranular zu blockieren oder umzulenken (Tekeoğlu & Tosun, 2015).

Die Hotspot-Funktion muss in der Lage sein, das WLAN-Netz am Tatort zu imitieren. Man spricht hierbei von einem „Evil Twin“, einer Kopie eines Originalhotspots. Der Router kann entweder als dezidierte Hardware aufgestellt werden oder in Form einer virtuellen Maschine auf einem Untersuchungsrechner implementiert werden.

Eine weitere Komponente im Labornetz ist eine Maschine, von der aus Untersuchungen vorgenommen werden. Es bietet sich an, für diese Maschine

eine Betriebssystem-Distribution zu nutzen, die alle benötigten Netzwerktools vorkonfiguriert enthält. Bekannte Vertreter sind Kali Linux und Parrot OS. Diese Distributionen wurden ursprünglich zur Durchführung von Penetrationstests und weiteren Sicherheitsanalysen entwickelt und eignen sich aufgrund ihrer breiten Auswahl an Werkzeugen zur Analyse von Netzwerken als Betriebssysteme für die Untersuchungsmaschine.

Auch die Untersuchungsmaschine kann als virtuelle Maschine betrieben werden. Es bietet sich an, dasselbe System als Router und Untersuchungsmaschine zu nutzen, da so alle benötigten Tools direkt an der Netzwerkschnittstelle verwendet werden können. Nach und nach werden die Smart-Home Geräte einzeln in das Labornetzwerk eingebunden. Hierbei wird der Netzwerkverkehr am Router in verschiedenen Stufen überwacht.

Stufe 1: Router ohne angeschlossene Geräte: Der Grundzustand des Routers wird aufgezeichnet. Um das ganzheitliche Bild des Netzwerkstroms nicht zu verändern, sollten die im Grundzustand vom Router übermittelten Pakete nicht von der Analyse ausgeschlossen werden, sondern lediglich gefiltert. Wireshark bietet Filterfunktionen an, mit denen die Pakete verborgen werden können, ohne sie bereits beim Mitschnitt zu filtern.

Stufe 2: Einhängen des Gerätes: Während des Verbindungsvorgangs wird der Netzwerkverkehr mitgeschnitten, um später darlegen zu können, wie sich ein Gerät mit dem Netzwerk verbindet und welche Abfragen als Erstes getätigt werden.

Stufe 3: Nach Einhängen des Gerätes: Nach erfolgreicher Anmeldung am Netzwerk werden zum besseren Verständnis der Gerätefunktionen verschiedene Szenarien durchgespielt.

Szenario 1: Ruhezustand, kein Auslösen von Steuerfunktionen:

In dieser Versuchsreihe wird überprüft, wie sich ein Gerät ohne äußere Einwirkungen im Netzwerk verhält. Es entsteht ein zeitlich regelmäßiges Muster aus verschiedenen Abfragen am Router. Dieses Muster wird dokumentiert und kann später gefiltert werden.

Szenario 2: Auslösen von Steuerfunktionen:

Es werden verschiedene Gerätefunktionen ausgelöst. Beispielsweise kann das Gerät aus- und wieder eingeschaltet werden, es können Abfragen von Sensordaten erfolgen oder es wird eine Alarmfunktion ausgelöst. Forensiker*innen müssen hierbei möglichst jede verfügbare Gerätefunktion in das Szenario aufnehmen und die entstandenen Kommunikationsdaten dokumentieren.

Die Veränderung von Livedaten bei der Nutzung von Portscannern stellt Forensiker*innen am Tatort vor Probleme, da jede vorgenommene Veränderung am Ursprungszustand später erklärbar dokumentiert sein muss. Demgegenüber steht die Tatsache, dass die Geräte im Labor nicht mehr in ihrer ursprünglichen Umgebung am Tatort laufen. Dies gilt sowohl für die physikalische Umgebung als auch für die Netzwerkumgebung. Es muss abgewogen werden, in welcher Untersuchungsphase ein Portscan erfolgt. Wurden am Tatort bereits ausreichende Informationen über genutzte Geräte gesammelt, dann sollte der Portscan im Labor erfolgen. Besteht kein Zugriff auf die Konfigurationsoberfläche des Heimrouters, dann sollte der Eingriff am Tatort erfolgen, um möglicherweise bisher nicht aufgespürte Geräte zu ermitteln.

Eine weiterführende Maßnahme ist das Umleiten von DNS-Anfragen. Hierdurch kann ein Gerät dazu gebracht werden, die IP-Adresse weiterer alternativer Steuerungsserver abzufragen.

Versuche der Nachbildung eines angesprochenen Applikationsservers und Replay-Angriffe mit Steuerbefehlen werden an dieser Stelle nicht weiter betrachtet. Diese Methoden werden für Angriffe auf Smart-Home-Systeme verwendet, nicht für forensische Untersuchungen.

Die Netzwerkanalyse stellt IT-Spezialisten vor Herausforderungen, die im Folgenden genauer beschrieben werden.

Mithilfe von Verschlüsselungstechniken wird die Privatsphäre von Smart Home Nutzer*innen geschützt. Das Abhören der Kommunikation zwischen Geräten oder zwischen Servern und Geräten wird so erschwert. Die Daten liegen nicht im

Klartext vor und eine Entschlüsselung ist ohne Weiteres nicht möglich. Für die Forensik bedeutet das, dass nur die Informationen zur Analyse genutzt werden können, die im Klartext vorliegen. Hierzu gehören verwendete Protokolle, Quell- und Zieladressen sowie offene Paketinhalte wie URLs aus DNS-Abfragen. Die Debatte über Backdoors für Ermittlungsbehörden wird mit zahlreichen Pro- und Contraargumenten geführt (siehe z.B. Briegleb, 2020). Bislang sind solche Eingriffsmöglichkeiten nicht vorgesehen. Ein bereits dargestelltes Problem ist, dass eine Laborumgebung nicht mehr dem ursprünglichen Umfeld des Smart Home Gerätes entspricht. Dieser Anforderung wird begegnet, indem das Netzwerk des Tatortes möglichst originalgetreu nachgebildet wird. Vor Ort müssen alle dafür benötigten Informationen gesammelt werden. Hierzu gehören neben Anmeldeinformationen auch Gerätekonfigurationen und offene Schnittstellen im Router.

3.3.3 Routeranalyse

Der Router übernimmt in Heimnetzwerken eine zentrale Rolle. Neben Netzwerkrouting stellt das Gerät einen WLAN-Hotspot zur Verfügung, der als Koppelelement zwischen drahtlosen Geräten und kabelgebundenem Ethernet fungiert. Zudem stellen die meisten Heimrouter eine Firewall zwischen Heimnetz und Internet. Diese Firewall kann zusätzlich zu ihrer Schutzfunktion Portweiterleitungen stellen. Mithilfe dieser Technik wird der Zugriff aus dem Internet auf Geräte hinter der Firewall möglich. Daher sollten diese Geräte sicherheitstechnisch gehärtet werden.

Auch die zentrale Geräteverwaltung gehört zu den Aufgaben eines WLAN-Routers. Es können beispielsweise MAC-Adressen-Filter eingestellt werden, die nur Geräte mit bestimmten Kennungen zulassen. Weitere Anwendungen sind Zeitschaltungen für Internet-Zugang oder Jugendschutzfunktionen für einzelne Geräte. Einige Router bieten erweiterte Funktionen wie VPN-Tunnel an. So kann ein verschlüsselter Tunnel zwischen Endgerät und Heimnetzwerk in unsicheren öffentlichen Netzwerkumgebungen genutzt werden.

Ziel der Routeranalyse ist es, die Konfigurationen des Netzwerks forensisch zu sichern. In erster Linie sind Informationen über derzeit aktive, aber auch inaktive,

angemeldete Geräte die Forensik von Belang. Hierbei sind neben den Gerätenamen auch die MAC- und IP-Adresse zu dokumentieren. Die ersten drei Bytes der MAC-Adresse bezeichnet man als OUI (Organizationally Unique Identifier). Diese OUIs sind einem Hersteller zugewiesen, der so in öffentlichen Datenbanken ermittelt werden kann. So kann die Herstellerfirma eines Gerätes, bzw. des WLAN-Chips, ermittelt werden (macvendor.info, 2022).

Weiterhin können aus der Routerkonfiguration Zugangspasswörter zum WLAN oder Gastnetzwerken ausgelesen werden. Diese sind beim Aufbau eines originalgetreuen Labornetzes nützlich. Auch der Betrieb des Routers im 5 GHz Spektrum ist hier konfiguriert.

Neben Geräteeinstellungen finden sich in Routern Logdateien, die Meldungen zu Ereignissen im Netzwerk in chronologischer Reihenfolge beinhalten. Mithilfe dieser Logs sowie den Geräteinformationen kann eine Timeline aus Ereignissen erstellt werden. So kann ermittelt werden, welche Geräte zu einem bestimmten Zeitpunkt angemeldet waren. Außerdem ist es anhand dieser Zusammenstellung unter Umständen möglich festzustellen, ob ein Gerät im Gebäude bewegt oder genutzt wurde. Da die Logdateien in einem Ringpuffer geschrieben werden, ist die Routeranalyse als eine der ersten Maßnahmen am Tatort durchzuführen (Dorai et al., 2018).

Für die Analyse wird zunächst ein Untersuchungsrechner per LAN-Kabel an den Router angeschlossen. Der Vorteil gegenüber einer Anmeldung per WLAN liegt darin, dass kein Passwort benötigt wird und kein unnötiger Funkverkehr am Tatort erzeugt wird.

Fast alle gängigen Heimrouter bieten eine Weboberfläche zur Administration. Diese kann durch Eingabe der IP-Adresse des Routers in einem Browser aus dem Netzwerk aufgerufen werden. Meist wird diese Oberfläche durch ein Kennwort geschützt. Das Standardkennwort ist in vielen Fällen auf den Router gedruckt. Ist dies nicht der Fall, so kann versucht werden, das Handbuch und die Verpackung des Routers zu finden.

Die Weboberfläche bietet einen Überblick über die Geräte- und

Netzwerkeinstellungen. Daneben können auch Logdateien ausgelesen und in einer Datei exportiert werden.

Wurde das Standardkennwort geändert, kann versucht werden dieses über eine serielle Schnittstelle auszulesen. So bietet die Fritz!Box 4020 von AVM eine serielle Schnittstelle, die einen Zugriff auf das Dateisystem zulässt. Die Auswertung von Geräten über serielle Schnittstellen wird in Kapitel Geräteanalyse erläutert. Die Firmware der Fritz!Box basiert auf Linux, das Gerätepasswort liegt als Hash in der Datei /etc/passwd ab (Luber & Donner, 2019).

Das Projekt OpenWRT stellt Informationen über serielle Schnittstellen und Firmware von Routern verschiedener Herstellerfirmen zur Verfügung. So wird z.B. eine Anleitung zum Aufbau einer Shell-Session bereitgestellt (openwrt.org, 2022).

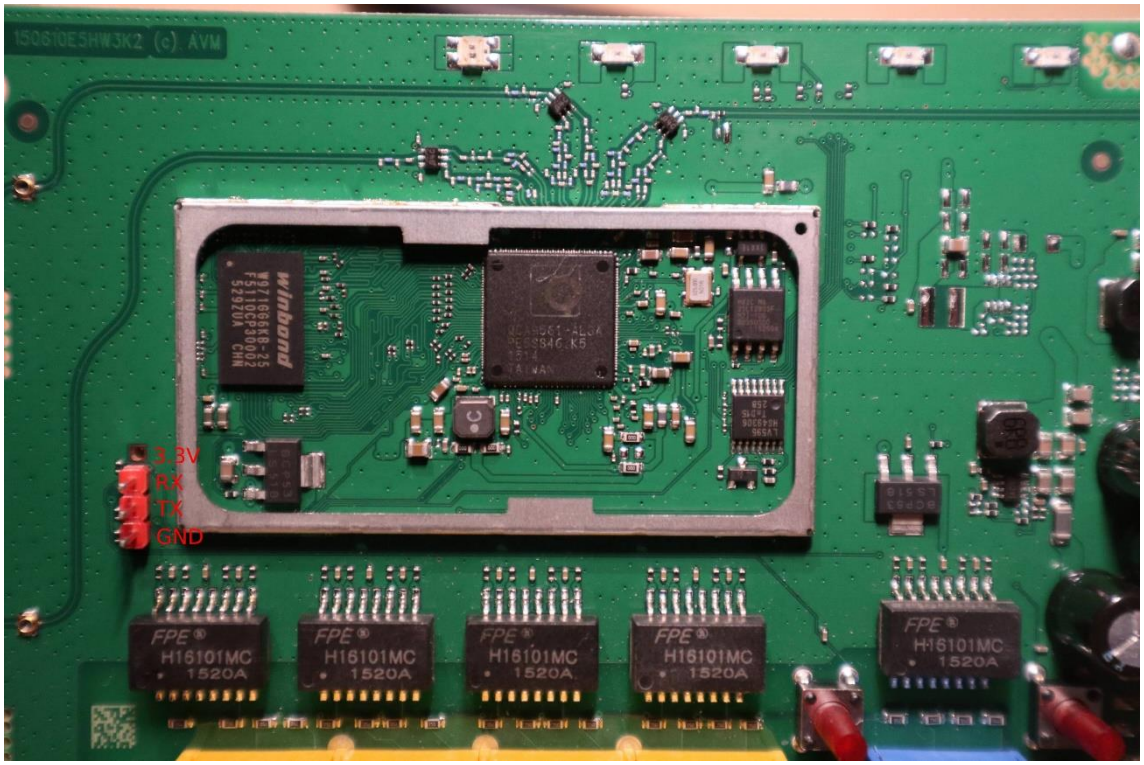


Abbildung 10 - UART-Pinout der Fritz!Box 4020 (Quelle: https://openwrt.org/toh/avm/fritz.box.4020)

Abbildung 10 zeigt die serielle Schnittstelle der Fritz!Box 4020. Der Zugriff auf das Dateisystem des Routers erlaubt neben dem Auslesen des

Standardkennwortes auch die Analyse weiterer Logdateien. Neben der seriellen Schnittstelle können einige Routermodelle mittels SSH oder Telnet administriert werden. Die dafür benötigten Daten liefert ein Portscan.

3.3.4 Appanalyse

Die Steuerung von Smart-Home-Systemen erfolgt in den meisten Fällen per Applikation über das Smartphone (Babun et al., 2018). Die Geräte können über eine Benutzeroberfläche verwaltet, angesteuert und Messwerte von Sensoren abgelesen werden. Auch Automationsfunktionen und Zugriffe auf Cloudspeicher erfolgen mittels App. Während der Nutzung werden Daten erzeugt, welche auf dem Smartphone oder in der Cloud abgespeichert werden. Ziel der Appanalyse ist es, diese Daten zu sammeln und auszuwerten, um daraus Hinweise auf das Verhalten des Smart-Home-Systems vor und während einer Tat zu erhalten. Apps können zudem dabei helfen, die grundlegenden Funktionalitäten eines Systems zu verstehen.

Weiterhin werden von Smartphone-Apps forensisch relevante Benutzerinformationen gespeichert. Beispielsweise stellt die Smart Home App von Tuya eine Familienfunktion bereit. So können mehrere Bewohner eines Haushaltes auf die gleichen Geräte zugreifen. Die Information, wer Zugriff auf das System hat, hat Auswirkungen auf die spätere Bewertung der gefundenen Nutzungsartefakte (tuya.com, 2022). Zu den Benutzerinformationen zählen des Weiteren eingestellte Automatismen oder Zeitpläne. Auch Zugangsdaten zu Cloudanwendungen können in Konfigurationsdateien vorliegen (Sevida & Casey, 2019).

In ihrer Funktion als zentrales Management des Systems beinhalten Apps Informationen über gesteuerte Geräte. Auch durch Zustandsänderungen ausgelöste Meldungen, beispielsweise Alarme bei Überschreiten eines Sensorschwellwertes, werden in der App angezeigt. Einige Geräte mit Bild- und Audioaufzeichnungsfunktion übermitteln Mediendateien. Diese werden meist in der Cloud abgespeichert, sind aber über die App abrufbar (tuya.com(1), 2022).

Für die Appanalyse ist entscheidend, ob Daten im Speicher des Smartphones oder im Cloudspeicher abgelegt werden. Das Öffnen der App im Flugzeugmodus kann einen Hinweis zu dieser Fragestellung geben. Daten, die ohne Netzwerkverbindung angezeigt werden, werden im Smartphone vorgehalten. Im Dateisystem des Smartphones werden forensisch relevante Informationen teils im Klartext, teils aber auch verschlüsselt oder codiert abgelegt (Babun et al., 2018). Zur späteren Auswertung der zweiten Kategorie wird später ein Parser notwendig. Eine weitere Datenquelle stellen Backups von Smartphones dar. Hierbei gilt jedoch zu beachten, dass die hier gesicherten Daten meist nur Nutzerdaten und keine durch Apps erzeugten Daten beinhalten.

Die Durchführung der Tatortmaßnahmen hängt maßgeblich davon ab, ob die Bildschirmsperre des Smartphones deaktiviert werden kann. Häufig ist das nicht der Fall. Liegt ein Smartphone entsperrt vor oder ist der Entsperrcode bekannt, so ist die am geringsten invasive Methode zur Vorabsicherung von Daten das Abfotografieren des Bildschirms. So können bereits erste Erkenntnisse über verwendete Smart Home Produkte gewonnen werden.

Im Netzwerklabor wird die Untersuchung in drei Schritte aufgeteilt. Zunächst wird die Applikation selbst einer genaueren Analyse unterzogen. Android-Apps können aus dem Playstore auf ein Vergleichsgerät heruntergeladen und anschließend über die ADB-Schnittstelle an einen Untersuchungsrechner übermittelt werden. Die App liegt auf dem Untersuchungsrechner als apk-Datei vor (Gupta, 2019). Das apk-Format ist ein komprimiertes Archiv, welches den Sourcecode, Konfigurationsdateien sowie benötigte Ressourcen und Bibliotheken der App enthält (developer.android.com(1), 2022). Der Sourcecode liegt im Dalvik Executable-Format (.dex) vor.

Eine für die Analyse benötigte Standardkonfigurationsdatei ist Manifest.xml. Hier werden unter anderem benötigte Berechtigungen, Systemvoraussetzungen, genutzte Bibliotheken sowie die Komponenten einer App deklariert. Die Manifest.xml-Datei gibt somit einen ersten Überblick über die unterschiedlichen Funktionen einer App (developer.android.com(2), 2022).

Android-Apps verfügen über unterschiedliche Komponenten, die in der folgenden

Kurzbeschreibung erläutert werden (developer.android.com(1), 2022).

- Activities: Aktivitäten beschreiben die Benutzeroberfläche einer App. Jeder Einzelbildschirm ist als eine eigene Aktivität. Liegen die benötigten Berechtigungen vor, dann kann eine App auch auf Aktivitäten von anderen Apps zugreifen. So kann beispielsweise die Kamerafunktion in Chatapplikationen genutzt werden.

- Services: Services definieren Hintergrundfunktionen in Apps. So muss die App nicht mit einem Userinterface gestartet sein, damit sie im Hintergrund laufen kann. Ein Beispiel ist eine Musikapplikation, die den Ton weiter abspielt, obwohl im Vordergrund eine andere App verwendet wird. Hierzu gehören aber auch viele Apps, deren Aktivität vom Benutzer*innen nicht bemerkt wird.

- Broadcast Receivers: Ein Broadcast Receiver gibt Meldungen, die von Apps im Hintergrund erzeugt wurden an den User weiter. Zum Beispiel können Chatapplikationen den Empfang einer neuen Nachricht melden, ohne die App im Vordergrund zu unterbrechen.

- Content Providers: Diese Komponenten sind für die Speicherung von Daten im Smartphone oder in Cloudspeichern zuständig. Ihnen obliegt die gesamte Verwaltung des Datenspeicherungsprozesses. Mit den nötigen Berechtigungen können Content Provider auch in den zugewiesenen Datenbereich anderer Applikationen schreiben. Ebenso kann mithilfe des Content Providers auf die erzeugten Daten anderer Apps zugegriffen werden.

Ebenfalls in der apk-Datei enthalten sind die benötigten Ressourcen einer App. Dazu gehören z.B. Bilder, Audiodateien, Stylesheets und Layouts. Jede dieser Ressourcen ist in einer eigenen xml-Datei deklariert.

Neben der Analyse der apk-Datei gilt es, das System vom Tatort mithilfe von

Vergleichsgeräten oder Emulatoren möglichst originalgetreu nachzubauen, um ein fundiertes Verständnis für die Funktionen zu erlangen. Dieser Schritt kann mit der Netzwerkanalyse zusammengelegt werden. Verschiedene Szenarien werden nachgestellt und das Verhalten des Smartphones während der Durchführung dokumentiert.

Es folgt die Analyse des Originalasservates. Smartphones sind mittlerweile stetige Begleiter, die zahlreiche höchstpersönliche Daten der Nutzer*innen enthalten (Babun et al., 2018). Dementsprechend werben Herstellerfirmen mit immer neuen Sicherheitsfunktionen. Die Analyse von Smartphones wird daher immer weniger erfolgsversprechend. Entwicklerfirmen von forensischer Software müssen mit einer sich schnell ändernden Technik schritthalten. Die Erstellung von vollständigen physikalischen Images des Gerätespeichers wird durch die neuen Sicherheitsmechanismen verkompliziert (Awasthi et al., 2018). Hinzu kommt, dass forensische Software in den meisten Fällen Exploits verwendet, um den Gerätespeicher auszulesen. Diese Technik steht aus datenschutzrechtlichen Gründen in der Kritik (Yaron & Benjakob, 2021).

Gelingt es, ein physikalisches Image des Gerätespeichers zu erstellen, können forensische Artefakte im Dateisystem gesammelt werden. Die durch Applikationen erzeugten Daten werden in verschiedenen Speicherorten im Dateisystem abgelegt, die im Folgenden erklärt werden (developer.android.com(3), 2022).

- Appspezifischer Speicher: Auf diesen Speicherplatz hat zunächst nur die Applikation selbst Zugriff. Es besteht die Möglichkeit, die abgelegten Daten durch Gewähren der nötigen Berechtigungen an andere Apps weiterzugeben. Die Daten werden unterteilt in persistente Dateien und Cachedateien. Bei Löschung der App wird auch der appspezifische Speicherbereich gelöscht.

- Geteilter Speicher: Hier werden durch Apps erzeugte oder empfangene Medien und Dokumente abgelegt. Alle Apps im Smartphone haben auf

diesen Speicherbereich Zugriff. Bei Löschung einer App werden die erzeugten Daten im geteilten Speicher nicht gelöscht.

- Einstellungen: Hier liegen Einstellungen der App in Form von key:value-Paaren ab. Bei Löschung der App werden die Einstellungen ebenfalls gelöscht.

- Datenbanken: Datenbanken werden im Ordner /databases gesichert. Zugriff auf ihre Datenbanken hat nur die App selbst. Bei Löschung werden die Datenbanken ebenfalls gelöscht.

3.3.5 Cloudanalyse

Die fortschreitende Vernetzung digitaler Infrastruktur bringt neue Anforderungen mit sich. Wachsende Datenmengen müssen jederzeit von überall erreichbar sein. Cloudinfrastrukturen wurden entwickelt, um dieser Anforderung gerecht zu werden. Beim Cloud Computing liegen Daten nicht mehr auf lokalen Geräten vor, sondern werden in entfernten Datenzentren gespeichert. Die Daten können dann über ein Netzwerk erreicht werden (Gruschka, 2021).

Die NIST definiert Cloud Computing durch 5 charakteristische Merkmale (NIST, 2011).

- On-demand self-service: Cloudnutzer*innen können die Cloudanwendung jederzeit und ohne Interaktion mit dem Cloudprovider nutzen.

- Broad Network Access: Die Cloudanwendung kann von überall und von jedem Endgerät aus erreicht werden.

- Ressource Pooling: Die Daten einzelner Cloudnutzer*innen werden im selben Datenspeichersystem abgelegt. So können Kosten für dezentrale Datenspeicherlösungen minimiert werden.

- Rapid elasticity: Sollte sich kurzfristig die Anforderung an Cloudspeicherplatz ändern, kann dieser Bedarf schnell und flexibel abgedeckt werden.

- Measured service: Das Cloudprodukt lässt sich feingranular abrechnen. Cloudnutzer*innen haben die Kontrolle über die genutzten Dienste und kann diese ggf. einzeln ankaufen oder abwählen.

Es existieren unterschiedliche Cloud-Servicemodelle. Diese unterscheiden sich anhand der Ressourcen, welche den Nutzer*innen angeboten werden. Ressourcen können hierbei Applikationen, Rechenleistung, Datenspeicher oder ganze Infrastrukturen sein (Simou et al., 2014).

Wird Nutzer*innen lediglich eine Anwendung über das Netzwerk mittels Softwareclient zugänglich gemacht, so spricht man vom Dienstmodell Software as a Service (SaaS). Die Software selbst und die dahinterliegende Infrastruktur werden vom Cloudanbieter administriert. Streaming-Plattformen stellen ein Beispiel für SaaS-Plattformen dar.

Beim Servicemodell Plattform as a Service (PaaS) wird den Cloudnutzer*innen ein System zur Verfügung gestellt, auf welchem er selbsttätig Applikationen installieren und administrieren kann. Ein bekannter Vertreter ist Microsoft Azure, eine Softwareentwicklungsplattform, die auch zum Aufbau von Testumgebungen genutzt werden kann (azure.microsoft.com, 2022).

Das dritte Servicemodell ist Infrastructure as a Service (IaaS). Hierbei werden den Cloudnutzer*innen ganze IT-Infrastrukturen oder Rechenleistung zur Verfügung gestellt. Die Virtualisierung der Systeme obliegt den Cloudnutzer*innen. So können virtuelle Infrastrukturen einfach und flexibel an den Bedarf angepasst werden. Als Beispiel dient die Google Cloud Infrastructure (cloud.google.com, 2022).

Für Smart Home Anwendungen bringt die Nutzung der Cloudtechnologie einige Vorteile mit sich. Die Nutzer*innen sparen sich den Aufbau einer eigenen

Infrastruktur zur Verwaltung des Systems. Weiterhin können die Smart Home Geräte von überall gesteuert werden. Da Smart Home Geräte kompakt aufgebaut sind verfügen sie über wenig Speicherplatz. Die ersten IP-Kameras speicherten aufgezeichnete Videodaten noch in internen Speichern oder angeschlossenen Speichermedien (Tekeoğlu & Tosun, 2015). Mittlerweile werben die meisten Smart Home Anbieter mit der Möglichkeit, die Daten cloudseitig zur Verfügung zu stellen und zu verwalten. Smart Home Anwendungen werden meist über einen Client per Weboberfläche oder Smartphone App gesteuert, das Cloudservicemodell ist SaaS.

Die Analyse von forensischen Artefakten in der Cloud wird somit erschwert, da die Daten vom Cloudprovider verwaltet werden. Cloudnutzer*innen haben kaum Zugriffsmöglichkeiten auf die Speichersysteme (Simou et al., 2014). Des Weiteren sind die angebotenen Cloudlösungen für Smart Home Systeme ähnlich heterogen wie die Systeme selbst.

(Ruan et al., 2011) unterteilen die neuen Herausforderungen in der Cloudforensik in die drei Teilaspekte technische, juristische und organisatorische Probleme. Zu den technischen Problemen zählt die große Datenmenge, die bei einer Untersuchung in Betracht gezogen werden muss. Die kostengünstigen Cloudspeicher erlauben es, überproportional viele Daten und Metadaten abzulegen. Hinzu kommt, dass die Metadaten und Logdateien häufig getrennt von Nutzerdaten aufbewahrt werden, sodass bei der Analyse zunächst ein Zusammenhang hergestellt werden muss. Die Volatilität der Daten ist ausschließlich dem Cloudprovider bekannt, es kann folglich in der Planung forensischer Maßnahmen keine Sicherungsreihenfolge festgelegt werden (Ruan et al., 2011). Da Cloudsysteme weltweit verteilte Datenzentren nutzen, muss vor Beginn der forensischen Analyse festgestellt werden, wo die Daten abgelegt sind. Diese Frage kann ohne Unterstützung des Cloudproviders nicht beantwortet werden (Zawoad & Ragib, 2015). Cloudanbieter werben mit immer neuen Sicherheitsmechanismen wie Zwei-Faktor-Authentifizierung oder Datenverschlüsselung. Diese Datenschutzfunktionen erschweren, neben fehlenden Zugangsdaten, die forensische Analyse von Cloudartefakten weiter (Chung et al., 2017).

Auch juristische Aspekte müssen im Bereich der Cloudforensik betrachtet werden. Durch die unter Umständen weltweite Verteilung der Daten, fällt die Beantragung einer forensischen Untersuchung durch Ermittlungsbehörden in unterschiedliche Rechtsräume. Die juristischen Prozesse im Hintergrund bremsen die strafrechtliche Ermittlung weiter aus. Hinzu kommt, dass unterschiedliche Akteure den Prozess beeinflussen, die in juristische Entscheidungsprozesse eingebunden sein müssen. Die Servicelevel-Agreements (SLA) der Cloudprovider definieren in den wenigsten Fällen das Vorgehen bei Anfragen von Ermittlungsbehörden (Ruan et al., 2011).

Zu den Akteuren im forensischen Prozess zählen die Ermittlungsbehörden selbst, die Cloudprovider, Cloudnutzer*innen und sog. Cloudnachbarn. Cloudnachbarn sind Cloudnutzer*innen, die im selben Speichersystem gesichert sind wie forensisch relevante Clouddaten. Es ist oftmals schwierig zu ermitteln, welche Daten auf einem Speichersystem welchen Nutzer*innen zuzuordnen sind. Literaturquellen bieten eine Fülle an Forschung zu Cloudanwendungen als Angriffsziel, beispielsweise Simou et al. (2014), jedoch wenig zu Cloud als Quelle forensischer Artefakte.

Die Cloudkomponente eines Smart Home Systems dient einerseits als Steuerungsserver, andererseits als Datenspeicher. Die Datenspeicherfunktion muss in vielen Fällen zugekauft werden (tuya.com(1), 2022). Zudem werden über die Cloud Softwareupdates für einzelne Geräte und der Gerätestatus verwaltet. Jede Kommunikation zwischen Nutzer*innen und Gerät findet über die Steuerungsserver in der Cloud statt. Die in der Cloud abgelegten Daten, Metadaten und Logs können für strafrechtliche Ermittlungen wertvolle Informationen enthalten. So können beispielsweise Aufzeichnungen von IP-Kameras einen Einbruch zeigen, oder Logdateien Aufschluss darüber geben, wann ein Alarm ausgelöst wurde.

In Kapitel Netzwerkanalyse wurde dargestellt, wie sich genutzte Cloudanwendungen ermitteln lassen. Im nachfolgenden Schritt muss ein Zugang zu diesen Clouddaten gefunden werden. Logische Datensicherungen aus Cloudspeichern sind abhängig vom Servicemodell (Simou et al., 2014). Werden die Daten von Cloudnutzer*innen selbst verwaltet, besteht oftmals die

Möglichkeit, diese direkt oder über eine Exportschnittstelle herunterzuladen. Hierbei ist zu beachten, dass die Metadaten nicht im Download zur Verfügung stehen (Kim et al., 2020). Auch einige Herstellerfirmen von forensischer Software bieten die Option, Daten bestimmter Cloudprodukte automatisiert logisch zu sichern (Axiom, 2022). Die Fülle an Cloudanbietern lässt diese Möglichkeit jedoch nur für die verbreiteten Produkte, meist soziale Medien, zu. Des Weiteren ändern sich Oberflächen von Webclients häufig. Die Suche nach Artefakten wird erschwert, da forensische Software den Quelltext der Webanwendung an vorgegebenen Stellen nach Artefakten durchsucht.

Weiterhin besteht die Möglichkeit, per Beschluss eine Kopie der Daten beim Cloudanbieter zu beantragen. Juristische Aspekte spielen hierbei eine Rolle, da die Daten nicht zwingend in Rechenzentren innerhalb der EU vorliegen.

Hannan Bin Azhar und Bate (2019) beschreiben eine Methode, Clouddaten über eine inoffizielle API herunterzuladen. In diesem Fall wird eine API für Amazon Echo genutzt. Andere Forschende beschreiben ein ähnliches Vorgehen mithilfe der Google API (Awasthi et al., 2018). Da es sich um inoffizielle APIs handelt ist fraglich, ob die Ergebnisse vor Gericht als verwertbar angesehen werden.

Eine physikalische Kopie der Datenspeicher ist schwer zu erstellen. Clouddaten liegen meist in virtualisierten Speichern vor, daher muss die Erstellung eines Images durch den Cloudanbieter erfolgen, der auf die Hostebene zugreifen kann (Ruan et al., 2011). Im Kapitel Ausblick wird hierfür eine organisatorische Lösung vorgestellt.

Unabhängig von der Art der Datensicherung stellen Simou et al. (2014) ein Modell vor, welches die Phasen der Cloudsicherung beschreibt. Der Prozess besteht aus den Schritten Identifikation, Konservierung, Sammlung, Untersuchung, Präsentation.

3.3.6 Geräteuntersuchung

Auch Smart Home Geräte selbst können Artefaktequellen sein. Gelingt es, den Gerätespeicher auszulesen, so können wertvolle Informationen gewonnen werden. Neben der Firmware, die als Betriebssystem eines Gerätes agiert, werden auch Benutzerinformationen und Gerätekonfigurationen im Gerät vorgehalten. Um diese Artefakte sammeln und auswerten zu können, müssen neue Herausforderungen überwunden werden.

Verglichen mit der klassischen IT-Forensik ist das Auslesen des Speicherchips eine komplexere Aufgabe. Während in der klassischen IT-Forensik Standardschnittstellen von Speichersystemen wie SATA (Serial Advanced Technology Attachment) oder M.2 genutzt werden können, um die Daten auszulesen, müssen Chips aus Smart Home Geräten mittels serieller Schnittstellen ausgelesen werden. Die Arbeit an Smart Home Geräten ist daher deutlich näher an der Hardware (Al-Masri et al., 2018). IT-Forensiker*innen müssen Fähigkeiten im Bereich Elektrotechnik und Elektronik erlernen, um forensische Analysen vornehmen zu können. Zudem muss eine Zerstörungsfreigabe eingeholt werden, bevor das Gerät geöffnet wird, da die Hardwareanalyse Beschädigungen nach sich ziehen kann. Auch im Bereich der Geräteanalyse wird die forensische Arbeit durch die Verwendung von Verschlüsselung in Firmware und Speichersystemen erschwert (Sadineni et al., 2019). Eine Analyse des Arbeitsspeichers ist nahezu unmöglich, da das Gerät im laufenden Betrieb geöffnet und auf die Schnittstellen zwischen Prozessor und RAM, oder die Verbindung zugegriffen werden müsste. Allerdings werden je nach Firmware Swapfiles und Hibernation Files angelegt. Diese können, sofern es gelingt auf das Dateisystem zuzugreifen, ausgelesen werden (Hildayanti & Riadi, Mai 2019).

Grundsätzlich kann die Firmware entweder auf einem eigenständigen Flash-Speicher (NOR, NAND oder eMMC) gespeichert sein, oder im SoC (System on a Chip) vorliegen. SoC bezeichnet Mikrochips, die unterschiedliche Systemaufgaben vereinen, beispielsweise Speicher-, Arbeitsspeicher-, Prozessor-, Zeitgeber- und Grafiksysteme (Gupta, 2019).

Vor der Geräteuntersuchung gilt es zu ermitteln, welches Speichersystem genutzt wird (Boztas et al., 2015). Hierzu können die Informationen aus der OSINT-Recherche verwendet werden. Zudem kann, nachdem das Gerät geöffnet und ein Zugriff auf die Platine ermöglicht wird, die Bezeichnung der Chips abgelesen werden. Diese Bezeichnung wird zur weiteren Recherche von Datenblättern und Untersuchungsanleitungen verwendet.

Die Software eines IoT-Gerätes besteht aus unterschiedlichen Komponenten. Auf der Firmware setzt die Middleware auf, die die Betriebssystemebene für laufende Anwendungen abstrahiert. Hierauf baut die Applikationsebene auf, in der die eigentlichen Anwendungen betrieben werden. Gegebenenfalls werden benötigte Dateien in einem File System abgelegt (Li et al., 2015). Um die Daten aus Flashspeichersystemen oder SoC-Systemen auszulesen, werden Kommunikationsschnittstellen der Chips genutzt.

Netzwerkkommunikation

Manche Systemchips bieten einen Netzwerkzugriff, beispielsweise über SSH oder Telnet. Voraussetzung hierfür ist eine Verbindung zum Chip, die entweder drahtlos über einen angeschlossenen WiFi-Chip oder über einen angeschlossenen Ethernet Controller aufgebaut wird (raspberrypi.com, 2020).

Serielle Protokolle

In einer seriellen Verbindung werden, im Gegensatz zur parallelen Verbindung, Daten Bit für Bit nacheinander über die Schnittstelle ausgetauscht. Sende- und Empfangsschnittstelle sind voneinander getrennt. Eine weit verbreitete serielle Schnittstelle für Microchipsysteme ist UART (Universal Asynchronous Receiver Transmitter). UART ist ein asynchrones Protokoll zur seriellen Verbindung zweier Geräte. Ein entscheidender Vorteil ist, dass für die Kommunikation kein Zeitgeber benötigt wird. UART wurde entwickelt, um eine direkte Verbindung zu Microchips aufzubauen (Gupta, 2019). Ziel innerhalb der forensischen Analyse ist es, den Bootprozess mit angeschlossener Konsole zu unterbrechen. Die Firmware stellt dann in vielen Fällen eine Shell zur Verfügung, über welche Kommandos auf dem Gerät ausgeführt werden können. Anschließend wird versucht mithilfe der

Bordmittel dieser Kommandozeile ein Image des Datenchips zu erstellen (Sevida & Casey, 2019). Weitere verbreitete Vertreter serieller Protokolle sind SPI und I2C, auf die an dieser Stelle nicht näher eingegangen wird.

JTAG

Der von der Joint Test Action Group (JTAG) entwickelte Standard IEEE 1149.1 wurde ursprünglich nicht als Kommunikationsprotokoll entwickelt. Vielmehr können mithilfe des Standards Debugging von Schaltkreisen sowie weitere Testszenarien durchgeführt werden (Gupta, 2019). Neue Funktionstests für Schaltkreise wurden in den 1990er Jahren notwendig, da die bisher genutzten Nagelbett-Tests aufgrund der Größenreduzierung elektronischer Bauteile zu viel Platz in Anspruch nahmen (Vishwakarma und Lee, 2018).

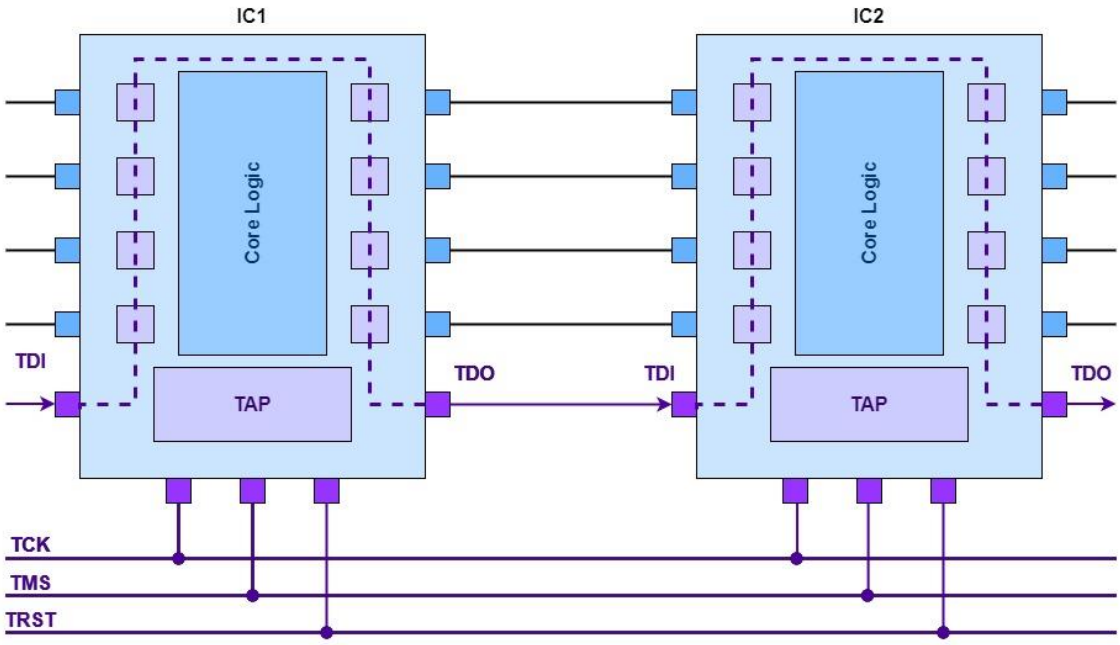


Abbildung 11 - JTAG Boundary Scan (angelehnt an xJTAG.com, 2022)

Abbildung 11 zeigt schematisch den Ablauf beim Boundary Scan. Die Testlogik ist in den Schaltkreis eingebaut. Die Aufgabe des TAP-Controllers ist es, die einzelnen Scan Cells in einen definierten Zustand zu setzen. Er erhält über drei PINs Signale (xjtag.com, 2022):

- TCK: Der Zeitgeber gibt das Taktsignal an, die Testausführung wird synchronisiert
- TMS: Über den Test Mode Select wird der Modus ausgewählt, in welchem das Bauteil getestet werden soll
- TRST: Über den optionalen Test Reset kann der TAP-Controller zurückgesetzt werden.

Nun wird ein Datenstrom durch die Scan Cells des Chips gesendet, dessen Änderung gemessen wird.

- TDI: Hier wird ein definierter Datenstrom in den Schaltkreis eingegeben. Dieser durchläuft die Scan Cells, welche sich in einem vorher definierten Zustand befinden.
- TDO: Der Datenstrom kann, nachdem er den zu testenden Schaltkreis durchlaufen hat, ausgelesen werden. So kann überprüft werden, ob sich der Datenstrom auf die erwartete Art ändert. Ist dies der Fall, dann gilt das Bauteil als funktionsfähig.

Mithilfe des Tests lassen sich nicht nur Kurzschlüsse finden, sondern auch die Funktion einzelner elektronischer Bauteile (beispielsweise Pull Up Widerstände oder Gatterschaltungen) überprüfen. Wie in Abbildung 11 gezeigt, können auch mehrere Schaltkreise in Reihe getestet werden.

Um mit JTAG auf den Inhalt eines Flashchips zuzugreifen, muss ein Kommunikationsprotokoll verwendet werden, welches auf der JTAG-Verbindung aufbaut. Ein Beispiel hierfür ist Open OCD (On Chip Debug) (openocd.org, 2022). Auch proprietäre Protokolle einzelner Herstellerfirmen können verwendet werden. Zum Auslesen von Speicherchips ist auf diese Weise der Zugriff auf die CPU mittels JTAG möglich. Es kann eine Shell geöffnet werden, mithilfe derer auf das Dateisystem zugegriffen werden kann.

Da JTAG keine standardisierte Schnittstelle ist, ist das Auffinden der richtigen Pins unter Umständen eine zeitaufwändige Aufgabe. Als Hilfestellung können Datenblätter der Chips dienen, die die JTAG-Pins auflisten. Zudem können bei

Onlinerecherchen (z.B. FCC.io, siehe Kapitel OSINT-Analyse) Informationen über JTAG-Schnittstellen gefunden werden.

Eine weitere Methode ist die Verwendung spezieller Hardware zum Auffinden von JTAG-Pins.

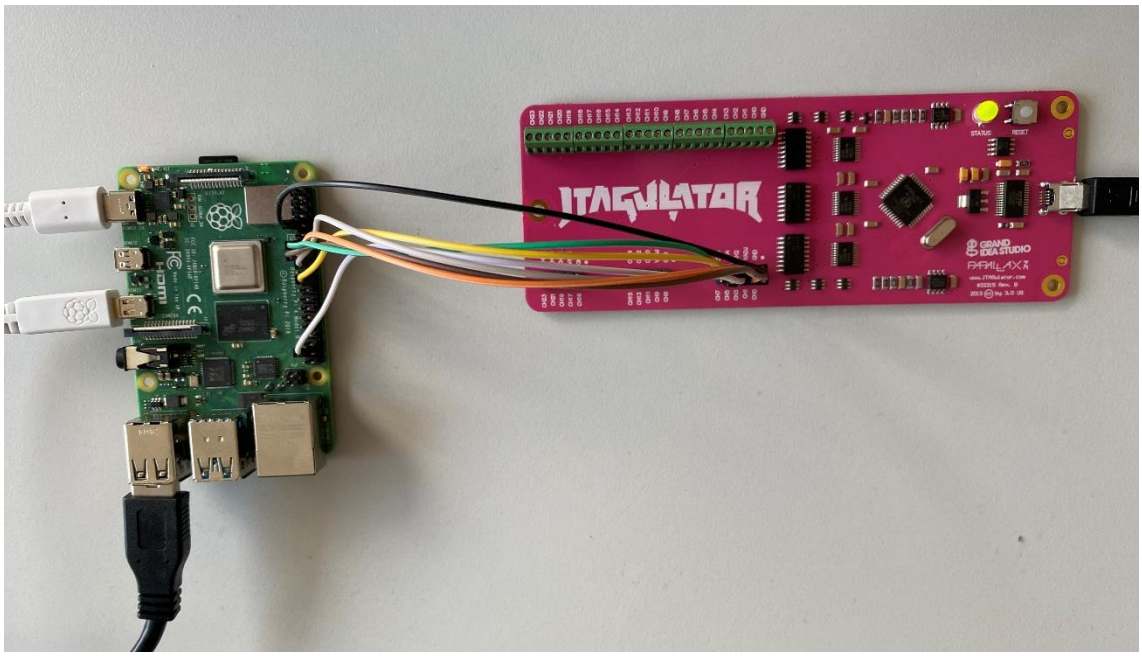


Abbildung 12 - JTAGulator angeschlossen an Raspberry Pi 4

Abbildung 12 zeigt eine solche Hardware, angeschlossen an einen Raspberry Pi 4. Es handelt sich um den JTAGulator der Grand Idea Studio, Inc. Bis zu 24 Pins eines untersuchten Gerätes können an den JTAGulator angeschlossen werden, um zu ermitteln, ob sie zu einem JTAG-Interface gehören. Auch die Rolle der Pins (TDI, TDO, TCK, TMS, TRST) kann mit dem JTAGulator ermittelt werden. Zusätzlich ist der JTAGulator in der Lage, das Pinout für UART zu finden (grandideastudio.com, 2022). Der JTAGulator wird mittels USB mit einem PC verbunden und die Kommunikation mit dem Terminalprogramm screen gestartet:

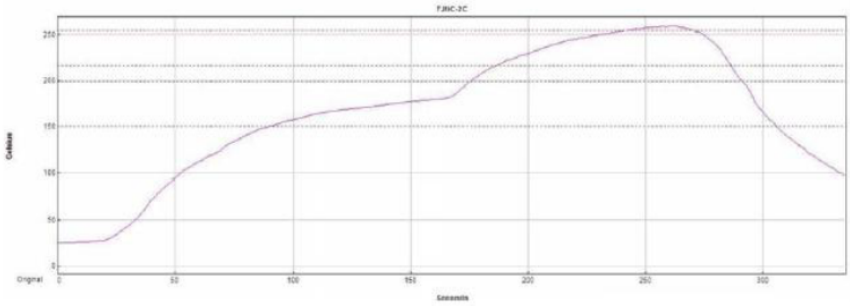
screen /dev/ttyUSB0 115200

Hierbei ist 115200 die Baudrate des JTAGulators. Anschließend werden die Parameter eingegeben und die Suche gestartet. Werden JTAG-Pins gefunden, dann wird die Position ausgegeben. Der JTAGulator kann nicht nur zum Auffinden der JTAG-Schnittstelle verwendet werden, sondern auch zum

Debugging mit Open OCD. Voraussetzung ist, dass für die untersuchte Hardware ein Open OCD Profil existiert. Gegebenenfalls kann über die Debugging-Schnittstelle eine Kommandozeile auf dem untersuchten Gerät aufgerufen werden. Über diese Kommandozeile kann anschließend ein Imaging- oder Datendownloadprozess gestartet werden (embeddedbits.org, 2021). Grundsätzlich sind zur Erstellung von Datensicherungen Bordmittel des Betriebssystems auf dem Asservat zu bevorzugen (Sadineni et al., 2019).

Chip-Off

Als Chip-Off bezeichnet man das thermische Auslöten oder das mechanische Abfräsen eines Speicherchips aus einer Platine (ultratecusa.com, 2020). Bei den thermischen Methoden wird die Platine erhitzt, bis das Lötzinn flüssig wird.



(Figure 6-1 Recommended Reflow Profile)

(Table 6-1 Reflow Profile Condition)

Max rising slope	2.59	73%
Max falling slope	-3.69	-8%
Preheat 150 – 200C	85.65	-57%
Reflow time /255C	29.24	-8%
Peak temperature	259.66	86%
Total time /217	99.81	-12%

Abbildung 13 - Temperaturprofil Anyka AK3918 – Quelle (Anyka, 2022)

Abbildung 13 zeigt das Temperaturprofil zum Erhitzen des SoC-Chips Anyka AK3918, welches dem Datenblatt entnommen werden kann (Anyka, 2022). Mithilfe dieses Temperaturprofils kann in einer programmierbaren Chip-Off-

Maschine (beispielsweise die Ersa HR 550) der Prozess vorgeplant werden. Ist die Temperatur von 255 °C erreicht, wird der Chip entweder durch leichten Druck oder durch Ansaugen mittels Unterdruck von der Platine gelöst. Zu beachten ist hierbei, dass Chips, insbesondere in neueren Geräten, meist verklebt werden. Dieser Kleber muss nach dem Auslöten so rückstandsfrei wie möglich entfernt werden, um auf die Pins am Speicherchip zugreifen zu können. Anschließend wird der gelöste und gereinigte Chip in einen speziellen Adapter mit Federpins eingelegt und mit einem PC verbunden. So kann auf die Dateien innerhalb des Speicherchips zugegriffen werden. Das Fräsen einer Platine an der gegenüberliegenden Seite des Chips ermöglicht ebenfalls den Zugriff auf die Pins. Beim Entfernen des Speicherchips wird das untersuchte Gerät endgültig zerstört. Daher ist Chip-Off als letzte Methode in einer Untersuchung anzusehen. Es muss vorab geprüft werden, ob ein Datenzugriff über eine andere Schnittstelle möglich ist.

4 Untersuchung des HAMA Smart Home Ökosystems

Die in Kapitel Methoden beschriebenen Vorgehensweisen zum Auffinden und Untersuchen von Smart Home Systemen werden im folgenden Kapitel praktisch angewendet. Sie werden in mehreren Versuchsaufbauten anhand des Smart Home Ökosystems der Herstellerfirma Hama erläutert. Auf eine Beschreibung des Systems folgt die Versuchsreihe zum Auffinden von Geräten am Tatort. Anschließend wird die fünfstufige Analyse, bestehend aus OSINT-, Netzwerk-, App-, Cloud- und Geräteanalyse erläutert. Die Ergebnisse werden zum Abschluss vergleichend dargestellt.

4.1 Auswahl des Smart Home Ökosystems

Zur Auswahl eines Smart Home Ökosystems für die Versuchsreihe wurden folgende Kriterien angesetzt:

- Zum System wurde bisher keine forensische Forschungsarbeit geleistet.
- Das System ist weit verbreitet.
- Das System beinhaltet Komponenten unterschiedlicher Smart Home Bereiche.
- Das System ist mittels App steuerbar.

Zum Hama Smart Home Ökosystem konnten keine Quellen gefunden werden, welche die forensische Analyse beschreiben. Zudem wird das System bei gängigen Elektronik-Händlern zum Kauf angeboten. Auch die Nutzerbewertungen wurden als Kriterium für die Verbreitung berücksichtigt. Auf der Seite von Hama wird das System in die Bereiche Licht, Smarte Steckdosen, Sicherheit sowie Heizung und Klima unterteilt (de.hama.com/produkte/smart-home, 2022). Außerdem ist es über eine herstellereigene App und die Google Home App steuerbar. Das System erfüllt folglich alle oben genannten Voraussetzungen.

4.2 Beschreibung der Komponenten

4.2.1 App „HamaSmartHome“

Die App „HamaSmartHome“ spielt eine zentrale Rolle im Hama Smart Home Ökosystem. Die App stellt alle Funktionen zur Geräteverwaltung zur Verfügung. Das Einbinden von Geräten funktioniert mittels Suchfunktion. Hierzu muss die Smart Home Komponente in einen von zwei Kopplungsmodi versetzt werden. Standardmäßig wird die Kopplung mittels EZ-Modus durchgeführt. Hierbei wird ein UDP-Paket an eine Broadcast-Adresse versendet, welches die benötigten Netzwerkinformationen enthält. Dieses Paket wird vom Gerät empfangen und entschlüsselt. Das Gerät kann sich anschließend selbstständig im WLAN anmelden. Der zweite Kopplungsmodus ist der AP-Modus. In diesem Fall öffnet das Smart Home Gerät einen WLAN-Accesspoint, mit welchem sich das Smartphone verbindet, um die Netzwerkparameter direkt zu übergeben (support.tuya.com, 2022).

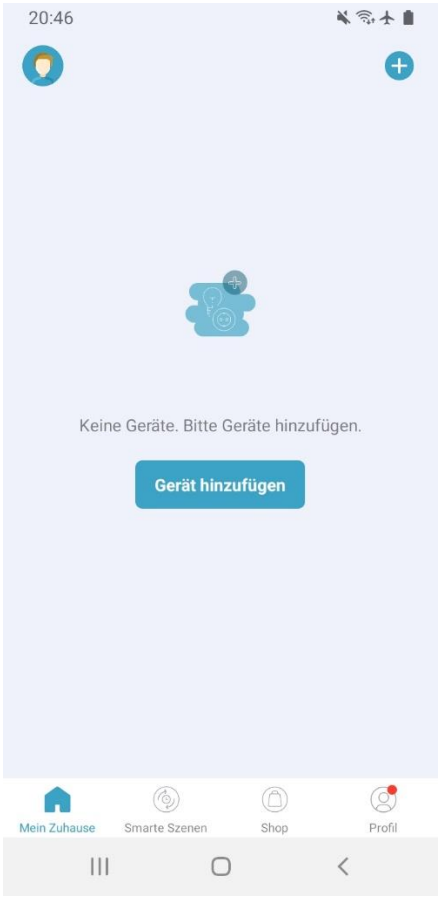


Abbildung 14 – Screenshot: Startbildschirm Hama Smart App

Neben der Geräteverwaltung wird, wie in Abbildung 14 gezeigt, auch die Benutzerverwaltung und die Automation von Zeitplänen und Szenen mithilfe der App organisiert (de.hama.com/produkte/smart-home, 2022).

Zur Durchführung der Versuche wurde die App auf ein Samsung Galaxy A6 (SM-A600FN/DS) mit dem Betriebssystem Android 10 installiert. Anschließend wurde ein Gmail-Account angelegt und dieser in der App registriert. Des Weiteren wurde die Option zugekauft, den Kamerastream für einen Monat in der Cloud zu speichern.

4.2.2 WiFi-Outdoor-Kamera

Die IP-Kamera „WiFi-Outdoor-Kamera“ aus dem Sicherheitssortiment der Hama Smart Home Solution wurde zum Einsatz im Außenbereich konzipiert. Die Kamera wird mit einem Netzteil inklusive Kabel sowie Montagezubehör geliefert. Sie verfügt über ein Mikrofon zur Tonaufnahme, einen Lautsprecher zur Sprachwiedergabe und einen SD-Kartenslot zur Speicherung von Videodateien. In der App können Zusatzfunktionen aktiviert werden. Hierzu gehören die Bewegungs- und Geräuschmelderfunktion. Für beide Funktionen kann ein Alarm aktiviert werden, der eine Benachrichtigung auf das Smartphone schickt. Der Alarm kann auch in eine Automation programmiert werden, sodass beispielsweise eine smarte LED des Ökosystems aktiviert wird. Eine weitere Funktion ist die Sicherung von Videodateien in einem Cloudspeicher. Das kostenpflichtige Produkt wird mit unterschiedlichen Laufzeiten angeboten. Es wird der gesamte Videostream abgespeichert, außerdem werden ausgelöste Alarme im Stream markiert.

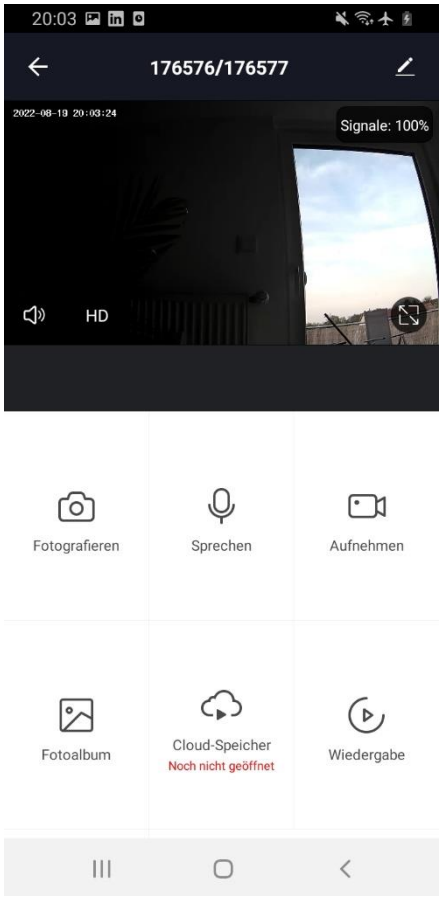


Abbildung 15 – Screenshot: Kameramenü Hama Smart App

Abbildung 15 zeigt das Kameramenü. Hier kann der Livestream beobachtet werden. Zudem besteht die Möglichkeit Aufnahmen zu erstellen oder durch das Mikrofon der Kamera zu sprechen (de.hama.com/produkte/smart-home, 2022).

4.2.3 WiFi-Tür- / Fenster-Kontakt

Der smarte Türkontaktsensor von Hama gehört ebenfalls zum Sicherheitssortiment. Das batteriebetriebene Gerät besteht aus zwei Komponenten, einem Magneten und einem Magnetfeldsensor. Der Magnet wird am Türrahmen angebracht, der Magnetfeldsensor an der Tür. Der Sensor nimmt die Änderung des Öffnungszustandes wahr. In der App kann für die Zustandsänderung ein Alarm eingestellt werden, es folgt eine Benachrichtigung auf das Smartphone (de.hama.com/produkte/smart-home, 2022).

4.2.4 WiFi-Heizungssteuerung

Die smarte Heizungssteuerung von Hama besteht aus einer kabelgebundenen Zentrale, an die mehrere batteriebetriebene Heizungsthermostate angeschlossen werden können. Diese übernehmen die Funktion von Temperatursensoren sowie die Aktorfunktion als Heizungsthermostat. Die Zentrale wird wie alle anderen Geräte des Ökosystems per WLAN gesteuert. Die Verbindung zwischen Zentrale und Thermostaten hingegen wird mit ZigBee aufgebaut (de.hama.com/produkte/smart-home, 2022).

4.2.5 WLAN-Steckdose „Mini“

Der WLAN-Steckdosensockel des Hama Smart Home Ökosystems wird in eine 230V-Schutzkontaktsteckdose eingesteckt und kann den Schaltzustand angeschlossener Geräte verändern. Die Schaltung funktioniert mittels Relay, welches über WLAN per App oder manuell auf Knopfdruck angesteuert wird (de.hama.com/produkte/smart-home, 2022).

4.2.6 WLAN-LED

Die ausgewählte WLAN-LED-Lampe besitzt einen GU 10 Lampensockel und kann Licht im gesamten RGBW-Farbspektrum emittieren. Zudem verfügt sie über eine Ein- und Ausschaltfunktion (de.hama.com/produkte/smart-home, 2022).

4.2.7 Router Technicolor CGA6444VF Vodafone

In der ersten Versuchsreihe zum Auffinden von Smart Home Geräten nahm der Router Technicolor CGA6444VF die Rolle des zentralen WLAN-Routers ein. Das Gerät wird von der Vodafone GmbH vertrieben und ist WLAN 6-fähig. Es baut WLAN-Netze sowohl im 2,4 als auch im 5 GHz-Spektrum auf. WAN-seitig ist der Router per Koax-Breitbandkabel angeschlossen. Er verfügt über vier LAN-Dosen und einen Telefonanschluss (Technicolor, 2019).

4.2.8 Virtual Box mit Ubuntu

Als Untersuchungsrechner wurde für die nachfolgenden Versuche das Notebook Lenovo Legion 5 (Modellnummer 15ARH05H) genutzt. Auf dem Notebook war die Virtualisierungssoftware Virtual Box (Version 6.1.16 r140961) installiert. Die vorgestellten Softwarewerkzeuge wurden im virtualisierten Betriebssystem Ubuntu (Version 22.4.1 LTS) verwendet. Die Nutzung einer Virtualisierungssoftware bietet Vorteile. Testumgebungen können mit einfachen Mitteln aufgebaut und exportiert werden, USB-Geräte sind leicht einzubinden. Zudem können mehrere virtuelle Umgebungen mit unterschiedlichen Aufgaben zeitgleich auf demselben Untersuchungsrechner genutzt werden.

4.2.9 USB-WLAN-Dongle

Zur Durchführung der WLAN-Scans und zum Aufbau eines Access-Points wurde der USB-WLAN-Dongle CSL USB 2.0 WLAN-Adapter genutzt. Der Dongle besteht aus dem WLAN-Chipsatz Ralink RT5572 sowie zwei Antennen. Er unterstützt die Standards 802.11a/b/g/n sowie alle gängigen Verschlüsselungsmechanismen. Er erreicht eine Datenübertragungsrate von bis zu 300 Mbit/s.

4.2.10 Texas Instruments CC2531 USB-Dongle

Zur Aufklärung des Funkstandards IEEE 802.15.4 wurde der USB-Dongle CC2531 von Texas Instruments genutzt. Dieser wurde mit einer Firmware zum Einsatz als Sniffer ausgeliefert. Die Firmware kann auf der Herstellerseite heruntergeladen und eigenständig in den Flashspeicher geladen werden (Texas Instruments, 2022).

4.3 Auffinden von Geräten – Hama Smart Home

In der ersten Versuchsreihe wurde ein realitätsnaher Tatort nachgestellt. Es handelte sich um eine Zwei-Zimmer-Wohnung mit insgesamt 60qm. Ziel der Untersuchung war es, Geräte anhand der Signalstärke ihrer Funkabstrahlung zu lokalisieren. Als weiteres Ziel wurde ein Praxisvergleich zwischen den

unterschiedlichen Recon-Tools durchgeführt. Es zeigte sich aber bereits nach den ersten Tests, dass sich die Stärken der jeweiligen Tools am besten nutzen lassen, wenn man sie in Kombination verwendet.

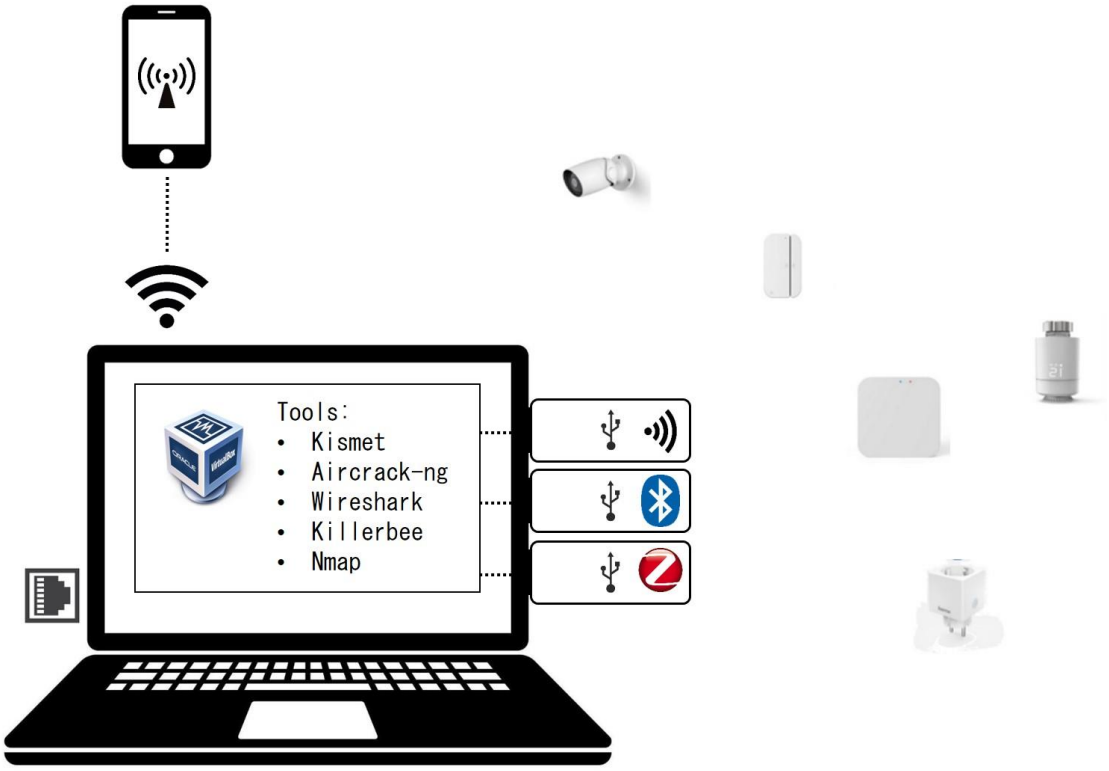


Abbildung 16 - Untersuchungsrechner am Tatort

Abbildung 16 zeigt schematisch den Untersuchungsrechner für die Arbeit am Tatort. Alle Netzwerkinterfaces werden an die virtuelle Maschine weitergegeben. Zudem wird eine mobile Internetverbindung aufgebaut.

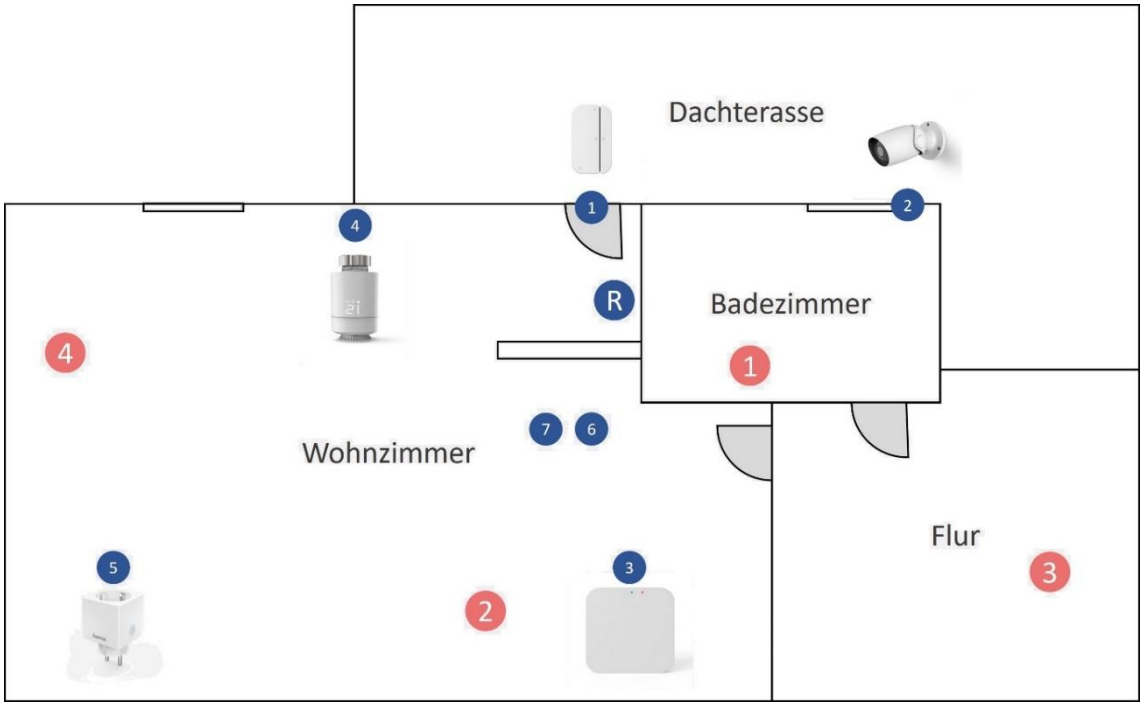


Abbildung 17 – Position der Smart Home Komponenten (Quelle der Einzelbilder: https://de.hama.com/produkte/smart-home)

Die Smart-Home-Komponenten wurden wie in Abbildung 17 dargestellt in der Wohnung verteilt.

Nummer Abbildung 17	Gerät	MAC-Adresse	MAC-Vendor-Lookup
1	Türkontaktsensor	E8:68:E7:XX:XX:XX	Espressif Inc.
2	Kamera	84:7A:B6:XX:XX:XX	AltoBeam (China) Inc.
3	Heizungssteuerung	84:E3:42:XX:XX:XX	Tuya Smart Inc.
4	Thermostat	-	-
5	Steckdosensockel	3C:61:05:XX:XX:XX	Espressif Inc.

6	iPhone	5A:8D:8F:XX:XX:XX	Not Found (Random MAC)
7	Laptop	E0:D4:E8:XX:XX:XX	Intel Corporate
R	Router	EC:A8:1F:XX:XX:XX	Technicolor CH USA Inc.

Tabelle 1 – Smart Home Komponenten

Die Herstellerfirmen der Geräte wurden anhand der MAC-Adresse zugeordnet (macvendor.info, 2022). Anschließend wurden an den rot markierten Messstellen die Signalstärken von WiFi- und ZigBee-Geräten in der Umgebung aufgezeichnet.

Für die Messungen wurde am Untersuchungsrechner der WLAN- und der ZigBee-Dongle angeschlossen und an die virtuelle Maschine übergeben. Im Anschluss wurde das Tool Kismet gestartet, um einen ersten Überblick des Funkverkehrs in der Wohnung zu erhalten. Der Vorteil von Kismet für diese Versuchsreihe ist die übersichtliche Oberfläche, die Pakete unterschiedlicher Protokolle in einer Tabelle auflistet. Diese Liste dient als Entscheidungshilfe für weitere Aufklärungsmaßnahmen. Kismet wurde mit dem Befehl *sudo kismet* gestartet, im Anschluss wurde die Adresse *localhost:2501* im Browser aufgerufen.

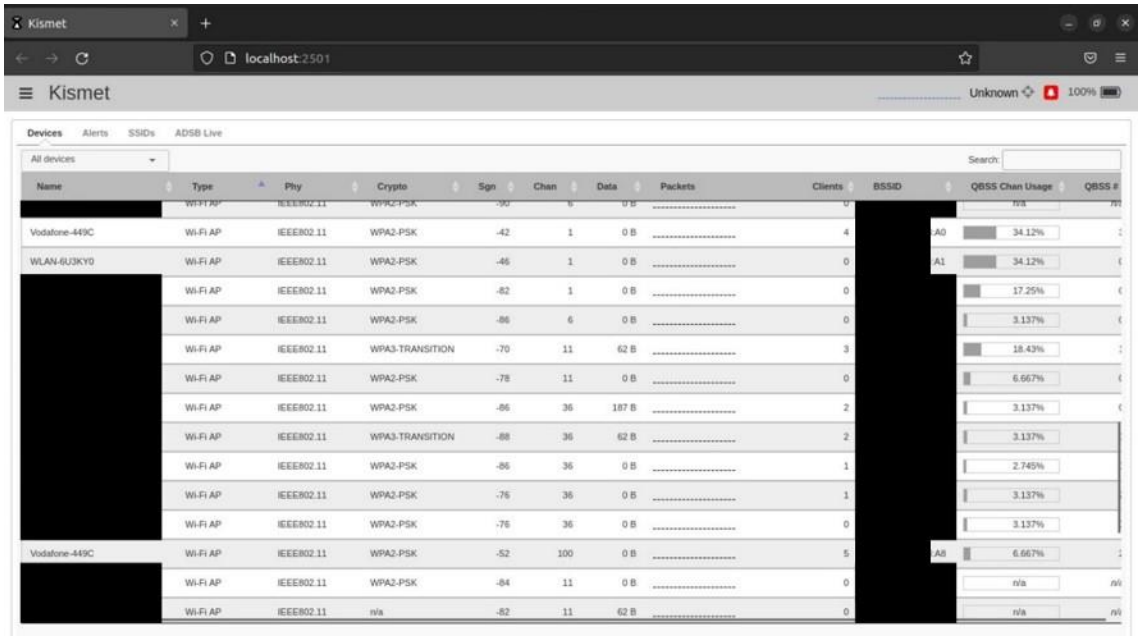


Abbildung 18 – Screenshot: WLAN-Scan mit Kismet

Abbildung 18 zeigt den vom WLAN-Dongle aufgezeichneten Funkverkehr an Messpunkt 1 in der Kismet-Weboberfläche. Es wird deutlich erkennbar, wie WLAN-Accesspoints aus umliegenden Haushalten die Analyse beeinträchtigten (die Accesspoints wurden geschwärzt). Ebenfalls erkennbar war eine deutlich höhere Signalstärke der WLAN-Accesspoints „Vodafone-449C“ sowie WLAN-6U3KYO. Zudem wird ersichtlich, dass der Accesspoint „Vodafone-449C“ sowohl auf 2,4 GHz (Kanal 1), als auch auf 5 GHz (Kanal 100) betrieben wurde. Die Ähnlichkeit der MAC-Adressen der genannten Accesspoints ließ einen Rückschluss darauf geben, dass ein Gastnetzwerk auf der gleichen Hardware betrieben wurde. Eine Abweichung in der Angabe von Signalstärken zeigte sich, da das Gastnetz eine Differenz von -4dBm aufwies. Diese Ungenauigkeit wurde in allen weiteren Versuchen festgestellt und dementsprechend berücksichtigt.

Mit den gesammelten Informationen wurde im weiteren Verlauf Airodump-ng genutzt, um die Hotspots näher zu analysieren. Hierzu wurde das Kommando mit den passenden Parametern aufgerufen, um nur Kanal 1 und Kanal 100 sowie ausschließlich die betreffenden Hotspots und daran angeschlossene Geräte zu überwachen.

*sudo airodump-ng -bssid ec:a8:*XX:XX:XX*:a0 -c 1 wlan0mon*

*sudo airodump-ng -bssid ec:a8:*XX:XX:XX*:a8 -c 100 wlan0mon*

Da der WLAN-Dongle über separate Antennen verfügt, konnte er beide Kanäle zur gleichen Zeit überwachen.

Abbildung 19 – Screenshot: WLAN-Scan mit airodump-ng Messpunkt 1

Abbildung 19 zeigt die Aufzeichnung des WLAN-Verkehrs an Messpunkt 1 mit airodump-ng. Auffällig ist die Änderung der Signalstärke um -6 dBm in beiden Kanälen. Dies ist als ist als Messfehler zu betrachten.

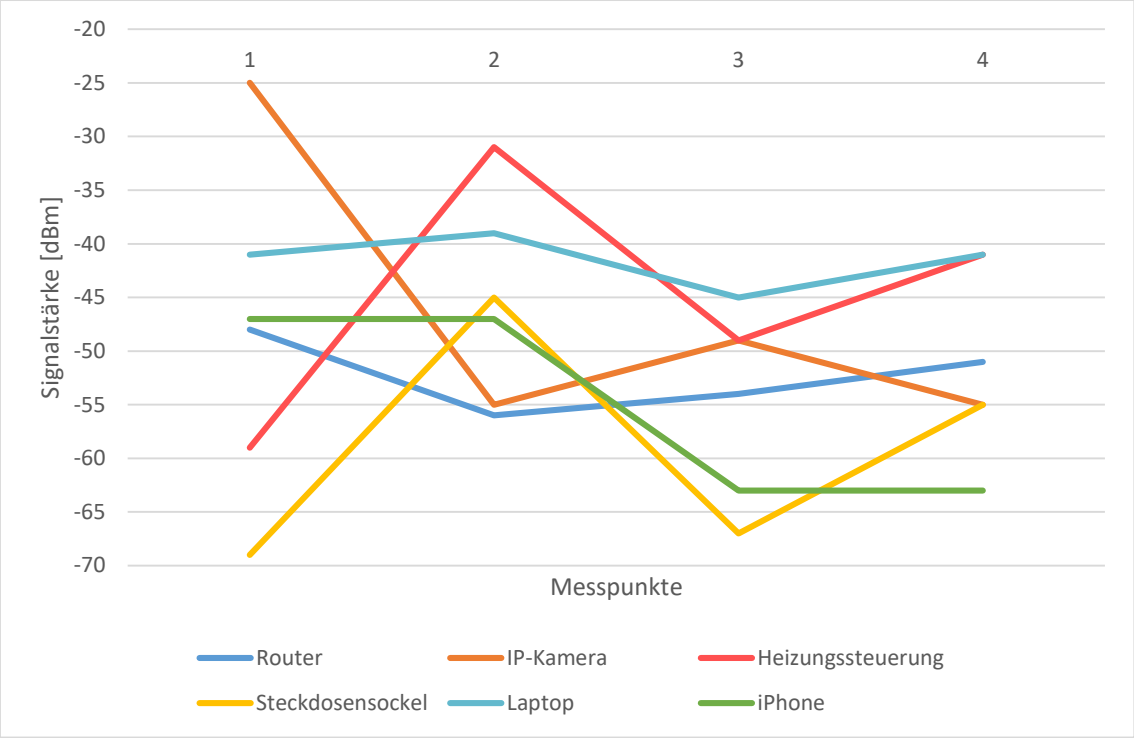


Abbildung 20 - Änderung der Signalstärken bei der WLAN-Aufklärung

Die Messungen wurden in ein Diagramm (Abbildung 20) übertragen. Im Anschluss wurde die Messung am Punkt 2 vorgenommen. Besonders hervorzuheben ist, dass sich bei dieser Messung die Signalstärke im Vergleich zum Messpunkt 1 im Kanal 1 stark ändert, während sie im Kanal 100 nahezu konstant bleibt. Am dritten Messpunkt ließen sich große Unterschiede in der Differenz der Signalstärke feststellen. So änderte sich die Signalstärke von Gerät 84:E3:42:XX:XX:XX (Heizungssteuerung) und 3C:61:05:XX:XX:XX (Smart Plug) um -18 und -22 dBm, während die Änderung bei allen weiteren Geräten deutlich geringer ausfiel. Obwohl iPhone und Laptop nebeneinander positioniert waren, änderten sich die Signalstärken unterschiedlich stark. Auch an Messpunkt 4 ist eine geringere Änderung am Router sowie am iPhone und am Laptop festzustellen.

In keiner der durchgeführten Messungen konnte die Signalstärke des Türsensors ermittelt werden. Dieser Umstand liegt im Netzwerkverhalten des Gerätes begründet. Um Energie zu sparen, wird der Sensor ausschließlich aktiv, wenn eine Änderung des Öffnungszustandes der Tür wahrgenommen wird. Smart Home Komponenten, die dieses Verhalten aufweisen, sind am Tatort mit

Netzwerkanalysemethoden schwer aufzufinden. Dennoch ist es nötig, Informationen über das Gerät zu erhalten. Zu diesem Zweck wurde in der vorliegenden Versuchsreihe eine weitere Messung durchgeführt. Hierbei wurde absichtlich eine Zustandsänderung ausgelöst.

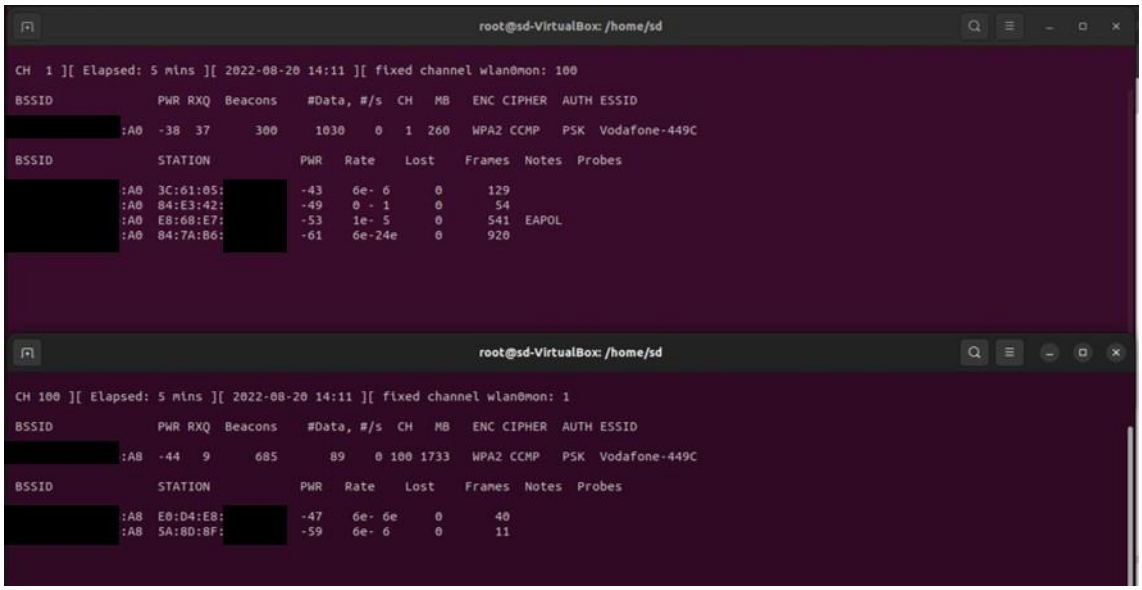


Abbildung 21 – Screenshot: WLAN-Scan mit airodump-ng - Türsensor

Abbildung 21 wurde während der Türöffnung aufgezeichnet. Ein weiteres Gerät, welches am Router angeschlossen war, wurde ermittelt. E8:68:E7:XX:XX:XX ist die MAC-Adresse des Türsensors.

Es konnte festgestellt werden, dass Geräte mit einer stärkeren Abstrahlleistung über Strecken weniger Signalstärke verlieren als Geräte mit einer schwachen Abstrahlleistung. Der Absolutwert der Signalstärke lässt somit keinen Rückschluss auf die Entfernung eines Gerätes zu. Ausschlaggebend für das Auffinden von Geräten ist die Änderung der Signalstärke. Diese korreliert mit der Entfernung.

Im Anschluss wurde die Funkaufklärung für ZigBee durchgeführt. Zu diesem Zweck wurde der ZigBee-Dongle von Texas Instruments an den Untersuchungsrechner angeschlossen und Kismet am Messpunkt 1 gestartet. Zunächst wurde über längeren Zeitraum kein ZigBee-Funkverkehr abgefangen.

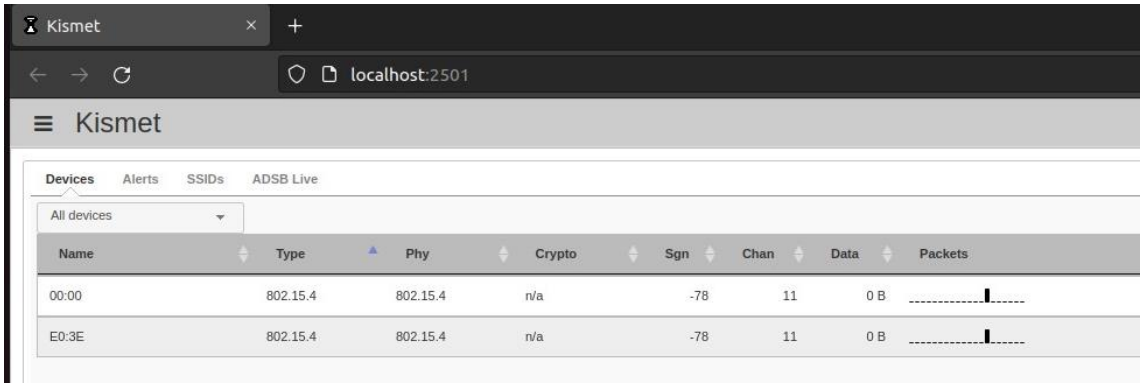


Abbildung 22 - Screenshot: ZigBee-Aufklärung mit Kismet

Erst nach 13 Minuten wurde ein Gerät erkannt (siehe Abbildung 22). Anhand der Adressierung 00:00 konnte ermittelt werden, dass es sich um eine Central handelte. Der lange Zeitraum lässt sich mit dem Standby-Modus erklären, in welchem sich die Heizungssteuerung und das Thermostat befanden. Um dieses Problem zu umgehen, wurde durch Drehen am Thermostat die Funkverbindung ausgelöst. Weiterhin wurde festgestellt, dass es sich bei den beiden angezeigten Geräten in ZigBee um die Heizungssteuerung handelte.

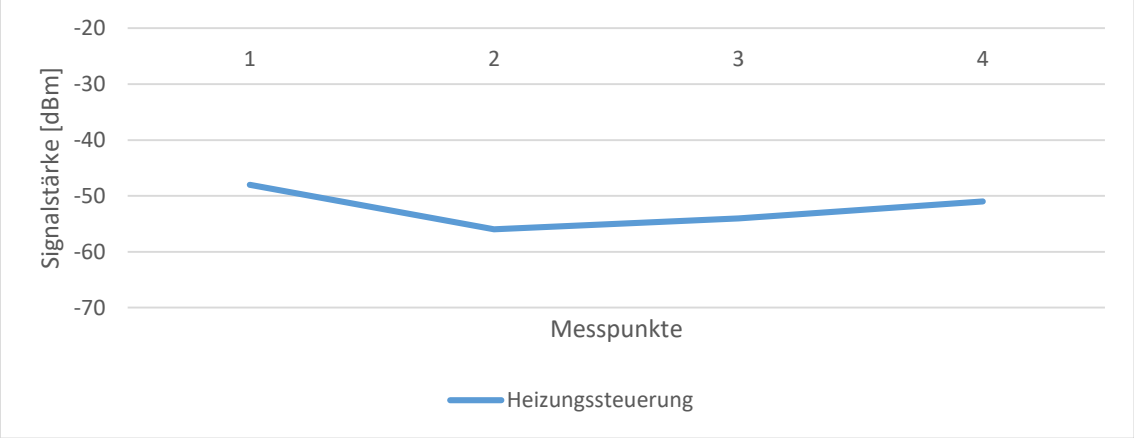


Abbildung 23 – Änderung der Signalstärke bei der ZigBee-Aufklärung

Abbildung 23 zeigt den Verlauf der Signalstärke an den verschiedenen Messpunkten. Es war festzustellen, dass die Signalstärke sich, verglichen mit der WLAN-Messung, nur schwach änderte.

4.4 Routeranalyse - Hama Smart Home

Im Anschluss an die Funkaufklärung wurde der Router von Technicolor analysiert. Zunächst wurde der Untersuchungsrechner mittels LAN-Kabel an den Router angeschlossen und die Weboberfläche über die IP 192.168.0.1 aufgerufen. Anschließend wurde das Passwort eingegeben und das Hauptmenü geöffnet.

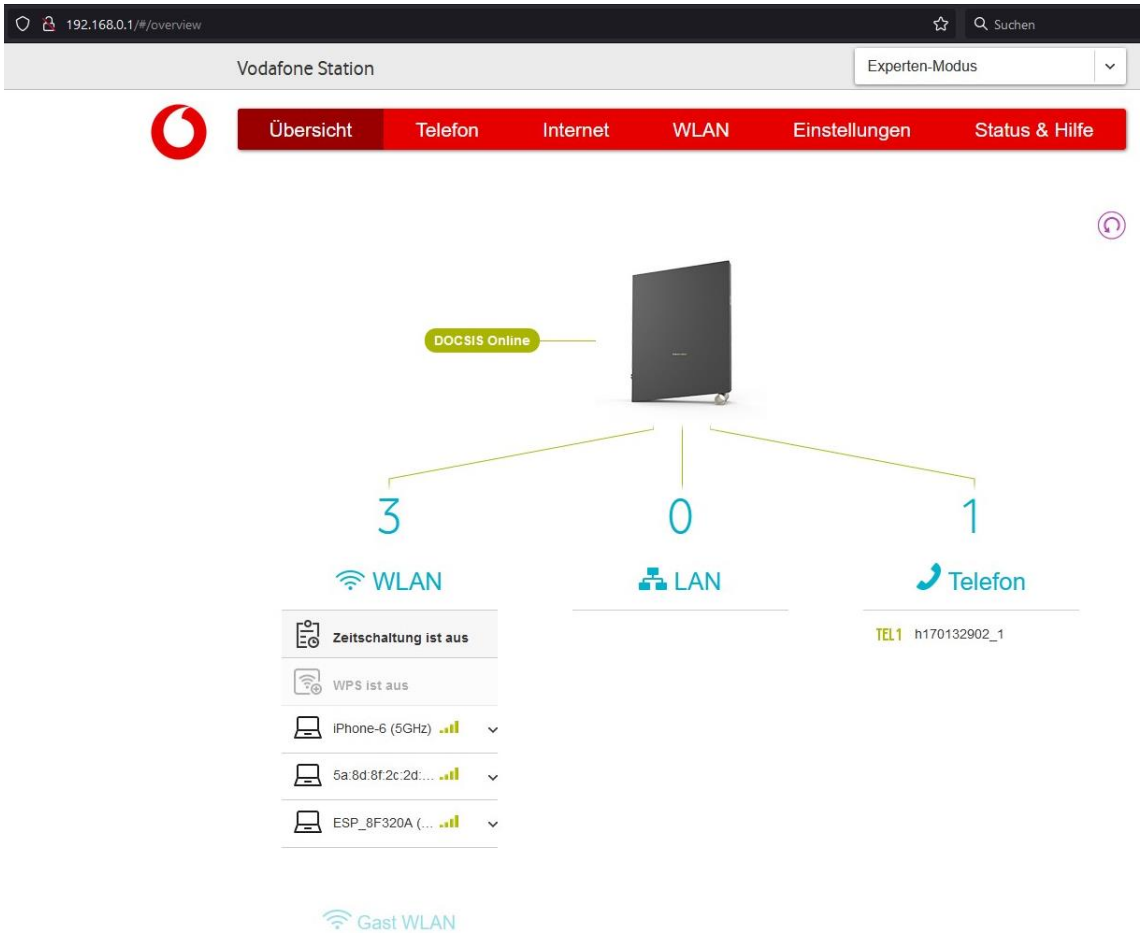


Abbildung 24 – Screenshot: Weboberfläche Router

Im Menü (Abbildung 24) wurde der Expertenmodus ausgewählt und die Firmwareversion (19.3B57-1.0.41) notiert. Anschließend wurden die momentan verbundenen Geräte betrachtet und deren MAC- sowie IP-Adressen und Gerätenamen dokumentiert. Im Folgenden wurde das WLAN-Menü aufgerufen. Hier konnte ermittelt werden, dass ein Gast-WLAN (SSID: WLAN-6U3KY0) betrieben wurde. Zudem wurde der Router sowohl auf dem 2,4 GHz als auch auf dem 5 GHz Band betrieben. Anschließend wurde das Ereignisprotokoll

aufgerufen und als ZIP-Datei heruntergeladen. Die Datei wurde genauer analysiert. Hierbei konnte festgestellt werden, dass Informationen zu an- und abgemeldeten Geräten geloggt wurden. Im nächsten Schritt wurde der Türsensor absichtlich ausgelöst, um zu ermitteln, ob ein Logeintrag genertiert wird. Erneut wurde das Ereignisprotokoll heruntergeladen und betrachtet.

2022-08-20 14:11:54 log.warn [88000102]: 13,2022-08-19 21:14:10 k1_SSID Device E8:68:E7:XX:XX:XX disconnected from SSID Device.WiFi.SSID.1 REASON-CODE=1.

2022-08-20 14:12:19 log.warn [88000102]: 14,2022-08-19 21:10:16 k1_SSID Device E8:68:E7:XX:XX:XX disconnected from SSID Device.WiFi.SSID.1 REASON-CODE=4.

Hierbei konnten die beiden obenstehenden Logeinträge zum Verbindungsabbau gefunden werden, jedoch keine Einträge zum Verbindungsaufbau.

Ein weiterer Versuch folgte am 20.09.2022. Der Türsensor wurde um 19:08:30 ausgelöst.

2022-09-14 21:44:38 log.warn [88000102]: 9,2022-09-05 19:08:36 k1_SSID Device E8:68:E7:XX:XX:XX disconnected from SSID Device.WiFi.SSID.1 REASON-CODE=1.

2022-09-20 19:06:59 log.warn [88000102]: 67,2022-08-30 08:30:32 k1_SSID Device E8:68:E7:XX:XX:XX disconnected from SSID Device.WiFi.SSID.1 REASON-CODE=4.

Die Analyse des Ereignisprotokolls ergab, zwei Logeinträge (siehe oben). Diese Logeinträge wurden mit dem Nachrichtencenter der Smartphone-App abgeglichen.

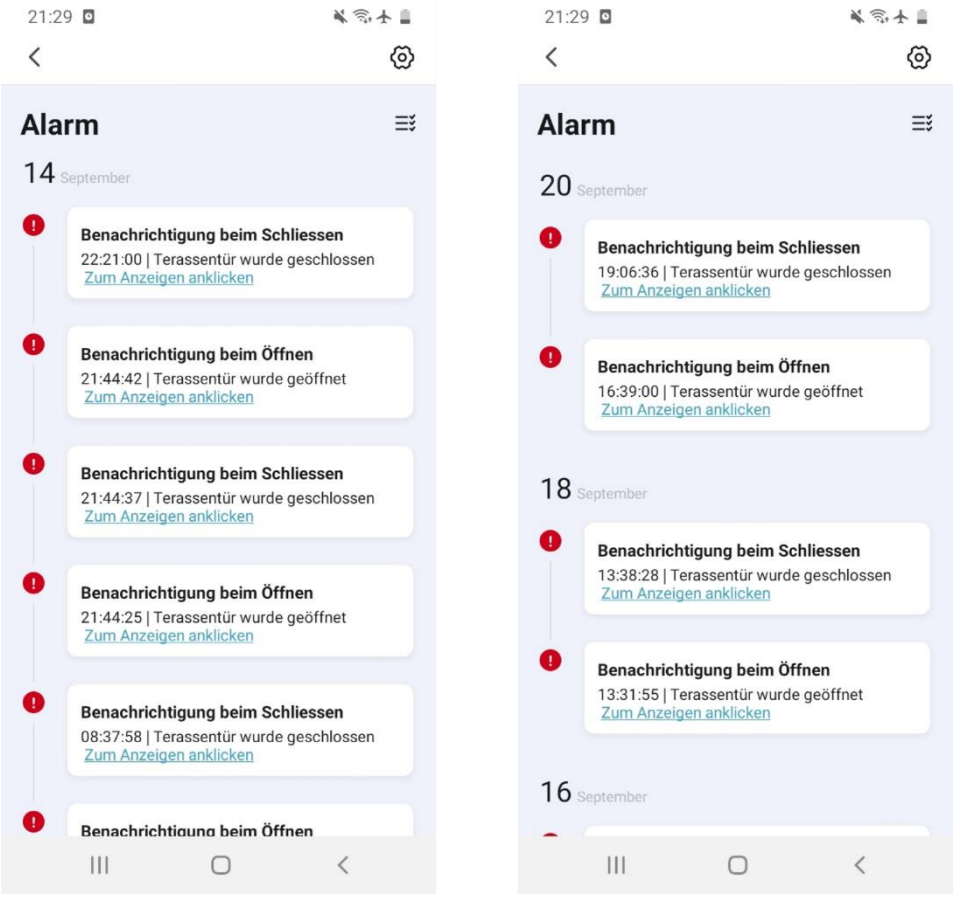


Abbildung 25 - Screenshots: Nachrichtencenter der Hama Smart App

Es wurde ersichtlich, dass zwar bei beiden Zeitstempeln der Türsensor ausgelöst wurde, aber zwischenzeitlich mehrere Mitteilungen gesendet wurden, die sich nicht im Protokoll des Routers widerspiegeln (Abbildung 25).

4.5 Netzwerkanalyse am Tatort

Die Netzwerkanalyse wurde direkt am Tatort gestartet. Nach Abschluss der Untersuchung vor Ort wurden mit Vergleichsgeräten in einer angepassten Laborumgebung weiterführende Analysen durchgeführt, um die Ergebnisse vom Tatort besser beschreiben zu können.

Für die Analyse am Tatort wurde weiter der Untersuchungsrechner zum Auffinden von Geräten verwendet. Die Komponenten mussten zunächst in ein Netzwerk eingebunden werden, welches den Router am Tatort simuliert. Hierzu wurde die Hotspot-Funktion von Ubuntu als simulierter Router verwendet. Damit die Geräte sich ohne weiteres mit diesem Netzwerk verbanden, musste die

gleiche SSID und das gleiche WPA2-Passwort verwendet werden. Dieses konnte wie in Kapitel Geräteanalyse - Hama Smart Home beschrieben aufgedeckt werden. Zu beachten war außerdem, dass die virtuelle Maschine eine Internetverbindung über den Untersuchungsrechner benötigte, um die Steuerungsserver des Smart Home Systems erreichen zu können. Der Untersuchungsrechner wurde zu diesem Zweck mit dem WLAN-Hotspot eines Smartphones verbunden, um die mobile Arbeit am Tatort zu simulieren. Nachdem die Geräte mit dem Hotspot verbunden wurden, wurde Wireshark gestartet und der Netzwerkverkehr der WLAN-Schnittstelle mitgeschnitten. Dieser Mitschnitt wurde später mit den Ergebnissen der Laboruntersuchung abgeglichen, um das Verhalten der einzelnen Geräte beschreiben zu können. Zum Abschluss wurde jedes Gerät einem Portscan unterzogen.

4.6 OSINT-Analyse - Hama Smart Home

4.6.1 Webseite der Herstellerfirma

Hier finden sich bereits die ersten Informationen zum System. Das System wird per App gesteuert. Daneben kann eine Steuerung auch über die Smart Assistants Google Home oder Amazon Alexa erfolgen. Diese Assistants bieten auch eine Sprachsteuerungsfunktion an. Zum Ökosystem gehören Geräte der Kategorien Licht, Strom, Sicherheit und Heizung. Das gesamte Ökosystem läuft über WLAN, einzige Ausnahme sind die Heizungssteuerungselemente, welche per ZigBee mit ihrer Zentrale kommunizieren. Der direkte Anschluss der Geräte an den Heimrouter wird beworben. In der Beschreibung der IP-Kamera auf der Webseite wird ersichtlich, dass die Streams entweder lokal auf einer Mikro-SD-Karte oder in der Cloud gesichert werden können. Der Gerätespeicher wird mit 128 GB angegeben. Es wird zunächst nicht ersichtlich, ob es sich um den internen Gerätespeicher oder um den Cloudspeicher handelt. Zudem wird die Nachtsichtfunktion und die Möglichkeit einen Bewegungs- und Geräuschalarm einzustellen beworben (de.hama.com/produkte/smart-home, 2022).

4.6.2 Bedienungsanleitungen

Auf der Webseite finden sich die Bedienungsanleitungen. Hier werden die Gerätefunktionen, z.B. das Einbinden der Geräte in ein neues Netzwerk und in die App, erklärt. Es gibt zwei unterschiedliche Kopplungsmodi, den EZ- und den AP-Modus. Im EZ-Modus übergibt das Smartphone die benötigten WLAN-Informationen direkt Peer-to-Peer an das zu verbindende Gerät. Im AP-Modus startet das zu verbindende Gerät einen Hotspot, mit dem sich das Smartphone verbindet, um dann die Informationen zu übermitteln. Die Bedienungsanleitung der App erläutert das Erstellen von Automatismen und das Anlegen von unterschiedlichen Benutzerprofilen. So können beispielsweise die Lichter eingeschaltet werden, wenn ein Türkontaktsensor eine Türöffnung meldet. In der Bedienungsanleitung des Türkontaktsensors wird beschrieben, dass dieser sich normalerweise im Stand-By-Modus befindet und nur online geschaltet wird, wenn eine Änderung des Öffnungszustandes geschieht. Diese Information wird für die Analyse in Kapitel Routeranalyse relevant (de.hama.com/produkte/smart-home, 2022).

4.6.3 FCC-ID

Keines der Geräte verfügt über eine FCC-ID.

4.6.4 Google Scholar

In der wissenschaftlichen Datenbank Google Scholar wurden keine Veröffentlichungen bezüglich des Hama Smart Home Ökosystems gefunden. Dies war einer der Gründe, das System zur Untersuchung auszuwählen. Auch für die WLAN-Chips wurden keine wissenschaftlichen Arbeiten dokumentiert (scholar.google.com, 2022).

4.7 Netzwerkanalyse im Labor

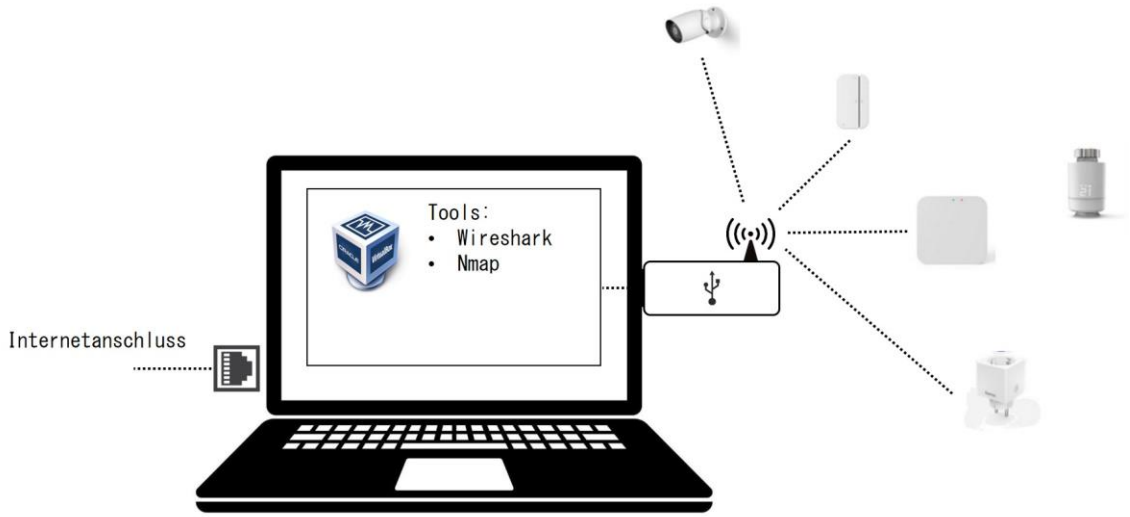


Abbildung 26 - Untersuchungsrechner Netzwerkanalyse

Abbildung 26 zeigt schematisch die Netzwerkuntersuchung. Die Geräte wurden einzeln nacheinander in den Hotspot am PC eingebunden.

Für jedes Gerät wurde eine eigene Versuchsreihe durchgeführt. Diese Versuchsreihen bestanden aus mehreren Szenarien. Jedes Szenario wurde während der Durchführung mit Wireshark beobachtet und die Ergebnisse auf forensische Artefakte untersucht. Im ersten Versuch wurde der Netzwerkverkehr des Routers ohne angeschlossene Geräte aufgezeichnet, um diese Pakete später aus der Analyse herausfiltern zu können.

Die Erkenntnisse aus den folgenden Versuchen wurden tabellarisch dargestellt. In die Tabellen wurden lediglich die relevanten Verbindungen aufgenommen. Zudem wurden nur Verbindungen dargestellt, die im jeweiligen Versuch neu hinzukamen. Die Verbindungen wurden als einzelne Zeile dargestellt, als Quelle wurde angegeben, von welchem Gerät oder Server die jeweilige Verbindung gestartet wurde. Ziel war der Verbindungspartner. Zudem wurden, sofern die Muster in regelmäßigen zeitlichen Abständen auftraten, die zeitlichen Abstände dokumentiert.

4.7.1 IP-Kamera

Protocol	Quelle	Ziel	Delta t [s]	Erkenntnisse
UDP	Kamera	Broadcast	5	Rohdaten an ff:ff:ff:ff:ff:ff
TCP	3.120.XX.XX	Kamera	n.a.	Verbindungsaufbau
TLSv1.2	3.120.XX.XX	Kamera	n.a.	Anwendungsdaten MQTT, TLS-verschlüsselt
TCP	3.120.XX.XX	Kamera	21	Offenhalten der Verbindung mittels Keepalives

Tabelle 2 – Auswertung Wireshark-Mitschnitt - Keine Interaktion

Im ersten Versuch wurde die Kamera an das Labornetzwerk angeschlossen und ohne weitere Interaktion der entstehende Netzwerkverkehr beobachtet (Tabelle 2). Es konnte eine Verbindung der IP-Kamera zum Server 3.120.XX.XX ermittelt werden, innerhalb derer MQTT-Daten übermittelt wurden.

Bei MQTT (Message Queueing Telemetry Transport) handelt es sich um ein Publish-Suscribe-Protokoll, welches für Maschine-zu-Maschine-Kommunikation mit wenig Bandbreiteanforderung geschaffen wurde und auf TCP/IP aufbaut. Hierbei übersenden Clients Daten zu einem definierten Thema an einen Server, der als MQTT-Broker bezeichnet wird. Diese Clients werden als Publisher bezeichnet. Der Broker leitet die Daten an sog. Subscriber weiter. Hierbei handelt es sich um Clients, die Benachrichtigungen zum Thema abonniert haben. Bezogen auf das Hama Smart Home System funktionieren die einzelnen Komponenten als Publisher. Sie übersenden ihre Sensordaten an den Broker. Dieser leitet sie an Subscriber, beispielsweise einen Anwendungsserver für die Smartphone-App, weiter (Bök et al., 2020).

Der Server wurde mit Censys untersucht. Hierbei konnte anhand der gescannten Services festgestellt werden, dass es sich um einen MQTT-Broker handelte.

Anhand der Serverzertifikate ließ sich erkennen, dass es sich um einen Server des Smart Home Anbieters Tuya Smart handelte. Diese Informationen wurden anschließend als Einstiegspunkt für die Untersuchung der Cloud (siehe Kapitel Die Manifest.xml-Datei gewährte einen Überblick über die Funktionen der App. Ziel der apk-Analyse war in diesem Fall, im Sourcecode Hinweise auf die Codierung der Logdateien zu finden. Hierzu mussten die Teile des Quellcodes untersucht werden, in welchen die Erstellung der Dateien programmiert wurde.

Der Quellcode lag in Form von .dex-Dateien vor. Diese wurden mithilfe des Tools dex2jar in Java-Container exportiert und anschließend mit dem Java-Decompiler JD-GUI analysiert. Hierbei konnte festgestellt werden, dass der Quellcode obfuskiert worden war. Die Quellcodeanalyse konnte somit nicht durchgeführt werden.

CloudanalyseCloudanalyse - Hama Smart Home)

Protocol	Quelle	Ziel	Delta t [s]	Erkenntnisse
TLSv1.2	Kamera	3.120.XX.XX	21	Siehe Tabelle 2
TCP	Kamera	18.195.XX.XX	n.a.	Moment des Alarms / Verbindungsaufbau
TLSv1.2	Kamera	35.157.XX.XX	n.a.	Datenaustausch Application Data (encrypted)
TCP	Kamera	18.195.XX.XX	n.a.	Verbindung wird geschlossen
TLSv1.2	Kamera	52.219.XX.XX	n.a.	Datenaustausch Application Data (encrypted)

Tabelle 3 - Auswertung Wireshark Mitschnitt - Kamera: Alarm ausgelöst

Im anschließenden Versuch wurde ein Bewegungsalarm der Kamera ausgelöst (Tabelle 3). Die Verbindung zu den Servern war verschlüsselt. Der Server unter 18.195.XX.XX war der Domain iot-dns.com zuzuordnen. Hierbei handelt es sich

um einen DNS-Server, der mittels DNS-over-TLS angesprochen wird. Da die Verbindung verschlüsselt war, konnten keine DNS-Requests nachvollzogen werden. Die weiteren angesprochenen Server waren Tuya Smart zuzuordnen.

Protocol	Quelle	Ziel	Erkenntnisse
TCP	Smartphone	52.29.XX.XX	Verbindungsaufbau bei Start der App
DNS-Request	Smartphone	Hotspot	mall.tuyaeu.com Response: 18.185.XX.XX
TLSv1.2	18.185.XX.XX	Smartphone	AppData
DNS-Request	Smartphone	Hotspot	static1.tuyaeu.com Response: 13.226.XX.XX
TLSv1.3	13.226.158.97	Smartphone	AppData
DNS-Request	Smartphone	Hotspot	beacons.gcp.gvt2.com Response: 142.250.XX.XX
TLSv1.3	Smartphone	142.250.XX.XX	Verbindungsaufbau
DNS-Request	Smartphone	Hotspot	tytm.tuyaeu.com Response: 52.28.XX.XX
TLSv1.2	Smartphone	52.28.XX.XX	AppData
STUN	Smartphone	18.158.XX.XX	Binding Request, Verbindungsaufbau
TLSv1.2	Smartphone	Kamera	Direkter Verbindungsaufbau Smartphone zu Kamera

STUN	Kamera	Smartphone	Binding Request, Verbindungsaufbau

Tabelle 4 - Auswertung Wireshark-Mitschnitt - Kamera: Abspielen Stream

Der in Tabelle 4 dargestellte Versuch bildet das Starten der App auf dem Smartphone, sowie das Starten des Kamerastreams in der App ab. Durch das Smartphone wurden mehrere unverschlüsselte DNS-Requests an Server des Anbieters Tuya gestellt. Mit dem Starten des Kamerastreams wurde durch den Server eine direkte STUN-Verbindung (Session Traversal Utilities for NAT) zwischen Smartphone und Kamera initiiert. In einem nachfolgenden Versuch wurde festgestellt, dass dieses Verhalten ebenfalls auftritt, wenn das Smartphone sich im mobilen Netz befindet. Es konnte somit nicht ermittelt werden, auf welchem Cloudserver der Kamerastream gesichert wurde.

Port	State	Service
6668/tcp	open	irc?
3702/udp	open	tcpwrapped
3703/udp	open	tcpwrapped

Tabelle 5 - Auswertung Portscan Kamera

Abschließend wurde ein Portscan vorgenommen (Tabelle 5). Die folgenden Optionen wurden gewählt, um mit nmap Informationen über Ports und Services zu erhalten:

sudo nmap -sS -sU -sV 10.42.0.195

Zum offenen Port 6668 konnte kein Service ermittelt werden. Die Ports 3702 und 3073 wurden als tcpwrapped angezeigt. Ein TCP-Wrapper ist ein Programm zur Zugriffskontrolle, welches hinter einen offenen TCP- oder UDP-Port geschaltet wird. Der Port ist demnach geöffnet, dennoch kann auf den Service nicht ohne Berechtigung zugegriffen werden (null-byte.wonderhowto.com, 2016).

4.7.2 Smart LED

Protocol	Quelle	Ziel	Delta t [s]	Erkenntnisse
UDP	Lampe	Broadcast	5	Rohdaten an ff:ff:ff:ff:ff:ff
ARP	Dongle	Lampe	60	Vendor: TuyaSmart
TLSv1.2	Lampe	3.121.XX.XX	60	Application Data (verschlüsselt)
TCP	Lampe	3.121.XX.XX	60	Offenhalten der Verbindung mittels Keepalives

Tabelle 6 - Auswertung Wireshark-Mitschnitt - Keine Interaktion

Wieder wurde im ersten Versuch das Verhalten des Gerätes ohne Interaktion aufgezeichnet (Tabelle 6). Hierbei war festzustellen, dass das Gerät mit einem Tuya-Server Applikationsdaten austauschte. Die Untersuchung auf search.censys.io ergab, dass es sich um einen MQTT-Broker handelte.

Protocol	Quelle	Ziel	Delta t [s]	Erkenntnisse
TLSv1.2	Lampe	35.156.XX.XX	60	Application Data (verschlüsselt)
TCP	Lampe	35.156.XX.XX	60	Offenhalten der Verbindung mittels Keepalives
TLSv1.2	Lampe	18.192.XX.XX	n.a.	Application data (verschlüsselt)

Tabelle 7 - Auswertung Wireshark-Mitschnitt - Smart LED: Ein- und Ausschalten

Im nächsten Versuch wurde die Lampe per App gesteuert ein- und ausgeschaltet. Hinter der IP 35.156.XX.XX steht, analog zur IP 3.121.XX.XX (siehe Tabelle 7) ebenfalls ein MQTT-Broker von Tuya. Bei Ausführung des Einschaltkommandos in der App wurde von der Smart LED eine Verbindung zum Server 18.192.XX.XX

aufgebaut. Da die Verbindung direkt nach dem Einschalten aufgebaut wurde, kann geschlussfolgert werden, dass die Lampe nicht nur als Publisher mit dem Broker kommuniziert, sondern auch als Subscriber. Eingehende Befehle werden vom Broker an die Smart Home Komponenten als Subscriber weitergeleitet. Für die Steuerung der Smart LED war es unerheblich, ob sich das Smartphone im selben Netzwerk befand. In beiden Fällen wurden die Kommandos über die MQTT-Broker übermittelt.

4.7.3 Heizungssteuerung und Smart Plug

Sowohl die Heizungssteuerung als auch der Smart Plug wurden analog zur Smart LED angesteuert. Die Steuerung verlief ebenfalls über MQTT-Broker der Firma Tuya Smart.

4.7.4 Türsensor

Der Türsensor befindet sich im Gegensatz zu den anderen Komponenten dauerhaft im Standby-Modus. Er wird nur aktiv, wenn die Tür geöffnet oder geschlossen wird. Zur Analyse der Netzwerkeigenschaften muss das Gerät folglich getriggert werden.

Gerät	MAC-Adresse	IP-Adresse
Türsensor	E8:68:E7:XX:XX:XX	10.42.0.219

Tabelle 8 - Netzwerkadressen Türsensor

Tabelle 8 listet die Gerätedaten des Türsensors auf. Diese konnten erst ermittelt werden, nachdem der Sensor getriggert wurde.

Protocol	Quelle	Ziel	Delta t [s]	Erkenntnisse
DHCP	Türsensor	Broadcast	n.a.	Der Türsensor fragt den DHCP-Server des Hotspots nach einer IP-Adresse an

DHCP	Hotspot	Türsensor	n.a.	Der Türsensor erhält vom Hotspot die IP 10.42.XX.XX
DNS-Request	Türsensor	Hotspot	n.a.	m2.tuyaeu.com Response: 3.65.XX.XX
TCP	Türsensor	3.65.XX.XX	n.a.	Verbindungsaufbau
DNS-Request	Türsensor	Hotspot	n.a.	a2.tuyaeu.com Response: 3.121.XX.XX
TCP	Türsensor	3.121.XX.XX	n.a.	Verbindungsaufbau
TLSv1.2	Türsensor	3.65.95.68	n.a.	Application data (verschlüsselt)
TLSv1.2	Türsensor	3.121.XX.XX	n.a.	Application data (verschlüsselt)

Tabelle 9 - Auswertung Wireshark-Mitschnitt - Türsensor: Auslösen des Sensors

Tabelle 9 zeigt den Netzwerkverkehr bei Öffnung der Tür. Zunächst wird der Sensor ins Netzwerk eingebunden. Anschließend baut er eine verschlüsselte Verbindung zu den Servern 3.121.XX.XX und 3.65.XX.XX auf. Wieder handelt es sich um MQTT-Server der Firma Tuya Smart. Der Türsensor agiert ausschließlich als Publisher. Er kann keine Kommandos entgegennehmen.

4.7.5 Vergleich vor Ort und Labor

Die Arbeit mit Vergleichsgeräten ist notwendig, um das Verhalten der Geräte bei Interaktion mit der App zu beobachten, ohne sie zurücksetzen zu müssen. Das Verhalten der Geräte im Grundzustand konnte in den Versuchen mit dem Netzwerkscan am Tatort abgeglichen werden. Hierbei wurde die Übereinstimmung verifiziert.

4.7.6 Gewonnene Erkenntnisse

Es konnte ermittelt werden, dass der OEM (Original Equipment Manufacturer) des Hama Smart Home Systems die Firma Tuya Smart ist. Sowohl die Cloudinfrastruktur als teilweise auch die WLAN-Chips werden von Tuya gestellt. Die Geräte werden über das MQTT-Protokoll von einem Broker angesteuert. Sie agieren demnach sowohl als Publisher (zum Übermitteln von Sensordaten) als auch als Subscriber (zum Entgegennehmen von Kommandos aus der App oder aus Automatismen). Der Videostream der Kamera wird über das STUN-Protokoll direkt an das Smartphone übertragen, sofern sich dieses im selben Netzwerk befindet. Andernfalls wird der Datenstrom über einen Server der Firma Tuya umgeleitet. Diese Erkenntnisse werden für die Untersuchung der Cloud in Kapitel Clouduntersuchung Hama Smart Home genutzt.

4.8 Appanalyse - Hama Smart Home

Die Untersuchung des Smartphones musste möglichst zeitnah erfolgen, da sonst möglicherweise Logdateien o.ä. überschrieben worden wären. Es wurden zwei Tools zur Erstellung eines Abbildes des Dateisystems getestet und im Anschluss verglichen. Zunächst wurde der Imaging-Prozess mit dem Programm Axiom (Version 5.7.0.27176) der Firma Magnet Forensics Inc. durchgeführt. Anschließend wurden sowohl ein Full-File-System-Image als auch ein physikalisches Image des Datenspeichers mithilfe der Software UFED von Cellebrite erstellt. Zu jedem Image wurden Hashwerte der vorhandenen Dateien erstellt. Es war festzustellen, dass Axiom einige Nachteile mit sich brachte, die unten näher beschrieben werden. Die Variante wurde verworfen. Anschließend wurde das mit UFED erstellte Full-File-System-Image und das physikalische Image verglichen. Hierbei konnte festgestellt werden, dass beim physikalischen Image alle Partitionen einbezogen werden, während beim Full-File-System-Image nur die Dateien des Android-Dateisystems gespiegelt werden. Mehrere Partitionen, beispielsweise die Bootpartition werden nicht gesichert. Grundsätzlich ist ein physikalisches Image zu bevorzugen, da die Möglichkeit

besteht, gelöschte Dateien durch Carving wiederherzustellen.

4.8.1 Imageerstellung mit Axiom

Axiom bot unterschiedliche Möglichkeiten zur Erstellung eines forensischen Images aus Smartphones. War das Gerät gesperrt, dann konnte versucht werden mithilfe eines angepassten Wiederherstellungsimages die Sperre zu umgehen. Als weitere Option konnte das Aufspielen der modifizierten Firmware, Axiom Rootrechte für die ADB-Schnittstelle geben. Diese Option setzte voraus, dass für das Asservat ein Recovery-Image von Axiom zur Verfügung gestellt wird. Auf der Webseite von Magnet fand sich unter *https://www.magnetforensics.com/resources/advancedmobile/* eine Liste mit kompatiblen Smartphones (magnetforensics.com, 2022). Für das Samsung SM-A600-FN wurde kein Recovery-Image bereitgestellt.

Eine weitere Option war das Versetzen des Gerätes in den MTP-Modus (Media Transfer Protocol). Ein in diesen Modus versetztes Smartphone wird als Massenspeichergerät erkannt. So können die Dateien des Ordners /storage/emulated/0 übertragen werden. Das untersuchte Samsung SM-A600-FN konnte nicht in den MTP-Modus versetzt werden.

Die verbleibende Option war der Datendownload über die ADB-Schnittstelle. Ein entscheidender Nachteil dieser Vorgehensweise ist die Voraussetzung, dass das Asservat entsperrt sein musste. Dennoch wurde die Möglichkeit getestet. Zunächst wurde der Akku über 50% geladen und das Gerät entsperrt. Anschließend wurde der Flugzeugmodus aktiviert und USB-Debugging aktiviert. Danach wurde die ADB-Schnittstelle in den Entwickleroptionen aktiviert. Ab hier arbeitete Axiom ohne weiteren Eingriff. Ein Agent-Programm zum Herunterladen der Nutzerdaten über ADB wurde installiert und ausgeführt. Nachdem die Daten erfolgreich extrahiert wurden, wurde der Agent gelöscht.

Die gefundenen Artefakte wurden aufbereitet und kategorisiert zur Sichtung bereitgestellt. Ausschließlich die für den User sichtbaren Daten wurden mit dieser Methode gesammelt. Von Axiom wurden mit dieser Option nur die Teile des

Speichers gesichert, auf die Nutzer*innen Zugriff hat. Das Image muss daher als unvollständig betrachtet werden.

4.8.2 Imageerstellung mit UFED

Für das Samsung SM-A600-FN stellte UFED zwei Optionen zur Erstellung eines forensischen Images. Für keine der beiden Varianten wurde ein entsperrter Bildschirm vorausgesetzt. Die erste Option war ein Full-File-System-Image. Dabei wurde das gesamte Dateisystem als Ordnerstruktur in eine ZIP-Datei gesichert.

Zunächst wurde das Smartphone durch Drücken der Einschalt- und Leiser-Taste in den Downloadmodus versetzt. Anschließend wurde UFED gestartet und das Gerät mit dem Computer verbunden.

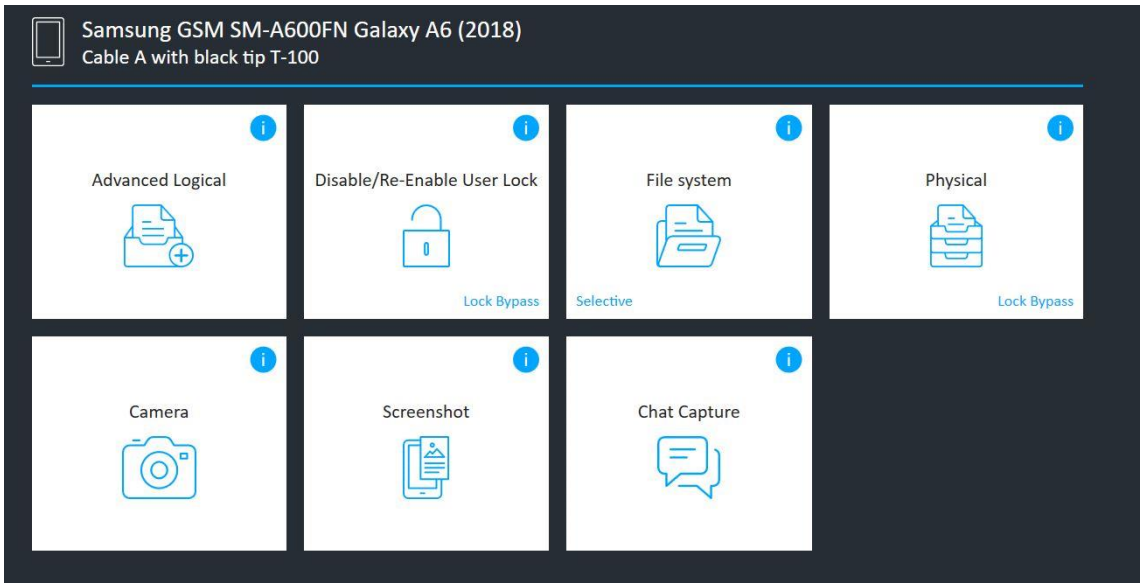
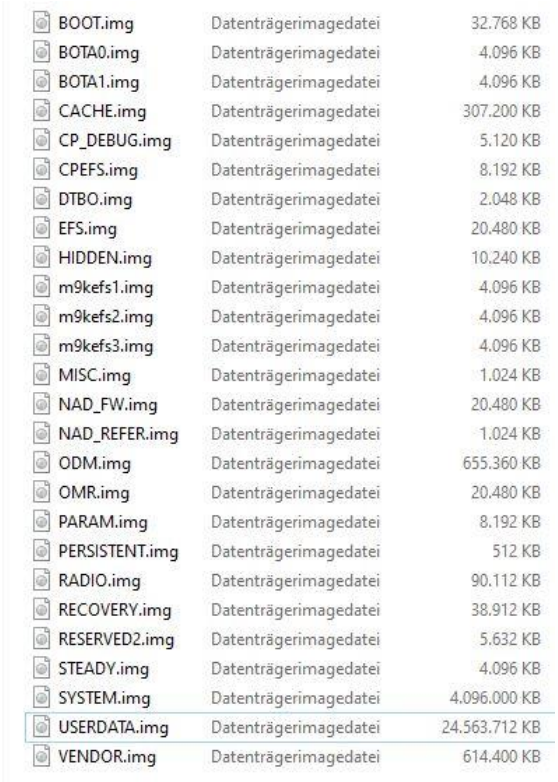


Abbildung 27 - UFED: Optionen zur Imageerstellung

Die Option „File System“ (siehe Abbildung 27) wurde ausgewählt und der Prozess gestartet. Anschließend wurde mit der Option „Physical“ ein zweites Image erstellt. Ein angepasster Bootloader wurde durch die Software auf das Smartphone geladen und gestartet. Somit konnte der physikalische Speicher des Asservats ausgelesen werden. Nach Abschluss des Vorgangs wurden die ausgelesenen Systempartitionen als einzelne Imagedateien abgelegt. Im nächsten Schritt wurden die Partitionen mit der forensischer Software FTK-

Imager (Version 4.3.1.1) eingelesen und die Artefakte extrahiert.



BOOT.img	Datenträgerimagedatei	32.768 KB
BOTA0.img	Datenträgerimagedatei	4.096 KB
BOTA1.img	Datenträgerimagedatei	4.096 KB
CACHE.img	Datenträgerimagedatei	307.200 KB
CP_DEBUG.img	Datenträgerimagedatei	5.120 KB
CPEFS.img	Datenträgerimagedatei	8.192 KB
DTBO.img	Datenträgerimagedatei	2.048 KB
EFS.img	Datenträgerimagedatei	20.480 KB
HIDDEN.img	Datenträgerimagedatei	10.240 KB
m9kefs1.img	Datenträgerimagedatei	4.096 KB
m9kefs2.img	Datenträgerimagedatei	4.096 KB
m9kefs3.img	Datenträgerimagedatei	4.096 KB
MISC.img	Datenträgerimagedatei	1.024 KB
NAD_FW.img	Datenträgerimagedatei	20.480 KB
NAD_REFER.img	Datenträgerimagedatei	1.024 KB
ODM.img	Datenträgerimagedatei	655.360 KB
OMR.img	Datenträgerimagedatei	20.480 KB
PARAM.img	Datenträgerimagedatei	8.192 KB
PERSISTENT.img	Datenträgerimagedatei	512 KB
RADIO.img	Datenträgerimagedatei	90.112 KB
RECOVERY.img	Datenträgerimagedatei	38.912 KB
RESERVED2.img	Datenträgerimagedatei	5.632 KB
STEADY.img	Datenträgerimagedatei	4.096 KB
SYSTEM.img	Datenträgerimagedatei	4.096.000 KB
USERDATA.img	Datenträgerimagedatei	24.563.712 KB
VENDOR.img	Datenträgerimagedatei	614.400 KB

Abbildung 28 - Gespiegelte Partitionen, physikalisches Image

Abbildung 28 zeigt die gespiegelten Partitionen des physikalischen Images. Von besonderer Bedeutung ist die USERDATA-Partition, da hier sowohl vom User als auch von Apps generierte Daten abgelegt werden.

4.8.3 Vergleich Full-File-System-Dump und physikalische Spiegelung

Im Vergleich zwischen Full-File-System-Dump und physikalischer Spiegelung wurde festgestellt, dass beide Optionen die vorhandenen Dateien vollständig enthalten. Der Vorteil des physikalischen Images ist die Möglichkeit, gelöschte Dateien durch Carving wiederherzustellen. Weiterhin können die einzelnen Image-Dateien der Systempartitionen leichter mit forensischer Software verarbeitet werden. So wird ein unbeabsichtigtes Verändern der Zugriffszeitstempel vermieden. Aus diesem Grund wurden die nachfolgenden Untersuchungsschritte mit dem physikalischen Image vorgenommen.

Die Zeitstempel wurden mithilfe des Tools fls aus der Sleuth Kit-Suite in einer Textdatei abgelegt. Der hierzu verwendete Befehl lautet:

fls -prl USERDATA.img > Timestamps_USERDATA.txt

Mithilfe des Befehls wurden auch gelöschte Dateien angezeigt, die nicht im Full File System Image inbegriffen sind. Diese Dateien wurden durch fls mit * und (realloc) markiert. Der Asterisk markiert gelöschte Dateien, realloc zeigt an, dass die Metadaten getrennt aufbewahrt wurden und ebenfalls gelöscht oder verschoben wurden. Es können somit keine Metadaten zu den gelöschten Dateien ermittelt werden (wiki.sleuthkit.org, 2014).

4.8.4 Versuchsreihen

Da viele Logeinträge nur über einen kurzen Zeitraum gespeichert werden, wurden zwei Spiegelungen durchgeführt:

- 29.08.2022 17:02
- 06.09.2022 17:45

Um die Arbeit mit einem Vergleichsgerät zu simulieren, wurde das Smartphone am 07.09.2022 zurückgesetzt und am 11.09.2022 eine weitere Versuchsreihe durchgeführt. Zunächst wurden alle Geräte in das geklonte Tatortnetzwerk im Labor integriert, sodass die Datenströme analysiert werden konnten.

Versuch Nr.	Beschreibung	Datum	Uhrzeit
1	Anmeldung Useraccount	11.09.2022	22:05:00
2	App Account angelegt	11.09.2022	22:11:00
3	App öffnen, App schließen	11.09.2022	22:21:00

4	Benachrichtigung Türsensor erzeugen (Flightmode aus)	11.09.2022	22:23:00
5	Benachrichtigung Türsensor erzeugen (Flightmode aus)	11.09.2022	22:23:05
6	Benachrichtigung Türsensor erzeugen (Flightmode aus)	11.09.2022	22:25:00
7	Benachrichtigung Türsensor erzeugen (Flightmode aus)	11.09.2022	22:25:05
8	Benachrichtigungscenter öffnen (Flightmode aus)	12.09.2022	22:26:30
9	Benachrichtigung Türsensor erzeugen (Flightmode an)	11.09.2022	22:28:00
10	Benachrichtigung Türsensor erzeugen (Flightmode an)	11.09.2022	22:28:05
11	Benachrichtigung Türsensor erzeugen (Flightmode an)	11.09.2022	22:31:00
12	Benachrichtigung Türsensor erzeugen (Flightmode an)	11.09.2022	22:31:05
13	Benachrichtigungscenter öffnen (Flightmode kurz vorher aus)	11.09.2022	22:31:15
14	Kamerastream öffnen (Cloud)	11.09.2022	22:33:15
15	Steckdosensockel schalten	11.09.2022	22:34:30

Tabelle 10 - Versuchsreihen Appuntersuchung

Im Anschluss wurden die in Tabelle 10 beschriebenen Versuche durchgeführt, um die Zeitstempel später mit einer weiteren Spiegelung des Smartphones abzugleichen. Die Spiegelung fand am 12.09.2022 um 10:06 statt.

4.8.5 Gefundene Artefakte

Das physikalische Abbild des Gerätespeichers beinhaltete Artefakte, die Hinweise auf die Nutzung der Hama-App geben. Teilweise handelt es sich um Systemartefakte, zum Anderen wurden die Artefakte durch die Applikation erzeugt.

EFS/wifi/.mac.info

Diese Datei enthielt die physikalische MAC-Adresse des untersuchten Smartphones (4C:DD:31:XX:XX:XX).

USERDATA/misc/wifi/WifiConfigStore.xml

Innerhalb dieser Datei wurden Informationen zu gespeicherten WLAN-Netzwerken abgelegt.
Neben SSID, MAC-Adresse, preSharedKey (Hashwert) sowie technischen Parametern wurde hier auch die Random-MAC des Smartphones gespeichert, welche im jeweiligen Netzwerk verwendet wird. Zudem finden sich Zeitstempel zur Anmeldung und letzten Nutzung des Netzwerks.

SSID	Random-MAC-Smartphone	Erste Anmeldung	Letzte Anmeldung
SmartLife-32B1	e6:5b:f5:XX:XX:XX	2022-08-19 21:43:47	2022-08-19 21:43:48
SmartLife-489F	7e:55:75:XX:XX:XX	2022-08-19 22:40:27	2022-08-19 22:40:32
Vodafone-449C	76:35:9c:XX:XX:XX	2022-07-26 20:09:08	2022-08-28 17:24:14

Tabelle 11 - Ausschnitt aus WifiConfigStore.xml

Die relevanten Netzwerke werden in Tabelle 11 aufgeführt. Hierbei handelt es sich zum einen um den Router am Versuchstatort. Zum anderen werden zwei Access-Points mit der Bezeichnung „SmartLife-XXXX“ aufgelistet. Hierbei handelte es sich, wie in Kapitel Beschreibung der Komponenten erläutert, um zwei Geräte des Hama Smart Home Ökosystems, welche im AP-Modus betrieben wurden. Es ließ sich demnach aus den gefundenen Informationen herleiten, wann ein Versuch zum Einbinden der Geräte in ein WLAN durchgeführt wurde. Es ließ sich nicht ermitteln, ob der Versuch erfolgreich war. Die Spiegelung des Vergleichsgerätes zeigte, dass sich durch das Zurücksetzen die Random-MAC-Adresse für das Netzwerk „Vodafone-449C“ in 76:e1:39:XX:XX:XX änderte.

USERDATA/system_de/0/accounts_de.db

Die Datei enthält die E-Mail-Adresse des angemeldeten Benutzers. Zudem werden die Zeitstempel der Einrichtung des Kontos auf dem Smartphone und der letzten Passworteingabe für das Google-Konto angegeben. Beides erfolgte am 27.06.2022 um 21:58:10.
Nachdem das Smartphone zurückgesetzt wurde, wurden beide Zeitstempel am 11.09.2022 um 22:05:28 geschrieben.

USERDATA/system/notification_log.db

In dieser Datei werden vom System alle Benachrichtigungen geloggt, die den Benutzer*innen angezeigt werden. Die Benachrichtigungen können nach Apps gefiltert werden. Die Benachrichtigungen werden 7 Tage lang gespeichert.

ID	Zeitstempel	Notification App	Erzeugt durch	nid	tag
2050	23.08.2022 21:11:00	21:10:59	Tür	0	FCM-Notification:17383017
2112	24.08.2022 19:06:38	19:06:33	Kamera	0	FCM-Notification:19180059

2140	24.08.2022 22:34:38				
2141	24.08.2022 22:34:38				
2142	24.08.2022 22:34:38				
2144	24.08.2022 22:35:05	22:35:00	Tür	0	FCM-Notification:20517281
2197	25.08.2022 08:30:00	8:29:53	Tür	0	FCM-Notification:22122567
2198	25.08.2022 08:50:45	8:50:38	Kamera	0	FCM-Notification:22141998
2199	25.08.2022 09:48:13	09:48:05	Mitteilung	0	FCM-Notification:22231862
2200	25.08.2022 09:48:13			2147483647	ranker_group
2305	27.08.2022 09:39:15				
2307	27.08.2022 09:39:17				
2308	27.08.2022 09:39:19				
2309	27.08.2022 09:39:19				
2312	27.08.2022 10:34:29	10:34:20	Tür	0	FCM-Notification:27107506
2314	27.08.2022 15:15:43	15:15:33	Kamera	0	FCM-Notification:27494383
231	27.08.2022	16:59:04	Kamera	0	FCM-Notification:27842871

8	16:59:13				
2321	27.08.2022 18:36:56	18:36:46	Tür	0	FCM-Notification:28067306
2322	27.08.2022 18:36:56			214748364 7	ranker_group
2376	28.08.2022 15:47:04				
2377	28.08.2022 15:47:04				
2378	28.08.2022 15:47:05				
2379	28.08.2022 15:47:05				

Tabelle 12 - Auswertung notification_log.db

In Tabelle 12 wurden die Artefakte aus dem Image vom 29.08.2022 aufbereitet. Die Datenbankeinträge wurden mit den im Nachrichtencenter der App gespeicherten Benachrichtigungen abgeglichen.

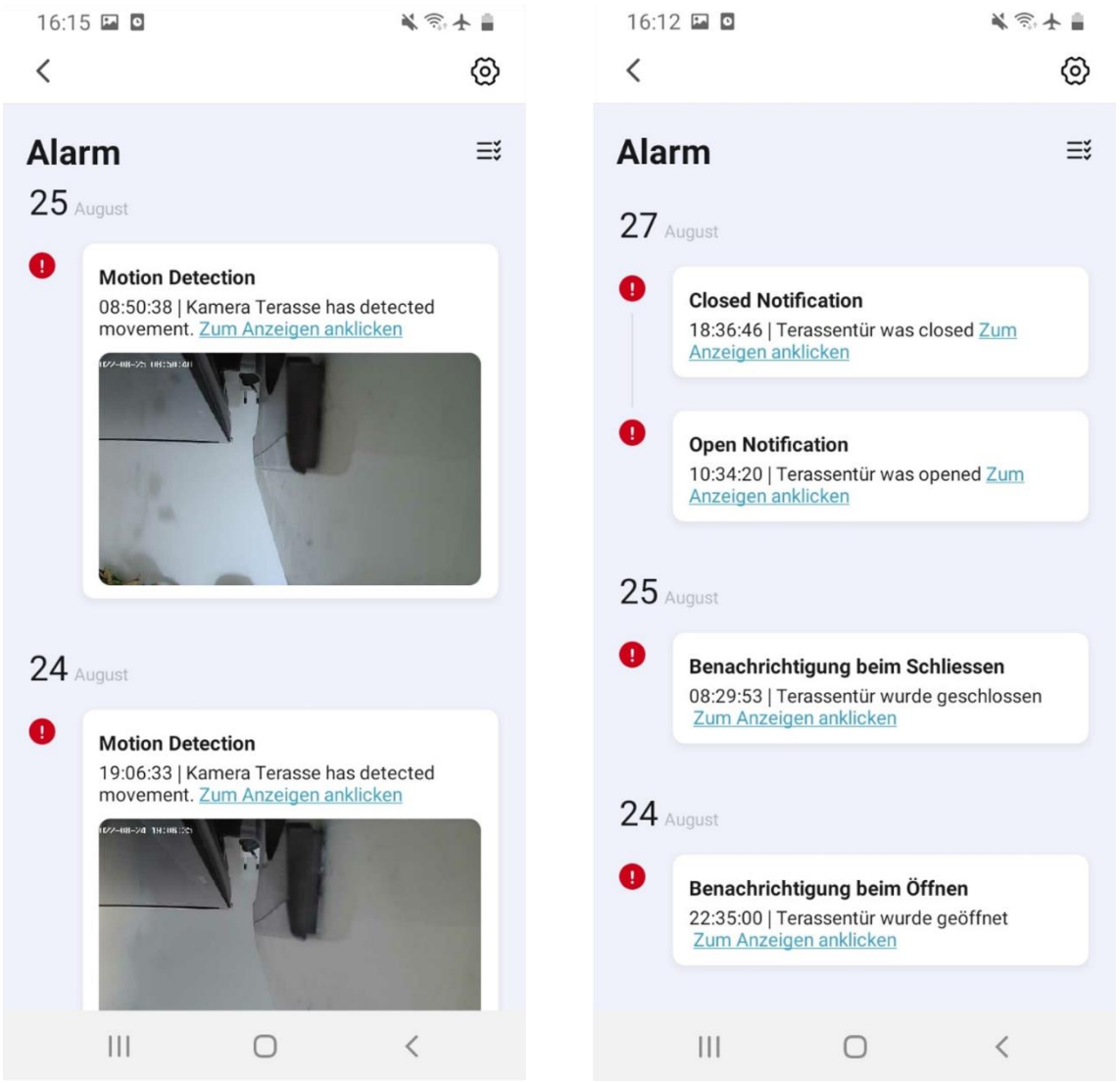


Abbildung 29 - Screenshots: Benachrichtigungscenter Hama Smart App

Abbildung 29 zeigt Auszüge aus dem Benachrichtigungscenter der App.
Hierbei wurde eine Übereinstimmung festgestellt. Dennoch war anhand der Datenbankeinträge nicht ersichtlich, von welchem Gerät die Benachrichtigung gesendet wurde.

USERDATA/system/usagestats/0/*/*

Die Nutzungslogs des analysierten Gerätes wurden innerhalb dieses Ordners in tägliche, wöchentliche, monatliche und jährliche Logs unterteilt. Bis Android 10 wurden diese Logs im xml-Format geschrieben. Seit Android 10 werden die Logdateien mit Protocol Buffers strukturiert (developers.google.com, 2022).

Diese wurden von Google entwickelt, um Daten, ähnlich wie im xml-Format strukturiert abzuspeichern. Der Vorteil gegenüber xml ist der geringe Overhead.
Zur Analyse der Usagestat-Logs musste ein Parser verwendet werden. Parsing stellt einen wesentlichen Bestandteil der Analyse forensischer Artefakte dar (github.com/abrignoni, 2020).
Der Parser besteht aus einem Python-Skript, welchem neben den Usagedata-Logs auch die Struktur dieser Dateien übergeben wurden. Diese Strukturen finden sich im GitHub-Repository des Android Open Source Project (github.com/aosp-mirror, 2022). Mit Ausführung des Skriptes wurde eine Datenbank und ein HTML-Bericht erstellt, in denen die Einträge der Usagestats detailliert aufgelistet wurden (abrignoni.blogspot.com, 2019).

In den Logdateien befanden sich Informationen über die Nutzung des Smartphones, wie z.B.:

- SCREEN_INTERACTIVE: Sperrbildschirm deaktiviert
- SCREEN_NON_INTERACTIVE: Sperrbildschirm aktiviert
- MOVE_TO_FOREGROUND: App wird im Vordergrund ausgeführt
- MOVE_TO_BACKGROUND: App wird in den Hintergrund geschoben
- NOTIFICATION_INTERRUPTION: Benutzer*in erhält eine Benachrichtigung
- NOTIFICATION_SEEN: Benutzer*in interagiert mit Benachrichtigung

Auch bei Benutzung der Hama-App wurden Logeinträge angelegt. Diese Logeinträge wurden mit den Zeiten in Tabelle 10 abgeglichen. Hierbei konnten folgende Erkenntnisse gewonnen werden:

- com.tuya.TuyaSplashActivity: Eine Splash-Activity ist das erste Userinterface, das geöffnet wird, wenn eine App gestartet wird. Dieser Logeintrag zeigt an, wann die App geöffnet wurde.
- com.tuya.smart.hometab.activity.FamilyHomeActivity: Diese Aktivität wird aufgerufen, sobald das Untermenü „Profil“ angewählt wird.
- com.tuya.message.base.activity.message.MessageDetailsActivity: Diese Aktivität wird aufgerufen, wenn das Nachrichtencenter geöffnet wird.

- com.tuya.smart.ipc.camera.rnpanel.activity.TYRCTSmartCameraPanelActivity: Diese Aktivität wird gestartet, sobald das Kameramenü geöffnet wird.
- com.tuya.smart.ipc.cloud.panel.activity.CameraCloudActivity: Wird der Clouddienst der IP-Kamera aufgerufen, so startet diese Aktivität.

USERDATA/system/users/0/runtime-permissions.xml

In dieser Datei wird definiert, welche Berechtigungen eine App besitzt. Der Hama-App wurden für die Versuchsreihe folgende Berechtigungen gewährt:

- Standort
- Dateizugriff (R/W)
- Kamera
- Mikrofon

USERDATA/media/0/Android/Data/com.hama.smart/cache/*

Die Cache-Dateien im Userverzeichnis liegen in einem codierten Format vor. In einem späteren Schritt wurde anhand des Quelltextes der App versucht, die Dateien zu decodieren.

USERDATA/data/com.hama.smart/cache/camera/*

In diesem Ordner konnte eine Bilddatei gefunden werden, die einen Aufnahme des Kamerastreams zeigte.

Abbildung 30 - Bilddatei aus Cache der Hama Smart App

Die Bilddatei ist in Abbildung 30 dargestellt. Mithilfe des Exif-Tools wurden die Exif-Informationen des Bildes extrahiert. Diese sind im Folgenden aufgelistet:

File Name	: tuya_camera.jpg
Directory	: F:/Extraktion_A6/Artefakte
File Size	: 339 kB
File Modification Date/Time:	: 2022:08:19 22:08:47+02:00
File Access Date/Time	: 2022:08:19 20:00:46+02:00
File Creation Date/Time	: 2022:08:19 20:00:46+02:00
File Permissions	: rw-rw-rw-
File Type	: JPEG
File Type Extension	: jpg
MIME Type	: image/jpeg
JFIF Version	: 1.01
Resolution Unit	: None
X Resolution	: 1
Y Resolution	: 1
Image Width	: 1920
Image Height	: 1080

Encoding Process	: Baseline DCT, Huffman coding
Bits Per Sample	: 8
Color Components	: 3
Y Cb Cr Sub Sampling	: YCbCr4:2:0 (2 2)
Image Size	: 1920x1080
Megapixels	: 2.1

Auffällig ist, dass die Creation Time um 20:00 angegeben wird, während der Zeitstempel im Bild 22:08 anzeigt.

Abbildung 31 - Bilddatei aus Cache der Hama Smart App

Zur weiteren Untersuchung wurde in einem zweiten physikalischen Image vom 06.09.2022 um 17:45 im gleichen Ordner ein Bild gefunden und analysiert (siehe Abbildung 31). Die Exif-Informationen zeigten, dass die Access und Creation Time bei beiden Bildern übereinstimmte. Lediglich die Modification Time unterschied sich. Diese entsprach dem Zeitstempel auf dem Bild. Die Zeitstempelanalyse zeigt, dass am 19.08.2022 um 20:00:46 zum ersten Mal ein Bild in dem Ordner angelegt wurde. Dieses Bild wurde nachfolgend überschrieben. Es konnte nicht ermittelt werden, nach welchem Muster die Bilddatei erstellt wurde.

USERDATA/data/com.hama.smart/cache/image_cache/*/*

Dieses Image-Cache enthielt 31 Bilddateien im .cnt-Format. Es handelte sich um Bilddateien der Benutzeroberfläche.

USERDATA/data/com.hama.smart/app_webview/Default/Cookies

Die Datenbankdatei beinhaltet Cookies, welche durch die Hama-App genutzt werden, um Verbindungen zu Webdiensten aufzubauen. Neben Cookies für den Bezahldienst Paypal finden sich Datenbankeinträge zu Webdiensten der Smart Home Herstellerfirma Tuya. Diese können möglicherweise während der Cloudanalyse genutzt werden, um auf Dienste zuzugreifen.

USERDATA/data/com.hama.smart/cache/okhttp3/*

Der Ordner okhttp3 enthielt Dateianfragen, die an Server des Infrastruktur-Anbieters Amazon Web Services (AWS) gerichtet waren. Die Dateien wurden am 11.09.2022 um 22:33 angelegt. Zu diesem Zeitpunkt wurde der Kamerastream in der App aufgerufen (siehe Tabelle 10). Die gefundenen Informationen gaben nicht nur Aufschluss über den angesprochenen Server, sondern beinhalteten auch einen TLSv1.2-Key, sowie Angaben über die Lebensdauer der Anfrage. In der URL wurde der Zeitstempel 1661840445 festgestellt. Dieser Zeitpunkt entspricht dem 30.08.2022 um 20:20:45. Es handelt sich dabei um den letzten aktiven Zeitpunkt der Kamera. Somit lässt sich der Zeitpunkt ermitteln, ab dem eine Hama-Kamera deaktiviert wurde.

USERDATA/data/com.hama.smart/app_webview/webview_pref_Store

Innerhalb dieser Datei konnte der Zeitstempel der Installation der Hama-App gefunden werden. Dieser wurde mit Tabelle 10 verglichen und lag am 11.09.2022 um 22:10:47.

USERDATA/data/com.hama.smart/cache/WebView/Default/HTTP Cache/Cache_Data/*

Mithilfe von Android SystemWebView können Webinhalte in Applikationen ohne Browser dargestellt werden. Teile dieser Inhalte werden im sogenannten WebView-Cache zwischengespeichert (makeuseof.com, 2021). Das Webview

Cache wurde wie von Rajewski (2016) beschrieben entpackt und analysiert. Hierbei konnten Benutzerinformationen und Domaininformationen festgestellt werden.

Gefundene Domaininformationen:

"domain":{"aispeechHttpsUrl":"https://aispeech.tuyaeu.com",

"aispeechQuicUrl":"https://i1.tuyaeu.com",

"deviceHttpUrl":"http://a.tuyaeu.com",

"deviceHttpsPskUrl":"https://a3.tuyaeu.com",

"deviceHttpsUrl":"https://a2.tuyaeu.com",

"deviceMediaMqttUrl":"s.tuyaeu.com",

"deviceMediaMqttsUrl":"ms.tuyaeu.com",

"deviceMqttsPskUrl":"m2.tuyaeu.com",

"deviceMqttsUrl":"m2.tuyaeu.com",

"fusionUrl":"https://apigw.tuyaeu.com",

"gwApiUrl":"http://a.gw.tuyaeu.com/gw.json",

"gwMqttUrl":"mq.gw.tuyaeu.com","httpPort":80,"httpsPort":443,"httpsPskPort":443,

"mobileApiUrl":"https://a1.tuyaeu.com",

"mobileMediaMqttUrl":"s.tuyaeu.com",

"mobileMqttUrl":"mq.mb.tuyaeu.com",

"mobileMqttsUrl":"m1.tuyaeu.com",

"mqttPort":1883,"mqttQuicUrl":"q1.tuyaeu.com",
"mqttsPort":8883,"mqttsPskPort":8886,

"pxApiUrl":"http://px.tuyaeu.com","regionCode":"EU",

"tuyaAppUrl":"app-support.tuyaeu.com",

"tuyaImagesUrl":"images.tuyaeu.com"},

Gefundene Userinformationen:

"email":"mastersonthesis@gmail.com",
"extras":{"developer":0,"passwordSet":1},
"headPic":"","id":"eu165886115813427haR",
"mobile":"",
"nickname":"",
"phoneCode":"49",
"regFrom":0,
"snsNickname":"",
"tempUnit":1,
"timezoneId":"Europe/Berlin",
"userType":1,"username":"mastersonthesis@gmail.com"},
"isPublic":true,
"appVersionMatch":true},
"globalError":{},

USERDATA/data/com.hama.smart/files/log/main/*

Die xlog-Dateien in diesem Verzeichnis waren codiert abgelegt. Es wurde versucht, die Dateien zu decodieren, um die möglicherweise forensisch relevanten Logs auslesen zu können. Hierzu wurde zunächst versucht, zusammenhängende Zeichenketten zu analysieren:

strings 5nt39473emacqwnc4fv9_1.2.8_Android_main_20220826_000113.xlog | sort | uniq -c | sort -nr

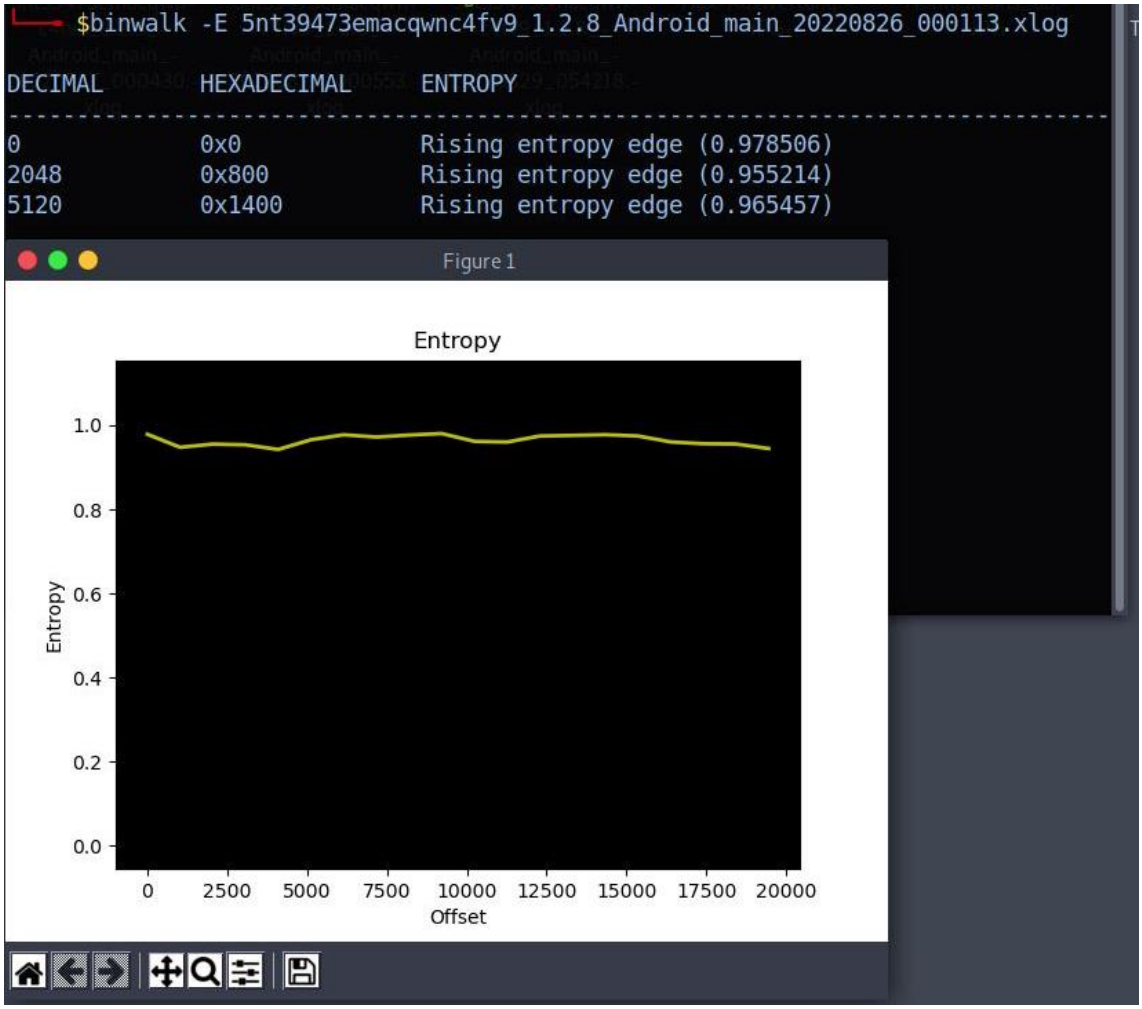


Abbildung 32 – Screenshot: Entropieanalyse mit Binwalk

Es folgte, wie in Abbildung 32 – Screenshot: Entropieanalyse mit Binwalk dargestellt, eine Entropienanalyse mit dem Tool Binwalk. Auch in Android Studio konnte die Datei mit den verschiedenen Decodern nicht decodiert werden. Anschließend wurde die Datei in einem Hexeditor analysiert. Der Fokus wurde hierbei auf die sich wiederholenden Pattern gelegt.

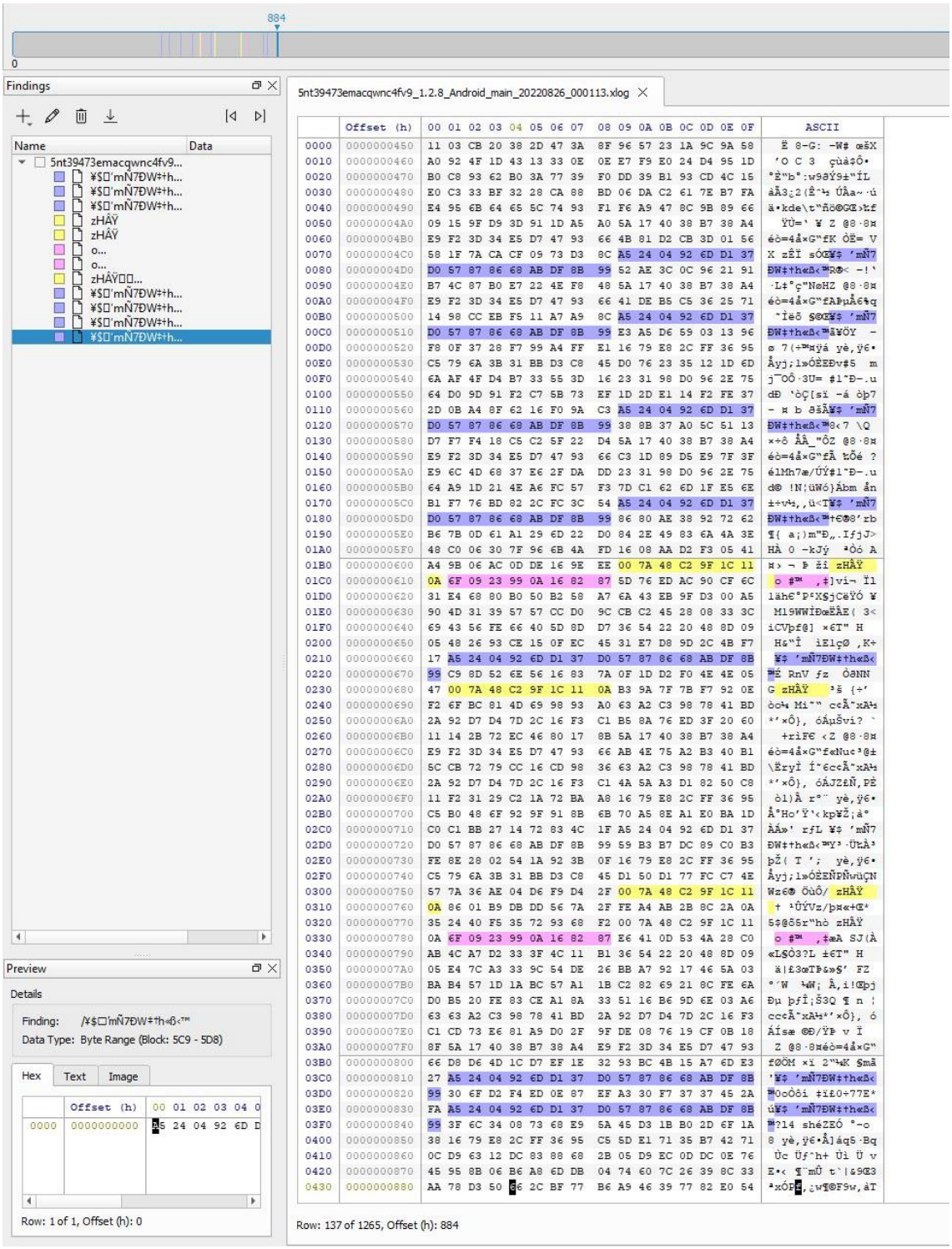


Abbildung 33 - Hexanalyse .xlog-Datei

Abbildung 33 zeigt die Patterns im Hexeditor. Trotz weiterer Decodierungsversuche konnten die Dateien nicht in ein lesbares Format übertragen werden.

USERDATA/app/com.hama.smart-KfOuyC4VFhzO41mX0EkXaw==/base.apk

Die Datei base.apk enthält den Quellcode sowie benötigte Ressourcen und Konfigurationsdateien der Hama-App. Zunächst wird die Datei aus dem physischen Image exportiert und in Android-Studio geöffnet. Die Manifest.xml-Datei gewährte einen Überblick über die Funktionen der App. Ziel der apk-Analyse war in diesem Fall, im Sourcecode Hinweise auf die Codierung der Logdateien zu finden. Hierzu mussten die Teile des Quellcodes untersucht werden, in welchen die Erstellung der Dateien programmiert wurde.

Der Quellcode lag in Form von .dex-Dateien vor. Diese wurden mithilfe des Tools dex2jar in Java-Container exportiert und anschließend mit dem Java-Decompiler JD-GUI analysiert. Hierbei konnte festgestellt werden, dass der Quellcode obfuskiert worden war. Die Quellcodeanalyse konnte somit nicht durchgeführt werden.

4.9 Cloudanalyse - Hama Smart Home

Im Kapitel Netzwerkanalyse im Labor wurde unter anderem ermittelt, dass das Hama Smart Home System auf der Cloudinfrastruktur des Anbieters Tuya Smart basiert. Sowohl die Steuerungsserver als auch die Datenzentren mit den gespeicherten Videodateien der IP-Kamera werden von Tuya Smart betrieben. Es ist jedoch kein direkter Zugriff auf die Serverinfrastruktur möglich. In diesem Kapitel wird beschrieben, wie nach Möglichkeiten gesucht wurde, forensische Artefakte aus der Cloud zu sammeln.

4.9.1 Tuya Development Platform

Tuya bietet für Entwickler*innen die Möglichkeit, ihre eigenen Smart Home und Smart Industry Produkte in der Cloud zu entwerfen. Neben Cloudanwendungen

können hier auch Apps und eigene Smart Home Produkte entwickelt werden. Um ein besseres Verständnis für die Tuya Cloud Infrastruktur zu erlangen wurde ein Account angelegt.

Anschließend wurde versucht eine App zu entwickeln, um möglicherweise Informationen über die Codierung der Logdateien zu erhalten. Auf der Plattform Tuya IoT Plattform konnte jedoch lediglich das Front End der App entworfen werden (iot.tuya.com, 2022). Nach der Appentwicklung wurde die Cloud-API nach Ansatzpunkten für eine forensische Analyse durchsucht. Es konnten keine hilfreichen Informationen zur Analyse der Smart Home Geräte von Hama gefunden werden.

4.9.2 AWS-Datarequests aus Appanalyse

Im Ordner data/data/com.hama.smart/cache/okhttp3/* wurden im Zuge der Untersuchung des Smartphones Data Requests an AWS-Server aufgefunden. Ebenfalls konnte ermittelt werden, dass die Anfragen durch Zugriff auf die Clouddaten der IP-Kamera erzeugt wurden. Es wurde versucht, mithilfe der Requests, einen Zugriff auf die Daten zu erlangen.

Die folgenden Informationen konnten dem Artefakt entnommen werden:

```
https://ty-eu-storage14.s3.eu-central-1.amazonaws.com/b54857-62146109-pp01aa5c88cd31d65aab/v/1661840445.jpg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20220911T203312Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3599&X-Amz-Credential=AKIAUTPMUJJJW6O5Q3MG%2F20220911%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Signature=063e82aa6288188f849b5206859e213b51988a6305e44068c8ac7a63dc76fc63

GET

0

HTTP/1.1 200 OK

12
```

x-amz-id-2: MlGlsUhtAfrJ/gaWASBl6JN2BPM0GikgPP/cbJe2P4BIL+ncuujjMLvi7hUSb2bqGgdRJKyEa2w=

x-amz-request-id: B6GJX9XX8T8JB44N

Date: Sun, 11 Sep 2022 20:33:14 GMT

Last-Modified: Tue, 30 Aug 2022 06:20:46 GMT

x-amz-expiration: expiry-date="Wed, 14 Sep 2022 00:00:00 GMT", rule-id="14day-auto-clean"

ETag: "339aa84fbf0c11ca95a9bb9320df318d"

Accept-Ranges: bytes

Content-Type: media/jpeg

Server: AmazonS3

Content-Length: 9824

OkHttp-Sent-Millis: 1662928391906

OkHttp-Received-Millis: 1662928391971

Zudem war ein TLSv1.2 Zertifikat angehängt.

Abbildung 34 – Screenshot: Fehlermeldung ty-eu-storage

Beim Öffnen des Links wurde eine Fehlermeldung angezeigt (siehe Abbildung 34). Auch ein Anpassen des Links führte zu keinerlei Ergebnis. Es wurde festgestellt, dass die Request-ID sich mit jeder Anfrage auf einen neuen Wert geändert wurde. Der Zugriff über die Datenanfragen war nicht möglich.

4.9.3 Google Home App

In einem letzten Schritt wurden die Smart Home Geräte in die Google Home App eingebunden. Anschließend wurde das Google Konto nach forensischen Artefakten aus der Smart Home Anwendung untersucht. Dieses Vorgehen blieb ebenfalls ergebnislos.

4.9.4 Anfrage bei Hama und Tuya

In einer strafrechtlichen Ermittlung ist es möglich, ein Auskunftsersuchen an den Cloudanbieter zu stellen. Im Kapitel Cloudanalyse wurden die juristischen und technischen Hürden hierfür beschrieben. Es ist zu betonen, dass es ohne Durchsuchungsbeschluss nahezu unmöglich ist, eine Auskunft zu erhalten.

4.10 Geräteanalyse - Hama Smart Home

Zur Geräteanalyse wurden die IP-Kamera und die smarte Heizungssteuerung ausgewählt. Ziel der Analyse war zunächst, eine Kommandozeile mit Zugriff auf die Firmware zu öffnen. Anschließend sollte ein Image des Datenspeichers erstellt werden.

4.10.1 Kamera

Die IP-Kamera wurde zunächst auseinandergebaut. Sie bestand aus einer Platine mit aufgesetzter Kamera und einem Stromanschluss. Angeschlossen waren die Peripheriegeräte Mikrofon, Lautsprecher sowie Infrarotsensor.

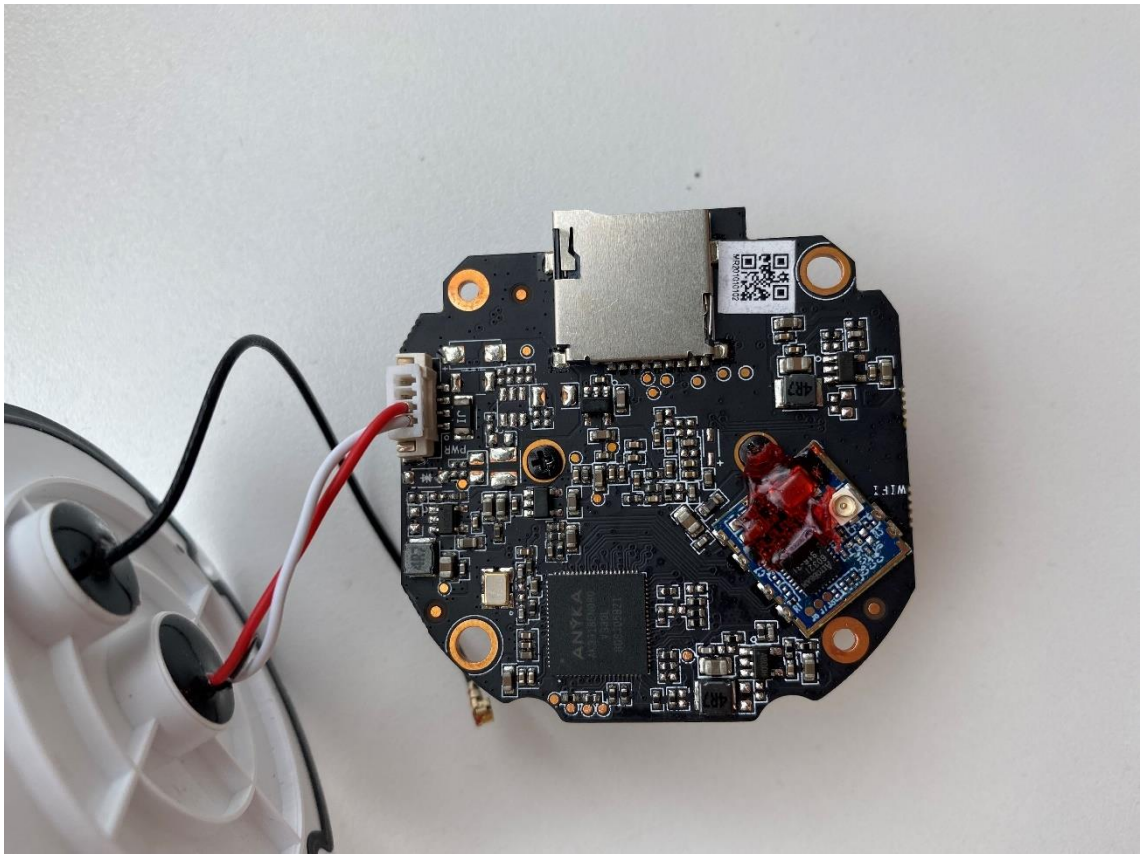


Abbildung 35 - Platine der IP-Kamera

Nachdem die Platine wie in Abbildung 35 dargestellt ausgebaut war, wurde sie in einer Platinenhalterung fixiert. Anschließend wurden die einzelnen Komponenten untersucht. Unter anderem konnte ein Chip von Anyka (Modell AK3918) identifiziert werden. Eine Recherche ergab, dass es sich um ein System on a Chip (SoC) handelt, welches speziell für IP-Kameras entwickelt wurde. Sowohl die Firmware als auch das Dateisystem waren in dem Chip lokalisiert.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	MIC_P	MIC_N	HPL	HPR	VCM3	BAT_RTC	XTAL32KO	XTAL32KI	GPIO[60]	GPIO[59]	GPIO[58]	GPIO[56]/ PWM5	CIS_D[3	CIS_D[2]	CIS_D[1]/ GPIO[9]	CIS_D[0]/ GPIO[8]	CIS_HSYNC
B	VREF	VCM2	Linein_R	AIN0	#RST	GPIO[57]/ I2S_DIN	GPIO[50]/ PWM3	GPIO[49]	GPIO[4]/ TDI/ RXD2/ PWM1	GPIO[54]/ I2S_BCLK	GPIO[48]/ PWM1	SPI1_CLK / GPIO[26]	CIS_D[9]/ GPIO[11	CIS_D[7]	CIS_D[5]	VDDIO_CIS	CIS_D[4]
C	HPVDD	AIN1	Linein_L	AVSS	AVCC	WAKEUP	OPCLK/ GPIO[47]/ PWM5	GPIO[3]/ TMS	GPIO[53]/ I2S_MCLK	#SPI1_CS / GPIO[25	GPIO[7]/ RTCK/ RTS2/ PWM4	GPIO[6]/ TCLK/ CTS2/ PWM3	CIS_SCLK	CIS_PCLK	CIS_VSYNC	CIS_D[8]/ GPIO[10]	CIS_D[6]
D	USB_DP	USB_RREF	HPVSS												VDDIO	RXD1/ GPIO[1]	TXD1/ GPIO[2]
E	USB_DM	AGND_USB	AVSS_PLL												VDD	I2C_CLK/ GPIO[27]	I2C_DAT/ GPIO[28]
F	AVDD12_PLL	VCCA_USB	MCI1_MCK/ GPIO[32]			VDDIO	VSS	VSS	VSSIO	VSSIO	VDD	VDD			DRAM_VREF	PWM5/ GPIO[30]/ #SPI2_CS/ I2S_MCLK	IRDA_DI/ GPIO[29]/ SPI2_CLK
G	MCI1_D[3]/ GPIO[36]	MCI1_MCMD/GPIO[31]	MCI1_D[0]/ GPIO[33]			VSS	VSS	VSS	VSSIO	VSSIO	VDD	VDD			#DRAM_CS	VDD_DDR2	VDD_DDR2
H	XTAL12MI	XTAL12MO	MCI1_D[1]/ GPIO[34]												VDD_DDR2	VSSIO _DRAM	VDDIO _DRAM
J	MCI1_D[4]/GPIO[37]/ SPI2_DIN[note4]/ SPI1_DIN[note3]	MCI1_D[5]/GPIO[38]/ SPI2_DOUT[note4] /SPI1_DOUT[note3]	MCI1_D[2]/ GPIO[35]												VSS_DDR2	VSSIO _DRAM	VDDIO _DRAM

Abbildung 36 - Pinout Datenblatt der IP-Kamera (Quelle Anyka, 2022)

Das heruntergeladene Datenblatt enthielt Informationen über zwei UART und eine JTAG-Schnittstelle (siehe Abbildung 36).

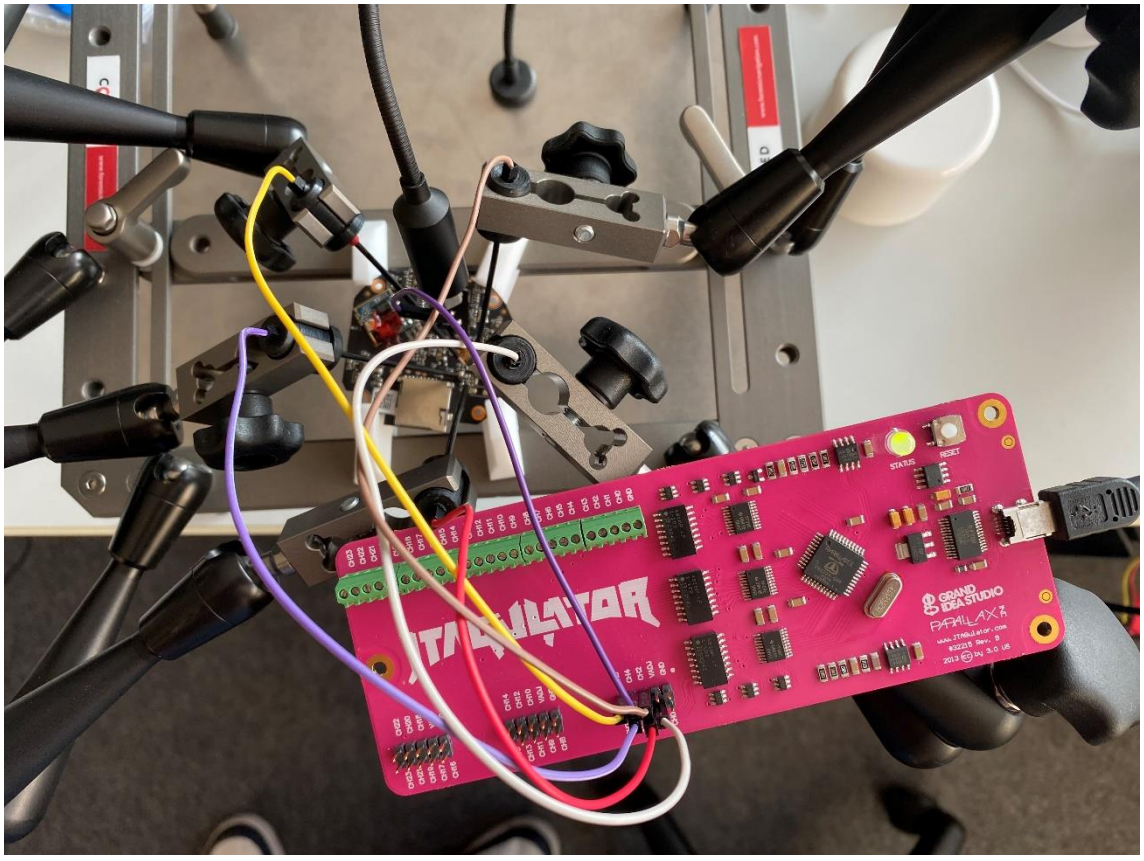


Abbildung 37 - Analyse der Platine der IP-Kamera mit JTAGulator

Die JTAG-Pins wurden zur Überprüfung an den JTAGulator angeschlossen (siehe Abbildung 37). Dieser Versuch blieb ergebnislos. Anschließend wurden die UART-Pins angeschlossen. Hierbei konnte die Sendeleitung (Tx) ermittelt werden. Das Tx-Interface wurde mittels seriellem Interface per USB an einen PC angeschlossen und das Programm Screen gestartet:

screen /dev/ttyUSB0 115200

Als Ergebnis wurde die Bootsequenz der IP-Kamera angezeigt.

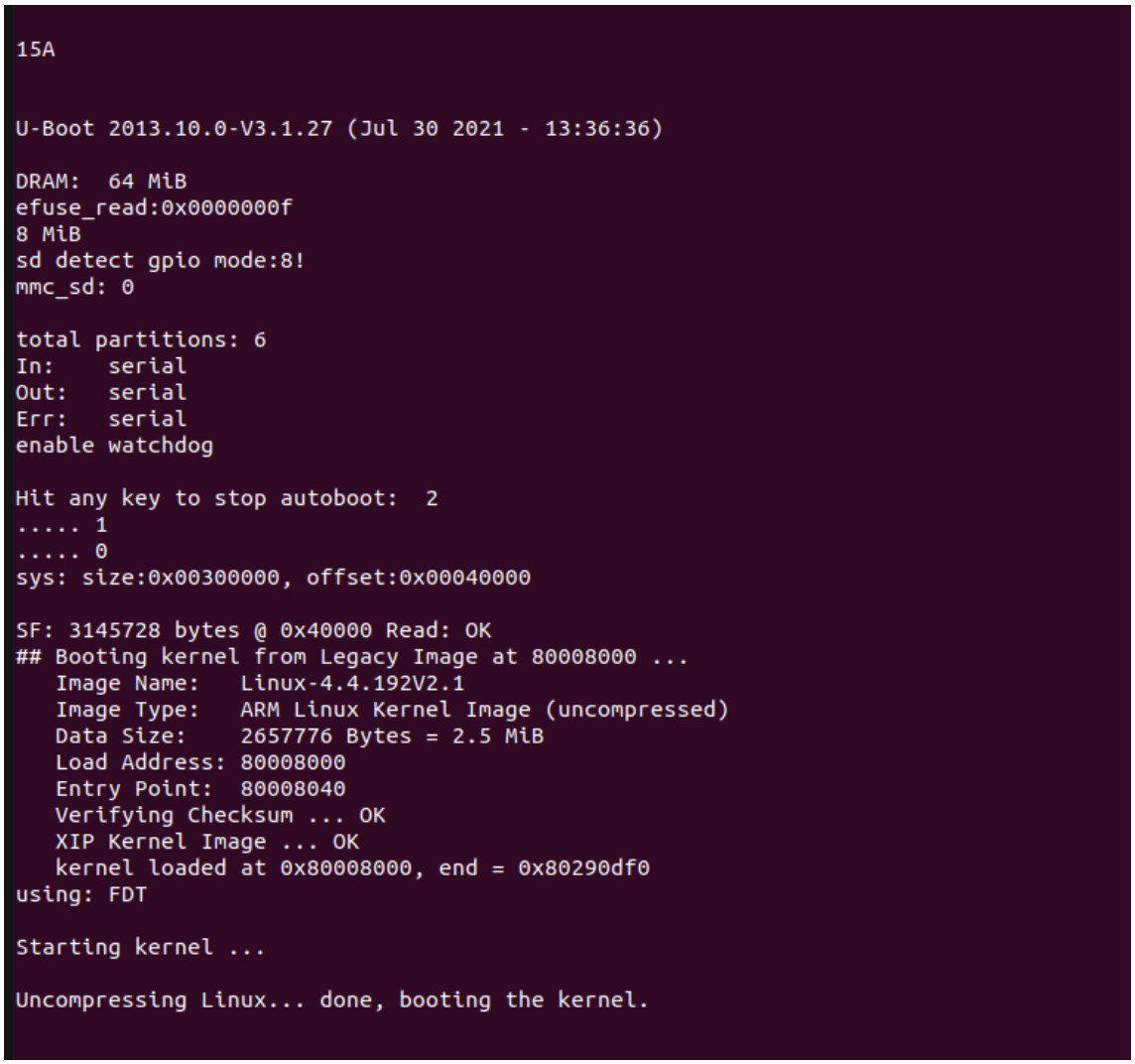


Abbildung 38 – Screenshot: Bootsequenz IP-Kamera

Diese ist in Abbildung 38 dargestellt. Es war zu erkennen, dass der Bootloader auf eine Interaktion wartete. Für diese Interaktion wurde die Receive-Schnittstelle (Rx) benötigt. Mittels JTAGulator wurden die entsprechenden Beine direkt am

Chip verbunden und das Interface erneut überprüft. Wieder verlief die Suche ergebnislos. Eine Spannungsmessung des im Datenblatt als Rx markierten Pins ergab, dass dauerhaft eine Spannung von 0V anlag. Eine Durchgangsmessung mit dem Multimeter bestätigte anschließend, dass der Pin auf Masse gezogen wurde. Demnach war keine Kommunikation mit der UART-Schnittstelle möglich.

4.10.2 Heizungssteuerung

Zunächst wurde die Platine aus dem Gehäuse ausgebaut. Anschließend wurde sich ein Überblick über die Bauteile und Schnittstellen verschafft.

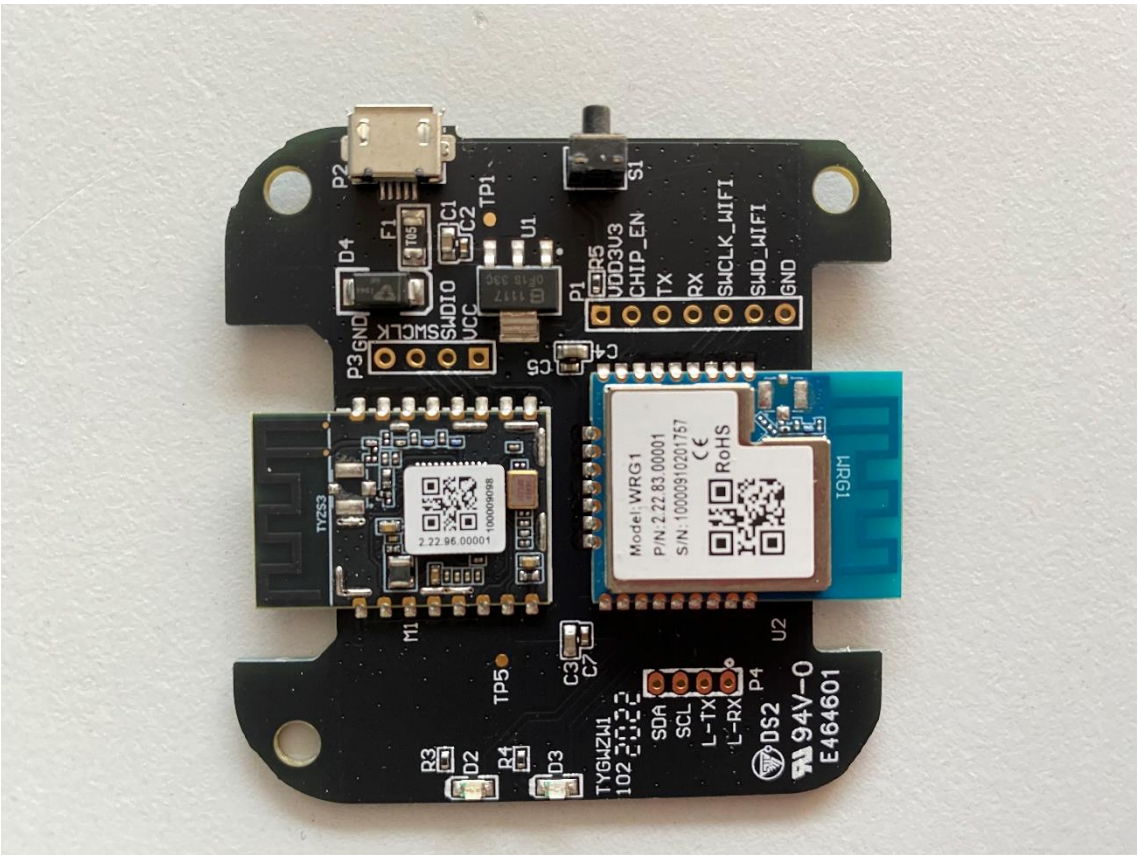


Abbildung 39 - Platine Heizungssteuerung

Die ausführliche Beschriftung der Schnittstellen machte die Suche überflüssig (siehe Abbildung 39). Auf der Platine saßen ein WiFi-fähiges System on a Chip (Tuya WRG1) und ein ZigBee-Chip (Tuya TYZS3). Zunächst wurden die Datenblätter heruntergeladen. Das Datenblatt enthielt unter anderem die UART-

Schnittstelle des WRG1-Chips. Das System wurde mittels RS232-USB-Adapter an einen PC angeschlossen und das Programm Screen gestartet:

screen /dev/ttyUSB0 115200

Abbildung 40 - Output der UART-Schnittstelle des WLAN-Chips

Da der Output in einem unlesbaren Format vorlag (siehe Abbildung 40), wurden zunächst die Standardbaudraten durchprobiert.

Baudraten	
75	28800
300	38400
1200	57600
2400	115200
4800	230400
9600	460800
14400	576000
19200	

Tabelle 13 - Standardbaudraten

Tabelle 13 listet die Standardbaudraten auf. Keine der Baudraten ergab einen lesbaren Output. Anschließend wurden auch die Pins direkt am Chip getestet.

Ihre Position wird im Datenblatt beschrieben. Erneut war die erzeugte Ausgabe nicht lesbar. Auf der Platine befand sich eine weitere Schnittstelle, die in Abbildung 39 mit L-RX und L-TX markiert ist. Laut Datenblatt war es mit dieser Schnittstelle möglich, WLAN-Logdateien mitzulesen. Die Schnittstelle wurde RS232-USB-Adapter an einen PC angeschlossen und das Gerät eingeschaltet.

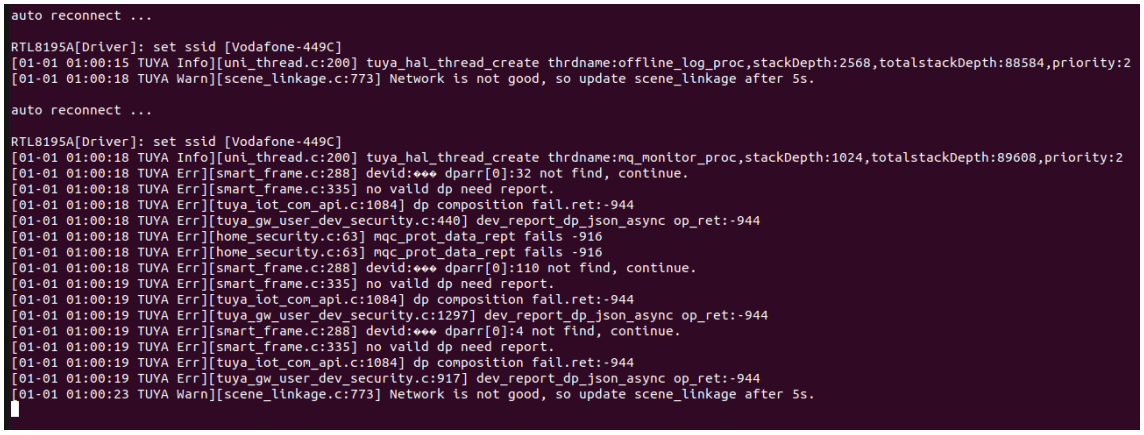


Abbildung 41 – Screenshot: Ausgabe der UART-Logschnittstelle der Heizungssteuerung

Abbildung 41 zeigt einen Ausschnitt der Ausgabe der Loggingschnittstelle. Die Übertragung erfolgte im Klartext. Es wurde ersichtlich, dass das im simulierten Tatort verwendete Netzwerk „Vodafone-449C“ gesucht, jedoch nicht gefunden wurde. Im nächsten Schritt wurde ein Hotspot mit der SSID des Tatortnetzwerks geöffnet. Das gewählte Passwort spielte hier zunächst keine Rolle.

Abbildung 42 – Screenshot: Ausgabe der UART-Logschnittstelle der Heizungssteuerung mit Passwort

Das daraus resultierende Logging ist in Abbildung 42 dargestellt. Das Klartextpasswort des Tatortnetzwerks (PkEcsagJJZnkN9zc) konnte ermittelt werden. Somit wurde eine Möglichkeit gefunden, ein Tatortnetzwerk auch zu simulieren, wenn das Standardpasswort des Routers geändert wurde und das neue Passwort nicht bekannt ist.

5 Diskussion der Ergebnisse

Die vorliegende Masterthesis kann als mehrstufiger Leitfaden für die Sammlung und Auswertung von forensischen Artefakten aus Smart Home Systemen genutzt werden, die als digitale Zeugen einer Straftat in Erscheinung treten ("IoT-as-a-witness"). Diese Artefakte können Rückschluss auf reale Ereignisse und auf das Nutzerverhalten geben. Die erzielten Ergebnisse wurden in mehreren Versuchsreihen geprüft.

Die forensische Analyse von Smart Home Systemen nimmt deutlich mehr Zeit in Anspruch als die klassische IT-Forensik von Datenträgern oder Smartphones. Dieser Umstand ist zum einen der Heterogenität der Systeme geschuldet, zum anderen den fehlenden Untersuchungsstandards und -tools.

Es wurde erwartet, dass sich Geräte an einem Tatort durch die Änderung der Funksignalstärke an mehreren Messpunkten lokalisieren lassen. Diese Hypothese wird durch die erfolgten Messungen gestützt. Die Ergebnisse lassen sich auf physikalische Phänomene wie Absorption, Reflexion und Brechung der Funkwellen zurückführen. Es ist zu beachten, dass ausschließlich die Änderung der Signalstärke zur Lokation dienen kann, nicht aber der Absolutwert. Es wurde festgestellt, dass die Signalstärke sich unterschiedlich stark ändert, obwohl die Geräte sich an der gleichen Position befinden. Eine Effizienzsteigerung lässt sich erzielen, indem die Signalstärke mit mehreren Untersuchungsrechnern an unterschiedlichen Standorten gemessen wird. So kann eine Triangulierung des Funksignals erfolgen. Der Vergleich des Funkanalysetools Kismet mit dem WLAN-Analysetool airodump-ng ergab, dass die beiden Programme am effektivsten gemeinsam genutzt werden. Hierbei kann mit Kismet ein Überblick über den WLAN-Verkehr am Tatort gegeben werden und anschließend mit airodump-ng eine detailliertere Analyse einzelner Access-Points erfolgen. Als problematisch erwiesen sich Geräte im Standby-Modus. Diese konnten, wie am Beispiel des Türsensors gezeigt, durch die Funkanalyse nicht detektiert werden. Der Türsensor wurde ausschließlich dann aktiv, wenn die Gerätefunktion durch Öffnen oder Schließen der Tür ausgelöst wurde. Daher fand sich keine Methode zur Umgehung dieses Problems.

In der Eventlog-Datei des WLAN-Routers vom Tatort wurden Abmeldungen von Geräten aus dem Netzwerk mit Zeitstempel protokolliert. Es wurde ein Rückschluss zur Aktivierung des Türsensors erwartet. Tatsächlich konnten Logeinträge zum Türsensor ermittelt werden, die mit Benachrichtigungen aus der Smartphone-App übereinstimmten. Es wurde jedoch in weiteren Versuchen ersichtlich, dass nur vereinzelt Aktivitäten des Türsensors protokolliert wurden. Die Ursache dieser Beobachtung konnte nicht geklärt werden. Zukünftige Forschungsarbeit kann sich mit der Protokollierung von Sensoraktivitäten anderer Router-Modelle beschäftigen.

Zur Analyse des Netzwerkverkehrs wurde, wie in Kapitel Netzwerkanalyse beschrieben, der Access-Point des Tatortes geklont und in einer Laborumgebung betrieben. Es konnte keine Methode zur spurenschonenden Arbeit im Originalnetzwerk entwickelt werden. Die Geräte meldeten sich wie erwartet am geklonten Access-Point an, da die Identität des Access-Points bei der WPA2-Verschlüsselung nicht überprüft werden kann. Die anschließend aufgezeichneten Netzwerkpakete offenbarten, dass das Hama Smart Home System mit dem MQTT-Protokoll, aufbauend auf TLS-Verschlüsselung betrieben wird. Die MQTT-Broker konnten dem Smart Home Anbieter Tuya Smart zugeordnet werden. Dieser betreibt das Back-End der Smart Home Anwendungen. Es konnten keine Managementschnittstellen (z.B. SSH oder Telnet) der Geräte gefunden werden. Diese werden in anderen Forschungsarbeiten verwendet, um auf das Dateisystem zuzugreifen (siehe Awasthi et al., 2018). Analog zur Funkaufklärung musste auch bei der Netzwerkuntersuchung der Türsensor ausgelöst werden, um eine Verbindung aufzubauen. Hierbei konnte ermittelt werden, dass der Türsensor keine Steuerbefehle entgegennimmt.

Zur Analyse der Applikation auf einem Android-Smartphone wurden zunächst die beiden forensischen Programme Magnet Axiom und Cellebrite UFED miteinander verglichen (siehe Kapitel Appanalyse Hama Smart Home). Da Axiom nur den für Nutzende sichtbaren Speicherbereich sicherte, wurde das durch UFED erzeugte physikalische Speicherabbild zur weiteren Untersuchung verwendet.

Die Analyse des Smartphones, auf dem die Hama Smart App betrieben wurde,

erwies sich als ergiebig (siehe Kapitel Appanalyse Hama Smart Home). Es wurden durch die Benutzung der App zahlreiche Artefakte erzeugt, die auf reale Ereignisse zurückzuführen sind. Zudem boten sich Rückschlüsse auf das Nutzungsverhalten des Smart Home Systems. So konnten Informationen über genutzte WLAN-Netzwerke ermittelt werden. Diese wurden verwendet, um zu ermitteln, wann ein Gerät des Hama Smart Home Systems in die App integriert wurde. Die hierbei genutzte Methode kann jedoch ausschließlich auf Geräte angewendet werden, die im AP-Modus integriert wurden. Weiterhin ließen sich die Benachrichtigungen, die auf dem Smartphone erhalten wurden, in Zusammenhang mit dem Auslösen der Alarmfunktion der IP-Kamera und des Türsensors setzen. Es ließ sich anhand der Datenbankeinträge nicht darlegen, welches Gerät die Benachrichtigung erzeugt hatte. Trotzdem kann die Methode aufschlussreich sein, wenn nur ein einziges Gerät mit einer Alarmfunktion am Tatort betrieben wird. Die Usagestats-Dateien des Android-Smartphones stellten eine Artefaktquelle dar, die Rückschlüsse auf die Nutzung der Applikation gaben. Die Protokolleinträge beinhalteten die aufgerufene Aktivität der App. So konnte beispielsweise das Aufrufen des Streams der IP-Kamera inklusive Zeitstempel nachgewiesen werden. Die durch die App selbst erzeugten Logdateien lagen codiert vor, zukünftig kann versucht werden, diese Dateien zu decodieren. Anhand von Serveranfragen im Cache der App ließ sich der Zugriff auf in der Cloud gespeicherte Aufzeichnungen der IP-Kamera datieren. Hier ließ sich außerdem ermitteln, welche Aufzeichnung des Streams angeschaut wurde. Diese Information ließ sich anhand der URL finden, die den Zeitstempel der gesuchten Aufzeichnung im Unix-Format enthielt. Das Webview-Cache wurde nach einer Methode von Rajewski (2016) entpackt und ausgewertet. Darin befanden sich unter anderem die in der App genutzte E-Mail-Adresse und der Nickname. Nachdem festgestellt wurde, dass der Quellcode der App aus der apk-Datei obfuskiert worden war, wurde die Quellcodeanalyse abgebrochen. Weitere Studien könnten an der Untersuchung des Quellcodes anknüpfen. Hierbei werden Hinweise zur Decodierung der Logdateien erwartet. Diese Logdateien könnten weitere wichtige Zeitstempel zur Appnutzung enthalten.

Wie erwartet blieb die Analyse forensischer Artefakte aus der Cloud ergebnislos (siehe Kapitel Cloudanalyse Hama Smart Home). Es konnten keine Methoden

für den Zugriff auf die in der Cloud gesicherten Daten entwickelt werden. Mehrere wissenschaftliche Arbeiten, z.B. Chung et al. (2017) und Al-Masri et al. (2018), beschreiben ähnliche Hürden der Cloudforensik. Zukünftig müssten technische und juristische Zugriffsmöglichkeiten auf Cloudspeicher geschaffen werden, um diese forensisch auswerten zu können. Ruan et al. (2011) zeigten hierfür bereits im Jahr 2011 mögliche Lösungen auf.

Ziel der Geräteanalyse war der Zugriff und die Auswertung des Gerätespeichers. Bei keinem der untersuchten Geräte war ein Zugriff auf den Gerätespeicher möglich, da weder die UART- noch die JTAG-Schnittstellen den Zugriff ermöglichten (siehe Kapitel Geräteanalyse Hama Smart Home). Die Schnittstellen waren verschlüsselt oder deaktiviert. Einzig auf die Logging-Schnittstelle der Heizungssteuerung konnte ein Lesezugriff erfolgen. Hierbei konnte das Klartextpasswort des WLAN-Netzwerks am Tatort ausgelesen werden. Weitere Forschungsarbeiten könnten sich mit dem Auslöten der Speicherchips befassen.

6 Zusammenfassung

Die vorliegende Masterarbeit beschäftigte sich mit der Sammlung und Analyse von forensischen Artefakten aus Smart Home Systemen. Es wurde zunächst der aktuelle Forschungsstand in einen Kontext gesetzt. Hierbei wurden die Smart Home Systeme als Zeugen einer Straftat (IoT-as-a-witness) und nicht als Angriffsziel oder –werkzeug betrachtet. Anschließend wurden Methoden zum Auffinden und zur Analyse von Artefakten aus Smart Home Systemen beschrieben. Diese Artefakte können Rückschluss auf reale Ereignisse und auf das Nutzerverhalten geben. Für die Analyse wurde ein mehrstufiger Arbeitsablauf entwickelt. Im ersten Schritt wurde gezeigt, wie man Smart Home Geräte an einem Tatort finden kann. Hierbei war festzustellen, dass die Messung der Funksignalstärken an mehreren Messpunkten zum Erfolg führte. Es ist jedoch zu beachten, dass der Absolutwert der Signalstärke keine Aussagekraft über die Entfernung des Gerätes hat. Vielmehr ist die Differenz der Signalstärken an den unterschiedlichen Messpunkten ein Idikator für die Position.

Im nächsten Schritt wurde eine Untersuchung des WLAN-Netzwerks und des Routers am Tatort durchgeführt. Hierbei konnten Logdateien im Router sichergestellt werden, die Hinweise auf die Benutzung von Geräten gaben. Des Weiteren wurden anhand der MAC-Adressen die Hersteller der WLAN-Komponenten ermittelt.

Nachdem die Analyse am simulierten Tatort beendet war, wurden Vergleichsgeräte in einem Labornetzwerk aufgebaut. Hierbei wurde der gesamte Netzwerkverkehr am Router mitgeschnitten. So konnte das Verhalten der Geräte im Netzwerk dokumentiert werden. Auch genutzte Cloudprodukte wurden ermittelt. Das Hama Smart Home System wird mittels MQTT über Server des Herstellers Tuya Smart gesteuert. Unter anderem wurde festgestellt, dass die IP-Kamera über einen Cloudspeicher verfügt.

Im Folgenden wurde die Android App "Hama Smart", sowie ein forensisches Abbild des genutzten Smartphones untersucht. Hierbei konnten Benutzerinformationen zur App gefunden werden. Des Weiteren fanden sich

Datenbanken mit Mitteilungen, die unter anderem Alarmmitteilungen des Türsensors und der IP-Kamera enthielten. Diese gaben Rückschluss auf den Zeitpunkt eines ausgelösten Alarms. Zudem wurden Logeinträge zur Nutzung der Hama Smart App gefunden. Hieraus ließ sich ein umfassendes Bild der Appnutzung erstellen. Auch Anfragen an den Cloudspeicher der IP-Kamera befanden sich im Image des Smartphones. Diese gaben Rückschluss über den Zeitpunkt, wann ein Stream angeschaut wurde.

Die Cloudanalyse erbrachte keine relevanten Ergebnisse.

Bei der Geräteanalyse konnte in der seriellen Loggingschnittstelle des WLAN-Chips der Heizungssteuerung das Klartext –WPA2-Passwort des WLAN-Netzwerks am Tatort ermittelt werden. Dieses konnte anschließend zur Netzwerkuntersuchung genutzt werden.

Zum Abschluss wurden die Methoden und Ergebnisse bewertet.

7 Ausblick

Ermittlungsbehörden verfügen bereits über ein breites Portfolio an Möglichkeiten zur Verwendung forensischer Artefakte aus Smart Home Systemen. In Zukunft werden diese Methoden von digitalen Tatortteams angewendet, die den Erkennungsdiensten direkt angegliedert sind. So kann ein Qualitätsstandard in der Arbeit mit forensischen Artefakten aus Smart Home Systemen garantiert werden.

Des Weiteren müssen für IT-Forensiker*innen Schulungsmöglichkeiten geschaffen werden, um stets auf dem neusten Stand der Technik zu arbeiten. In einigen Bundesländern wurden bereits Trainingszentren mit digitalen Tatorten errichtet, in denen neue Methoden getestet und erlernt werden können. Um die Arbeit mit Smart Home Systemen effizienter zu gestalten, könnten in der Digitalforensik Datenbanken genutzt werden, in denen Untersuchungsmethoden bereits analysierter Geräte gesammelt werden.

Im Bereich der Cloudforensik müssen technische und juristische Rahmenbedingungen zur Akquise forensischer Artefakte durch Ermittlungsbehörden geschaffen werden. Hierzu zählen neben technischen Maßnahmen wie API-Schnittstellen zum Download der Daten auch organisatorische Maßnahmen wie Serviceportale für Strafverfolgungsbehörden.

Weiterhin müssen Standards und technische Möglichkeiten geschaffen werden, um neu entwickelte forensische Software im Bereich Smart Home zu validieren. Hierzu können Datensätze oder vorgegebene Szenarien erstellt werden, um die Zuverlässigkeit der Tools zu erproben und wissenschaftlich zu beschreiben.

8 Literaturverzeichnis

abrignoni.blogspot.com. (17. Februar 2019). Android Usagestats XML Parser, abgerufen am 29. August 2022 von https://abrignoni.blogspot.com/2019/02/android-usagestats-xml-parser.html

accessdata.com. (2022). Product Download, abgerufen am 16. August 2022 von https://accessdata.com/product-download/ftk-imager-version-4-7-1

aircrack-ng.org. (2022). Main Documentation, abgerufen am 21. August 2022 von https://www.aircrack-ng.org/documentation.html

aircrack-ng.org. (09. Februar 2022). Airmon-ng, abgerufen am 20. Juni 2022 von https://aircrack-ng.org/doku.php?id=airmon-ng

aircrack-ng.org/doku.php?id=airodump-ng. (2022). Airodump-ng , abgerufen am 20. Juni 2022 von https://aircrack-ng.org/doku.php?id=airodump-ng

Al-Masri, E., Bai, Y., & Li, J. (2018). A Fog-Based Digital Forensics Investigation Framework for IoT Systems. *IEEE International Conference on Smart Cloud*, 196-201, doi:10.1109/SmartCloud.2018.00040

Anyka. (2022). AK3918 HD IP Camera SoC Specification, abgerufen am 29.08.2022 von http://monitor.espec.ws/files/700cafec77419c2d7705f376c974b8d0ff72_986.pdf

Arshad, H., Jantan, A., & Abiodun, O. I. (2018). Digital Forensics: Review of Issues in Scientific Validation of Digital Evidence. *Journal of Information Processing Systems*, 14(2), 346-376. doi:10.3745/JIPS.03.0095

autopsy.com. (2022). Autopsy, abgerufen am 16. Juni 2022 von https://www.autopsy.com/

Awasthi, A., Read, H. O., Xynos, K., & Sutherland, I. (2018). Welcome pwn:

Almond smart home hub forensics. *Digital Investigation, v. 26,* 38-46, doi:10.1016/j.diin.2018.04.014

azure.microsoft.com. (2022). Erste Schritte mit Azure, abgerufen am 03. Mai 2022 von Erste Schritte mit Azure: https://azure.microsoft.com/de-de/get-started/

Babun, L., Sikder, A. K., Acar, A., & Uluagac, A. S. (2018). IoTDots: A Digital Forensics Framework for Smart Environments. doi:10.48550/arXiv.1809.00745

Bertko, C., & Weber, T. (2017). Smart Home-Funksysteme und -Anbieter im Vergleich. *Home, Smart Home*, 47-181. doi:10.3139/9783446454248.004

Bök, P.-B., Noack, A., Müller, M., & Behnke, D. (2020). Computernetze und Internet of Things, 390-402. Springer Vieweg.doi:10.1007/978-3-658-29409-0

Boztas, A., Riethoven, A., & Roeloffs, M. (2015). Smart TV forensics - Digital traces on televisions. *Digital Investigation, v. 12*, 72-80. doi:10.1016/j.diin.2015.01.012

Briegleb, V. (14. Dezember 2020). *heise.de.* Abgerufen am 14. September 2022 von Crypto Wars: Bitkom fordert "klares Verbot" von staatlichen Backdoors: https://www.heise.de/news/Crypto-Wars-Bitkom-fordert-klares-Verbot-von-staatlichen-Backdoors-4989269.html

bundesnetzagentur.de. (14. Juli 2021). Abgerufen am 20. Juni 2022 von https://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/DE/2021/20210714_WLAN6GHz.html

Burger, R. (15. Oktober 2019). Datenträgerspürhunde - Wie riecht ein USB-Stick? *Frankfurter Allgemeine Zeitung*.

cellebrite.com. (2022). Cellebrite UFED, abgerufen am 20. August 2022 von UFED: https://cellebrite.com/en/ufed/

Chung, H., Park, J., & Lee, S. (2017). Digital forensic approaches for Amazon

Alexa ecosystem. *Proceedings of the Seventeenth Annual DFRWS USA*, 15-25. doi:10.1016/j.diin.2017.06.010

cloud.google.com. (2022). Google Cloud Infrastructure, abgerufen am 03. Mai 2022 von https://cloud.google.com/infrastructure/

Copos, B., Levitt, K., Bishop, M., & Rowe, J. (2016). Is Anybody Home? Inferring Activity From Smart Home Network Traffic. *IEEE Symposium on Security and Privacy Workshops.* doi:10.1109/SPW.2016.48245

de.hama.com/produkte/smart-home. (2022). Smart Home Geräte, abgerufen am 14. April 2022 von https://de.hama.com/produkte/smart-home

developer.android.com. (2022). Android Studio, abgerufen am 12. August 2022 von Android Studio: https://developer.android.com/studio/

developer.android.com(1). (2022). Fundamentals, abgerufen am 15. Juli 2022 von https://developer.android.com/guide/components/fundamentals

developer.android.com(2). (2022). Manifest Intro, abgerufen am 16. Juli 2022 von Manifest-intro: https://developer.android.com/guide/topics/manifest/manifest-intro

developer.android.com(3). (2022). App data and files, abgerufen am 16. Juli 2022 von Data: https://developer.android.com/guide/topics/data

developers.google.com. (2022). Protocol Buffers , abgerufen am 28. August 2022 von https://developers.google.com/protocol-buffers

Dorai, G., Houshmand, S., & Baggili, I. (2018). I Know What You Did Last Summer: Your Smart Home Internet of Things and Your iPhone Forensically Ratting You Out. *In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES 2018) Association for Computing Machinery, art. 49, 1-10.* doi:10.1145/3230833.3232814

elektronik-kompendium.de. (2022). Bluetooth Low Energy (4.0 / 4.1 / 4.2), abgerufen am 24. Juni 2022 von https://www.elektronik-

kompendium.de/sites/kom/1805171.htm

elektroniknet.de. (2022). Abgerufen am 25. Juni 2022 von https://www.elektroniknet.de/kommunikation/wireless/bluetooth-low-energy-in-smartphones-wie-funktioniert-das.103061.html

embeddedbits.org. (2021). Extracting Firmware from Devices using JTAG , abgerufen am 30. August 2022 von https://embeddedbits.org/2020-02-20-extracting-firmware-from-devices-using-JTAG/

fcc.gov. (2022). What we do, abgerufen am 25. Juni 2022 von https://www.fcc.gov/about-fcc/what-we-do

fccid.io. (2022). Arlo Technologies, Arlo Base Station, abgerufen am 25. Juni 2022 von https://fccid.io/2APLE18300391

Georgiev, A., & Schlögl, S. (2018). Smart Home Technology: An Exploration of End User Perceptions. *Smarter Lives*, S. 64-78.

Gessler, R., & Krause, T. (2015). Wireless-Netzwerke für den Nahbereich. Wiesbaden: *Springer Vieweg*. doi:10.1007/978-3-8348-2075-4

github.com/abrignoni. (2020). Android Usagestats XML & Protobuf Parser, abgerufen am 28. August 2022 von https://github.com/abrignoni/Android-Usagestats-XML-Protobuf

github.com/aosp-mirror. (2022). Plattform System Core, abgerufen am 28. August 2022 von https://github.com/aosp-mirror/platform_system_core

github.com/pxb1988. (2021). dex2jar, abgerufen am 30. August 2022 von https://github.com/pxb1988/dex2jar

github.com/riverloopsec. (2022). Killerbee, abgerufen am 25. August 2022 von Killerbee: https://github.com/riverloopsec/killerbee

grandideastudio.com. (2022). JTAGulator, abgerufen am 30. August 2022 von http://www.grandideastudio.com/JTAGulator/

greatscottgadgets.com. (2021). Ubertooth One, abgerufen am 24. Juni 2022 von https://greatscottgadgets.com/ubertoothone/

Gruschka, P. (2021). *Studienbrief Sicherheit im Cloud-Computing.* WINGS - Wismar International Graduation Services GmbH.

Gupta, A. (2019). The IoT Hacker's Handbook - A Practical Guide to Hacking the Internet of Things. Walnut, CA, USA: *Apress Media LLC*.

Hahn, K. A. (2017). Der "Smart-Ort" als Tatort. Die Kriminalpolizei - Zeitschrift der Gewerkschaft der Polizei.

Hannan Bin Azhar, M., & Bate, S. (2019). Recovery of Forensic Artefacts from a Smart Home IoT Ecosystem. *The Fourth International Conference on Cyber-Technologies and Cyber-Systems CYBER*, S. 94-99.

Hildayanti, N., & Riadi, I. (2019). Forensics Analysis of Router On Computer Networks Using Live Forensics Method. *International Journal of Cyber-Security and Digital Forensics*, 74-81.

Hutchinson, S., Yoon, Y., Shantaram, N., & Karabiyik, U. (2020). Design, Implementation and Analysis of Smart Home Laboratory. *ASEE Virtual Annual Conference Content Access.* doi:10.18260/1-2–34868

informit.com. (09. Dezember 2009). Introduction to the ZigBee Wireless Sensor and Control Network, abgerufen am 06. August 2022 von https://www.informit.com/articles/article.aspx?p=1409785&seqNum=4

iot.tuya.com. (2022). Tuya Smart Developer Center, abgerufen am 09. September 2022 von https://iot.tuya.com/cloud/products?productType=all

java-decompiler.github.io. (2022). Java Decompiler, abgerufen am 30. August 2022 von https://java-decompiler.github.io/

Kebande, V. R., & Ray, I. (2016). A Generic Digital Forensic Investigation Framework for Internet of Things (IoT). *IEEE 4th International Conference on Future Internet of Things and Cloud*, 356-362. doi:0.1109/FiCloud.2016.57

Kim, S., Park, M., Lee, S., & Kim, J. (2020). Smart Home Forensics - Data Analysis of IoT Devices. *Electronics 2020*, 9, 1215, doi: 10.3390/electronics9081215.

kismetwireless.net. (2022). Kismet Documentation, abgerufen am 20. Juni 2022 von https://www.kismetwireless.net/docs/

Klöß, D., & Gentemann, L. (2020). Das intelligente Zuhause: Smart Home 2020. Berlin: *Bitkom e. V. - Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.*

Li, S., Choo, K.-K. R., Sun, Q., Buchanan, W. J., & Cao, J. (2015). IoT Forensics - Amazon Echo as a Use Case. *IEEE Internet of Things Journal*, 6 (4), pp. 6487-6497. doi:10.1109/JIOT.2019.2906946

Losavio, M. M., Chow, K., Koltay, A., & James, J. (2018). The Internet of Things and the Smart City: Legal challenges with digital forensics, privacy, and security. *Security Privacy 2018-1*. doi:10.1002/spy2.23

Luber, S., & Donner, A. (11. November 2019). Was ist eine Fritzbox?, abgerufen von https://www.ip-insider.de/was-ist-eine-fritzbox-a-883753/

macvendor.info. (2022). MAC address assignment lookup, abgerufen am 13. September 2022 von https://macvendor.info/

magnetforensics.com. (2022). Advanced Mobile Acquisition for Android , abgerufen am 21. August 2022 von https://www.magnetforensics.com/resources/advancedmobile/

magnetforensics.com(1). (2022). Cloud-Sicherung und -analyse leicht gemacht, abgerufen am 16. Mai 2022 von https://www.magnetforensics.com/de/resources/cloud-erfassung-und-analyse-leicht-gemacht/

makeuseof.com. (04. August 2021). What is Android System WebView And What Does It Do?, abgerufen am 30. August 2022 von https://www.makeuseof.com/what-is-android-system-webview/

NIST. (2011). The NIST Definition of Cloud Computing. Gaithersburg.

nmap.org. (2022). nmap, abgerufen am 16. August 2022 von https://nmap.org/book/man.html

null-byte.wonderhowto.com. (2016). What Is Tcpwrapped? How to Bypass it?, abgerufen am 29. August 2022 von: https://null-byte.wonderhowto.com/forum/what-is-tcpwrapped-bypass-it-0169685/

nvd.nist.gov. (2022). National Vulnerability Database, abgerufen am 25. Juni 2022 von https://nvd.nist.gov/vuln/search

openocd.org. (2022). OpenOCD Documentation, abgerufen am 18. August 2022 von https://openocd.org/pages/documentation.html

openwrt.org. (22. Juni 2022). Fritz.box.4020, abgerufen am 13. Juli 2022 von: https://openwrt.org/toh/avm/fritz.box.4020

Perumal, S., Norwawi, N., & Raman, V. (2015). Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology. *Fifth International Conference on Digital Information Processing and Communications (ICDIPC).* doi:10.1109/ICDIPC.2015.7323000

Rajewski, J.T. (2016). Internet of things forensics. *Enfuse 2016, Presentation*

raspberrypi.com. (2020). Remote Access, abgerufen am 14. August 2022 von https://www.raspberrypi.com/documentation/computers/remote-access.html

researchgate.net. (2022). Researchgate, abgerufen am 25. Juni 2022 von www.researchgate.net

rfwireless-world.com. (2022). BLE Advertising channels and data channels list, abgerufen am 24. Juni 2022 von https://www.rfwireless-world.com/Terminology/BLE-Advertising-channels-and-Data-channels-list.html

Ruan, K., Carthy, J., & Kechadi, T. (2011). Cloud Forensics: An overview. In:

Peterson, G., Shenoi, S. (eds) Advances in Digital Forensics VII. DigitalForensics 2011. IFIP Advances in Information and Communication Technology, Springer, vol 361, doi:10.1007/978-3-642-24212-0_3

Sadineni, L., Pilli, E., & Battula, R. B. (2019). A Holistic Forensic Model for the Internet of Things. *Advances in Digital Forensics XV, Springer*, 3-14. doi:10.1007/978-3-030-28752-8

Salamh, F. E. (2020). A Forensic Analysis of Home Automation Devices (FAHAD) Model: Kasa Smart Light Bulb and Eufy Floodlight Camera as Case Studies. *International Journal of Cyber Forensics and Advanced Threat Investigations*, doi: 10.46386/ijcfati.v1i1-3.16 .

Schemberg, A., Linten, M., & Surendorf, K. (2019). PC-Netzwerke - Das umfassende Handbuch. Bonn: *Rheinwerk.*

scholar.google.com. (2022). Search: Hama Smart, abgerufen am 03. 08 2022 von https://scholar.google.com/scholar?hl=de&as_sdt=0%2C5&q=hama+smart&btnG=

scholar.google.com(1). (2022). Search: Arlo Base Station, abgerufen am 25. Juni 2022 von https://scholar.google.com/scholar?hl=de&as_sdt=0%2C5&q=arlo+base+station&btnG=

Scientific Working Group on Digital Evidence. (2020). SWGDE Technical Notes on Internet of Things (IoT) Devices.

search.censys.io. (2022). Censys Search, abgerufen am 30. August 2022 von Censys: https://search.censys.io/

Servida, F., & Casey, E. (2019). IoT forensic challenges and opportunities for digital traces. *Digital Investigation*, *28, Supplement, April 2019, 22-29*.

Simou, S., Kalloniatis, C., Kavakli, E., & Gritzalis, S. (2014). Cloud Forensics: Identifying the Major Issues and Challenges. In: M. Jarke et al. (Eds.):

CAiSE 2014, LNCS 8484, pp. 271–284, 2014. *Springer International Publishing Switzerland.*

Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys & Tutorials Vol. 22*, 1191-1221. doi:10.1109/COMST.2019.2962586

support.tuya.com. (2022). Abgerufen am 21. April 2022 von https://support.tuya.com/en/help/_detail/K9hut3pgw8va5

tasteofthewildpetfood.com. (25. Februar 2021). Abgerufen am 03. Juli 2022 von Helping Solve Digital Crimes: Electronic Storage Detection Dogs: https://www.tasteofthewildpetfood.com/working-dogs/dogs-helping-solve-digital-crimes/

Technicolor. (2019). *Installations- und Benutzerhandbuch CGA6444VF.*

Tekeoğlu, A., & Tosun, A. Ş. (2015). Investigating Security and Privacy of a Cloud-Based Wireless IP Camera - NetCam. *24th International Conference on Computer Communication and Networks (ICCCN).* doi:10.1109/ICCCN.2015.7288421

Texas Instruments. (2022). ZigBee Packet Sniffer, abgerufen am 06. August 2022 von https://www.ti.com/tool/PACKET-SNIFFER

tutorial-reports.com. (2022). Wireless LAN (Wifi) Tutorial, abgerufen am 21. Juni 2022 von http://www.tutorial-reports.com/wireless/wlanwifi

tuya.com. (2022). All-in-One App, abgerufen am 16. Mai 2022 von https://www.tuya.com/product/app-management/all-in-one-app

tuya.com(1). (2022). IP-Camera , abgerufen am 13. Mai 2022 von https://www.tuya.com/solution/hardware/ip-camera

ultratecusa.com. (2020). Ultrapol Advance, abgerufen am 30. August 2022 von https://www.ultratecusa.com/product/ultrapol-advance/

Vishwakarma, G., & Lee, W. (2018). Exploiting JTAG and Its Mitigation in IOT: A Survey. *Future Internet* 10(12), 121. doi:10.3390/fi10120121

wiki.sleuthkit.org. (13. Januar 2014). fls, abgerufen am 23. August 2022 von https://wiki.sleuthkit.org/index.php?title=Fls

wiki.sleuthkit.org. (07. März 2015). Filesystem Analysis, abgerufen am 16. August 2022 von https://wiki.sleuthkit.org/index.php?title=FS_Analysis

wireshark.org. (2022). About Wireshark, abgerufen am 16. August 2022 von Wireshark: https://www.wireshark.org/#learnWS

Wu, T., Breitinger, F., & Baggili, I. (2019). IoT Ignorance is Digital Forensics Research Bliss: A Survey to Understand IoT Forensics Definitions, Challenges and Future Research Directions. *Ares 2019 - American Real Estate Research*. doi:10.1145/3339252

xJTAG.com. (2022). JTAG - A technical Overview , abgerufen am 30. August 2022 von https://www.xJTAG.com/de/about-JTAG/JTAG-a-technical-overview/

Yaron, O., & Benjakob, O. (25. April 2021). 'Stop Using Cellebrite': Israeli, U.K. Police Urged to Stop Using Phone-hacking Tech. *Haaretz*. Abgerufen am 26. Juli 2022 von Haaretz: https://www.haaretz.com/israel-news/tech-news/2021-04-25/ty-article/.premium/israeli-u-k-police-urged-to-stop-using-cellebrite-phone-hacking-tech/0000017f-e697-dc7e-adff-f6bf8aed0000

Zawoad, S., & Hasan, R. (2015). FAIoT: Towards Building a Forensics Aware Eco System for the Internet of Things. 2015 IEEE International Conference on Services Computing, 2015, pp. 279-284, doi:10.1109/SCC.2015.46

ZigBee Alliance. (2015). ZigBee Specification, abgerufen am 16. Juni 2022 von https://zigbeealliance.org/wp-content/uploads/2019/12/docs-05-3474-21-0csg-zigbee-specification.pdf

9 Abbildungsverzeichnis

10 Tabellenverzeichnis

11 Thesen

1. Die Akquise forensischer Artefakte aus Smart Home Systemen ist deutlich zeitaufwändiger als die Akquise in der klassischen IT-Forensik

2. Gut erprobte Standardforensiktools können im Bereich Smart Home kaum genutzt werden.

3. Die fünfstufige Analysemethode aus: OSINT-, Netzwerk-, App-, Cloud- und Geräteanalyse wird in wissenschaftlichen Arbeiten vielfach genutzt und ist der derzeitige Stand der Technik.

4. Die Position von Smart Home Geräten am Tatort kann anhand der Änderung ihrer Signalstärke ermittelt werden.

5. Das Smart Home System von Hama liefert Nutzerdaten, Nutzungsdaten und Benachrichtigungsdaten, die reale Ereignisse abbilden können.

6. Cloudanwendungen stellen die IT-Forensik im Bereich Smart Home vor Herausforderungen. Die Sammlung forensischer Artefakte in der Cloud ist ohne Durchsuchungsbeschluss kaum möglich.

12 Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.
Die Stellen der Arbeit, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Ich erkläre ferner, dass ich die vorliegende Arbeit in keinem anderen Prüfungsverfahren als Prüfungsarbeit eingereicht habe oder einreichen werde.

Die eingereichte schriftliche Fassung entspricht der auf dem Medium gespeicherten Fassung.

Ort, Datum (Unterschrift)