HOCHSCHULE
MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Frau
Nele Christin Schnaubelt

Virtualisierung von gesicherten Apple Systemen mithilfe von VirtualBox

Mittweida, September 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Virtualisierung von gesicherten Apple Systemen mithilfe von VirtualBox

Autorin:
Nele Christin Schnaubelt

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO20w6-B

Erstprüfer:
Prof. Dr. rer. nat. Spranger

Zweitprüfer:
Müting, Dipl. Inf.

Einreichung:
Mittweida, 29.09.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Virtualization of backed up Apple systems for forensic evaluation using VirtualBox

Author:
Nele Christin Schnaubelt

Course of Study:
General and Digital Forensic Science

Seminar Group:
FO20w6-B

First Examiner:
Prof. Dr. rer. nat. Spranger

Second Examiner:
Müting, Dipl. Inf.

Submission:
Mittweida, 29.09.2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung

Schnaubelt, Nele Christin:
Virtualisierung von gesicherten Apple Systemen mithilfe von VirtualBox. – 2023. – 49 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Referat

In dieser Arbeit wurde sich mit der Virtualisierung von gesicherten Apple-Systemen beschäftigt. Dafür wurden fünf unterschiedliche Sicherungen von drei verschiedenen Apple-Rechnern versucht zu virtualisieren. Dazu wurden verschiedene Ansätze bei den unterschiedlichen Sicherungen ausprobiert. Dabei wurde festgestellt, dass es zum einen nicht möglich ist, die gesamte Festplatte eines Rechners mit APFS und zum anderen einen APFS-Container, der mit DigitalCollector gesichert wurde, zu virtualisieren. Es ist jedoch möglich, einen APFS-Container von einem Apple-Rechner mit Intel-Chip ohne T2-Chip zu virtualisieren, sowie das Daten-Volume und die Ordner User, Applications und System. Bei den funktionierenden Virtualisierungen wurde mithilfe von Hashwerten der Dateien vor und nach der Virtualisierung die Datenintegrität überprüft. Als Ergebnis wurde festgestellt, dass keine Sicherung vor und nach der Virtualisierung zu 100 % übereinstimmt, da Dateien, die wichtig fürs System sind verändert wurden. Diese hatten jedoch keinen Einfluss auf die Sichtung der Daten des Nutzers des gesicherten Rechners. Aus dem Grund, dass nicht alle Dateien unverändert blieben, sollte die Virtualisierung im Zusammenhang mit anderen Datenaufbereitungsprogrammen verwendet werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AES ..... Advanced Encryption Standard

AFF ..... Advanced Forensic Format

AFF4 ..... Advanced Forensics File Format

APFS ..... Apple File System

CDR ..... Compact Disc Recordable

CPU ..... Central Processing Unit

DMA ..... Direct Memory Access

DMG ..... Disk Image

EWF ..... Expert Witness Format

GID ..... Gerätegruppen ID (engl. Group Identifier)

GUID ..... Globally Unique Identifier

HFS+ ..... Hierarchical File System Plus

SoC ..... System on Chip

UID ..... eindeutige Mac ID (engl. Unique Identifier)

VDI ..... Virtual Disk Image

VHD ..... Virtual Hard Disk

VM ..... Virtuelle Maschine

VMDK ..... Virtual Machine Disk

Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei meinem Hochschulbetreuer Herrn Spranger für die Betreuung meiner Bachelorarbeit bedanken. Dazu möchte ich mich bei Frau Felser für die konstruktive Kritik bei der Erstellung meiner Bachelorarbeit bedanken.
Ein besonderer Dank gilt dem Dezernat 63.1 vom LKA Niedersachsen für ihre herzliche Betreuung meiner Bachelorarbeit und Bereitstellung der technischen Geräte. Vor allem möchte ich bei meinem Betreuer Herrn Müting bedanken, der mir bei Fragen und Problemen immer zur Seite stand.
Des Weiteren möchte ich mich bei meiner Familie und meinen Freunden für ihre Hilfsbereitschaft bedanken und die zahlreichen Diskussionen die wir geführt haben, um auf neue Ideen zu kommen. Insbesondere möchte ich mich bei meinem Freund bedanken, der mich bei allem unterstützt hat.

1 Einleitung

Die rasante Weiterentwicklung moderner Technologien stellt die IT-Forensik der Polizei vor stetig wachsende Herausforderungen. In diesem Kontext sind kontinuierliche Verbesserungen bei der Datensicherung und -aufbereitung unerlässlich, da eine Vielzahl von Rechnern mit unterschiedlicher Hardware und Software im Einsatz ist. Die drei in Deutschland am meisten verwendeten Laptops gehören zu den Marken Acer, Lenovo und Apple [31]. Die Abbildung 1.1 zeigt die zehn meistverwendeten Rechner in Deutschland im Jahr 2023. Apple gehört mit einem Anteil von 17 % zu den meistgenutzten Computermarken in Deutschland [31]. Lediglich die Marken Lenovo mit 19 % und Acer mit 34 % wurden häufiger benutzt [31].

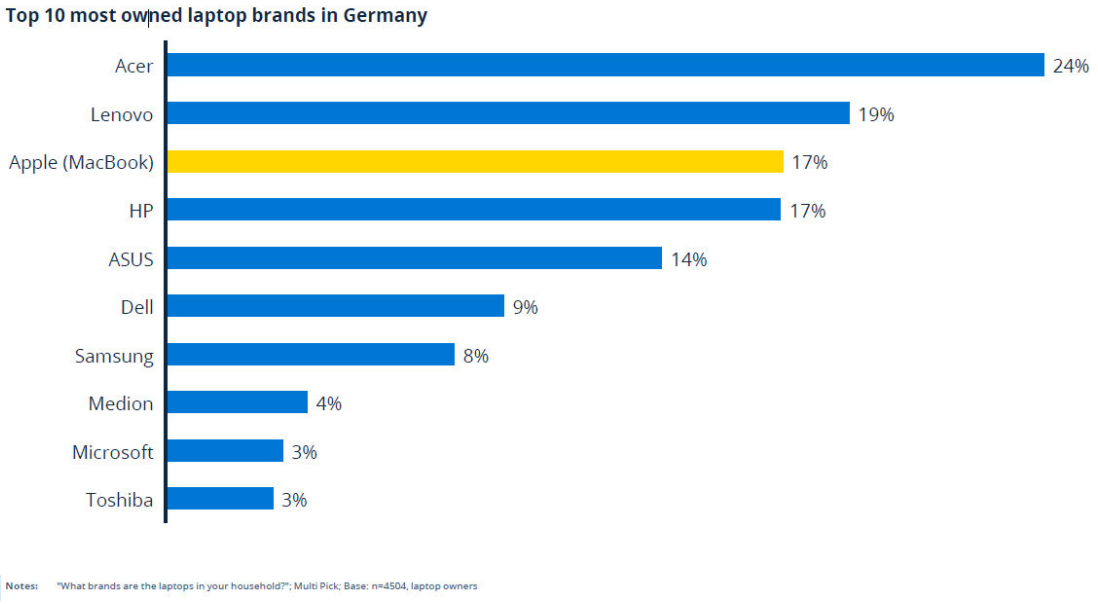


Abbildung 1.1: Die Grafik zeigt die 10 Laptopmarken, die am meisten in Deutschland verwendet werden. Die Stichprobe beträgt $n = 4504$. (Quelle: [31])

Aufgrund dessen, dass Apple einen signifikanten Marktanteil im Bereich der Computertechnologie ein nimmt [31], ist Apple ein wichtiger Teil des Bereichs IT-Forensik der Polizei. Dazu kommt, dass insbesondere Apple-Geräte sich aufgrund der ständigen Einführung neuer Sicherheitsfunktionen als besonders anspruchsvoll erweisen [66].

Eine wichtige Aufgabe im Bereich der IT-Forensik der Polizei ist die Datenaufbereitung der gesicherten Daten eines sichergestellten Rechners. Momentan kommen für die Datenaufbereitung verschiedene forensische Softwarelösungen wie X-Ways Forensics, Magnet Forensics Axiom oder Encase zum Einsatz, die Ermittlern die Möglichkeit bieten, auf dem Rechner vorhandene oder ehemals vorhandene Daten zu analysieren [38] [66]. Dennoch können diese Lösungen nicht immer den gleichen

Einblick in die Daten bieten, wie ihn der Benutzer des ursprünglichen Geräts hatte.

Die Virtualisierung stellt einen zusätzlichen Ansatz zur Untersuchung von gesicherten Computern dar und bietet den Vorteil, Live-Untersuchungen durchführen zu können, ohne die Integrität des zu untersuchenden Rechners zu beeinträchtigen. Dies ermöglicht Ermittlern, das System so zu betrachten, wie es der ursprüngliche Benutzer gesehen hat. Nicht nur dem Ermittler bietet dies ein besseres Verständnis für die vorliegenden Daten, sondern auch Richtern und Anwälten. Diese benötigen lediglich ein grundlegendes Verständnis über die Bedienung eines Rechners, um den Sachverhalt nachvollziehen zu können.

Aktuell sind wenige Programme verfügbar, die es ermöglichen, bestimmte bootfähige Images von Apple-Systemen zu virtualisieren [17] [43]. Allerdings können damit nur bootfähige Sicherungen virtualisiert werden, weshalb beispielsweise keine Virtualisierungen von gesicherten Apple File System (APFS)-Containern möglich sind [17] [43]. Außerdem sind diese Programme kostenpflichtig und können deshalb nicht ohne weiteres eingesetzt werden.

Das Hauptziel dieser Arbeit ist daher die Entwicklung einer kostenlosen Methode zur Virtualisierung von gesicherten Apple-Systemen mithilfe von VirtualBox [61]. Hierfür werden verschiedene Konfigurationen unterschiedlicher Apple-Systeme mit ihren unterschiedlichen Sicherungsmethoden verwendet. Dabei wird sich auf das APFS beschränkt, da es mittlerweile weit verbreitet und auf den meisten gesicherten Rechnern anzutreffen ist. Angesichts der hardwarebasierten und softwarebasierten Sicherheitsmerkmalen von Apple ist es nicht immer möglich, bootfähige Systeme zu sichern. Daher wird in dieser Arbeit vor allem auch die Virtualisierung von nicht bootfähigen Datensicherungen behandelt.

In der Arbeit wird zunächst in Abschnitt 2.1 auf die unterschiedlichen Konfigurationen von Apple-Systemen eingegangen. Dabei wird die Entwicklung der Sicherheitsfeatures im Bereich der Hard- und Software behandelt, sowie die Entwicklung der Betriebssysteme für Apple-Rechner. Anschließend wird in Abschnitt 2.2 die Datensicherung von Apple-Rechnern und die anschließende Datenaufbereitung dargestellt. Danach wird auf den Recovery-Mode von Apple-Rechnern eingegangen. Zum Schluss des Kapitels wird in Abschnitt 2.4 das Thema Virtualisierung und die Virtualisierungssoftware VirtualBox behandelt.
Nach dem theoretischen Teil dieser Arbeit wird in Kapitel 3 auf die Entwicklung der Methoden zur Virtualisierung eingegangen. Anschließend werden in Kapitel 4 die Ergebnisse diskutiert. Zum Abschluss erfolgt in Kapitel 5 das Fazit und der Ausblick der Arbeit.

2 Grundlagen

In diesem Kapitel wird auf die für die Arbeit benötigten Grundlagen eingegangen. Dabei werden die unterschiedlichen Konfigurationen von Apple-Rechnern inklusive der Entwicklung der Betriebssysteme behandelt. Anschließend wird auf die momentanen Arten der Datensicherung und Datenaufbereitungsmethoden eingegangen. Zum Abschluss wird das Thema Virtualisierung und explizit die Virtualisierungssoftware VirtualBox [61] erklärt.

2.1 Unterschiedliche Konfigurationen von Apple-Rechnern

Apple hat im Laufe der Jahre ihre Systeme stets erneuert und revolutioniert. Sie haben neuere hardwarebasierte Sicherheitsfeatures eingeführt und dafür Apple eigene Chips entwickelt [3] [4] [5]. Neben der Verbesserung der Hardware werden auch neue Betriebssysteme mit neuen Sicherheitsfeatures entwickelt [6]. Diese unterschiedlichen Konfigurationen beeinflussen nicht nur die Methode der Sicherung, sondern auch der Virtualisierung. Je nach Art der gerätespezifischen und betriebssystemspezifischen Sicherheitsfeatures müssen unterschiedliche Dinge bei der Virtualisierung beachtet werden.

2.1.1 Entwicklung der Sicherheitsfeatures

Die erste Möglichkeit der Verschlüsslung von Daten wurde im Jahr 2003 mit FileVault eingeführt [8]. FileVault bietet die Möglichkeit den home-Ordner zu verschlüsseln [8]. Mit macOS Lion wurde im Februar 2011 FileVault 2, im weiteren Verlauf als FileVault bezeichnet, eingeführt [6]. Es verschlüsselt im Gegensatz zu der früheren Version das gesamte Volume, einschließlich des Betriebssystems [8].

Im Oktober 2016 wurde der T1-Chip vorgestellt [50]. Die Hauptaufgabe des T1-Chips ist die Verarbeitung der Touch-ID zu regeln [50]. Dazu soll der T1-Chip sensitive Komponenten wie die Kamera und das Mikrofon schützen und überprüfen, ob macOS auf Apple-Hardware läuft [50].

Nach der Einführung vom T1-Chip wurde kurz darauf im Jahre 2017 der T2-Chip eingeführt [25]. In Abbildung 2.1 werden die Komponenten des T2-Chips dargestellt. Er besitzt einen Secure Enclave Coprozessor, der Grundlage für die Verschlüsselung des APFS-Speichers, den Secure-Boot und die Touch-ID ist [5]. Der Secure Encalve Coprozessor schützt die benötigten kryptografischen Schlüssel, die für FileVault und

den Secure-Boot benötigt werden [5]. Dazu verwaltet er die Fingerabdruck-Daten des Touch-ID-Sensors und bestimmt, ob ein Treffer vorliegt [5]. Er nutzt verschlüsselten Speicher und beinhaltet einen Zufallszahlengenerator [5]. Neben dem Secure Encalve Prozessor besitzt er eine Advanced Encryption Standard (AES) Crypto Engine, wie in Abbildung 2.1 zu erkennen, die in dem Direct Memory Access (DMA) zwischen der Central Processing Unit (CPU) und dem Flash-Speicher liegt [5]. Es verschlüsselt das komplette Volume mithilfe von AES-XTS [5]. Dadurch sorgt es dafür, dass alle Daten im Standby oder ausgeschaltetem Zustand verschlüsselt sind [5].

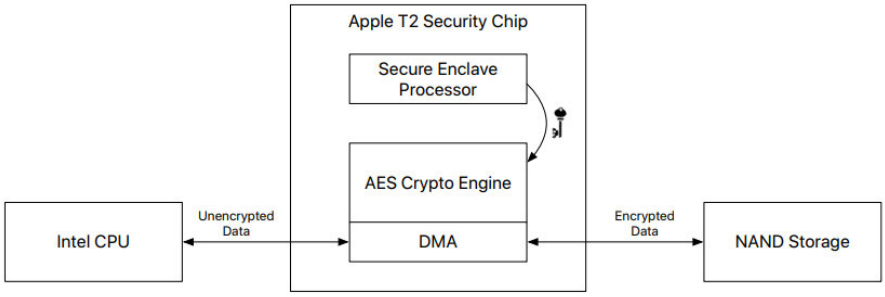


Abbildung 2.1: Komponenten des Apple T2-Chips und ihre Verbindungen mit anderen Komponenten. Der T2-Chip besteht aus dem Secure Enclave Prozessor, der AES Crypto Engine im DMA. (Quelle: [5])

Die eindeutige Mac ID (engl. Unique Identifier) (UID) und die Gerätegruppen ID (engl. Group Identifier) (GID) werden während der Herstellung jeweils mit einem 256-Bit Schlüssel in die Secure Enclave eingebunden [5]. Die UID steht in keinem Bezug zu einer anderen ID oder Kennung des Gerätes [49]. Durch die UID können Daten an ein bestimmtes Gerät gebunden werden [49]. Dies führt dazu, dass nicht auf die Daten zugegriffen werden kann, wenn der Speicherchip vom Gerät getrennt und ausgelesen wird [49]. Die GID ist für alle Geräte gleich, die ein bestimmtes System on Chip (SoC) besitzen [49]. Die AES Crypto Engine unterstützt sowohl Hardware- als auch Softwareschlüssel. Die Hardwareschlüssel bilden sich aus der UID und der GID [49]. Keine Software oder Firmware kann direkt auf die Schlüssel zugreifen und diese auslesen [5]. Die Schlüssel können nur über die AES Crypto Engine verwendet werden, die auf die Secure Enclave zugreifen kann [5]. Die AES Crypto Engine stellt lediglich die Ergebnisse der Verschlüsselung oder Entschlüsselung zur Verfügung, nicht aber die verwendeten Schlüssel selbst [5]. Durch die Aktivierung von FileVault wird die Sicherheit weiter erhöht, da nun zur Entschlüsselung der Daten, die Eingabe des Anmeldepasswortes benötigt wird [56]. Bei Apple Geräten mit T2-Chip nutzt FileVault die Sicherheitsfunktionen des T2-Chips aus [5]. Der FileVault-Schlüssel wird durch eine Schlüsselhierarchie implementiert, die auf den Hardwareverschlüsselungstechnologien des T2-Chips aufbaut [5]. Die gesamte Schlüsselverarbeitung findet in der Secure Encalve statt und niemals auf dem Intel-Prozessor [5].

Nach dem T2-Chip wurde im November 2020 der von Apple selbst entwickelte M1-Chip vorgestellt [3]. Dieser wurde speziell für den Mac entwickelt [3]. Mit dem M1-Chip vereint Apple den T2-Sicherheitschip und die CPU [3]. Er besitzt eine verbesserte Secure Enclave [3] als der T2-Chip. Im Juni 2022 wurde der M2-Chip, eine Verbesserung des M1-Chips, vorgestellt [4]. Dieser besitzt eine noch bessere Secure Enclave als der M1-Chip [4].

2.1.2 Entwicklung der Betriebssysteme für Apple-Computer

Nicht nur die Hardware von Apple hat sich in den letzten Jahren stark verändert, sondern ebenfalls die Betriebssysteme von Apple. Da es mittlerweile 18 verschiedene Betriebssysteme gibt [21], müssen diese unterschieden werden. Für die korrekte Virtualisierung wird somit ein Verständnis der verschiedenen Betriebssysteme benötigt.

Im Jahre 1997 wird mit der Version 7.6 Mac OS als Marke eingeführt [48]. 1998 wird diese durch Mac OS X ersetzt, welches bis heute die Grundlage aller weiteren Betriebssysteme darstellt. Mittlerweile wird es als macOS bezeichnet [48]. Im Folgenden wird auf die für diese Arbeit relevanten Unterschiede eingegangen.
Mit der Einführung des Betriebssystems High Sierra, wird das Dateisystem APFS eingeführt, welches nun neben dem Hierarchical File System Plus (HFS+) ausgewählt werden kann [35].
APFS nutzt an Stelle von Partitionen Volumes, die sich in einem APFS Container befinden [47], wie in Abbildung 2.2 zu erkennen ist.

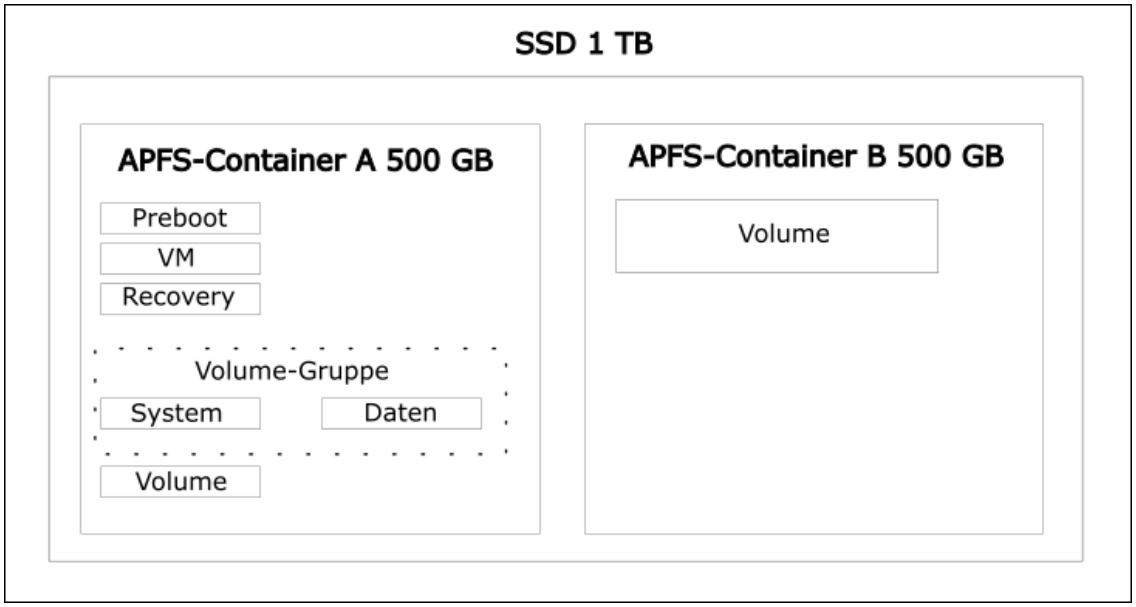


Abbildung 2.2: Beispiel APFS Aufbau bestehend aus 2 APFS-Containern, Volumes und einer Volumegruppe. Die Volumegruppe und die Volumes Preboot, VM und Recovery existieren erst ab macOS Catalina. (Quelle: Eigene Darstellung nach [47])

Dabei stellt das Dateisystem jedem Volume den Speicherplatz nach Bedarf zur Verfügung. Das heißt, dass wenn ein Container mehrere Volumes besitzt, diese sich den Speicherplatz teilen und je nach Bedarf unterschiedlich viel Speicherplatz des APFS-Containers belegen [47]. Somit entspricht der freie Speicherplatz der einzelnen Volumes immer der Größe des APFS-Containers minus die belegte Größe aller Volumes zusammen [47]. Dazu wurden mit APFS Snapshots und eine bessere Verschlüsselung eingeführt [51].

Mit der Einführung von Catalina, kann das Betriebssystem nur noch auf einem APFS-Volume installiert werden. Während unter macOS Mojave das Betriebssystem nur unter einem Volume des APFS Container installiert wird, werden ab macOS Catalina zwei Volumes verwendet [26]. Es wird das ehemalige System-Volume und ein neues Daten-Volume bei der Installation erstellt [26]. Dies sorgt dafür, dass das System-Volume schreibgeschützt ist und verhindert somit das Überschreiben kritischer Systemdateien [26]. Die Daten und Dateien des Nutzers werden auf dem Daten-Volume gespeichert [26]. Die beiden bilden eine Volumegruppe und werden im Finder nur als ein Volume angezeigt [64]. Neben den beiden im Festplattendienstprogramm sichtbaren Volumes wurden mit Catalina noch drei weitere versteckte Volumes zum APFS-Container hinzugefügt: das Preboot-Volume, das VM-Volume und das Wiederherstellungs-Volume [47]. Bei allen drei Volumes handelt es sich um unverschlüsselte Volumes [47]. Dabei beinhaltet das Preboot-Volume die Daten, die zum Booten des System-Volumes benötigt werden [47]. Das VM-Volume wird vom System genutzt, um verschlüsselte SwapDateien zu speichern [47]. Das Wiederherstellungs-Volume ist auch ohne Entsperrung des System-Volumes verfügbar und ist für das Starten des Recovery-OS zuständig [47].
Mit Einführung des Betriebssystem Big Sur wird das System-Volume in einem Snapshot erfasst [47]. Somit wird das Betriebssystem aus dem Snapshot des System-Volumes gebootet und nicht aus dem read-only System-Volume [47].
Ab dem Betriebssystem Monterey wird das System-Volume und das Daten-Volume in eine Volumegruppe im Festplattendienstprogramm angezeigt [22].
Mit dem Betriebssystem Ventura, was im Juni 2022 erschien, sorgt Apple dafür, dass nicht mehr alle Geräte unterstützt werden [36]. Um macOS Ventura installieren zu können, muss der Befehlssatz AVX2 unterstützt werden, was vor allem bei älteren Intel-Chips nicht der Fall ist. [34].

2.2 Datenträgersicherung und Aufbereitung von Apple Systemen

Bevor die Daten von sichergestellten Rechnern aufbereitet werden können, müssen zunächst Datensicherungen gefertigt werden. Für die Virtualisierung von Apple-Systemen ist die Art der Datensicherung wichtig, da sie einen Einfluss auf die Virtualisierung nimmt. Für die Sicherung wird das Anmeldepasswort benötigt.

2.2.1 Datenträgersicherung von Apple-Rechnern

Für die Sicherung gibt es verschiedene Möglichkeiten. Diese hängt von der Konfiguration des Apple-Systems ab. Allgemein unterscheidet man zwischen physikalischer und logischer Datensicherung [32]. Bei einer physikalischen Datensicherung wird eine vollständige Kopie aller adressierbaren Sektoren des Datenträgers erstellt [32]. In speziellen Fällen wird aber eine logische Datensicherung durchgeführt, wobei das Ziel das Erstellen einer selektiven Sicherung ausgewählter, nicht gelöschter Dateien ist [32]. Bei jeder Sicherung muss die Datenintegrität durch Schreibschutz gewährleistet werden. Dafür kann ein hardwarebasierter oder softwarebasierter Schreibschutz verwendet werden [32].

Zur Sicherung kann der Target-Disk-Mode, der Recovery-Mode oder ein externes Boot-Medium verwendet werden. In seltenen Fällen kann auch die Festplatte ausgebaut werden.
Mithilfe des Target-Disk-Modes können Rechner ohne M1 und M2-Chip durch einen anderen Rechner gesichert werden [50]. Befindet sich ein Rechner im Target-Disk-Mode kann er mithilfe eines Kabels an einen anderen Apple-Rechner angeschlossen werden und wird dort als externe Festplatte erkannt [12]. Wichtig hierbei ist den Schreibschutz zu gewährleisten, zum Beispiel durch die Software Disk-Arbitrator [18]. Auf dem anderen Rechner kann der gesicherte Rechner mithilfe eines Programmes wie ewfacquire [39] gesichert werden. Dabei wird das Image als Expert Witness Format (EWF) erstellt [39]. Bei Rechnern mit M1 und M2-Chip funktioniert diese Art der Sicherung nicht, da der Target-Disk-Mode durch den Mac-Sharing-Mode ersetzt wurde [37]. Dabei kann der Rechner zwar an einen anderen Rechner angeschlossen werden, verhält sich aber wie ein SMB-Server und es kann lediglich auf die Daten der Benutzerebene zugegriffen werden [37]. Somit ist eine physikalische Sicherung im Mac-Sharing-Mode nicht möglich.
Im Recovery-Mode besteht die Möglichkeit die Sicherung mittels des Terminals oder des Festplattendienstprogramms durchzuführen. Bei aktivem FileVault wird das Benutzerpasswort eines Admins benötigt, um in den Recovery Mode zu kommen. Im Terminal kann die Sicherung mittels des *dd*-Befehls durchgeführt werden. Das Image liegt im RAW-Format vor. Bei Rechnern mit Sicherheitschips sind die gesicherten Daten verschlüsselt. Im Festplattendienstprogramm besteht die Möglichkeit mittels der Menü-Auswahl Image von Ordner ein Image von einem Volume oder einem bestimmten Ordner durchzuführen [15]. Die Sicherung kann entweder als Disk Image (DMG) oder CDR Datei gespeichert werden [15] und funktioniert vom Daten-Volume nur bis Big Sur. Das Erstellen von einem Image mithilfe des Festplattendienstprogramms funktioniert zwar noch bei Rechnern mit T2-Chip, bei Rechnern mit M1 oder M2-Chips jedoch nicht mehr.
Um mithilfe eines externen Bootmediums eine Sicherung durchführen zu können, muss der Rechner ebenfalls im Recovery-Mode gestartet werden [46]. Dort müssen mithilfe des bekannten Benutzerpasswortes, bei bestimmten Betriebssystemen, die

Startsicherheitsrichtlinien auf reduzierte Sicherheit gesetzt werden [46]. Nun kann als Startoption ein externes Bootmedium verwendet werden. Als externes Bootmedium kann zum Beispiel der DigitalCollector [53] von Cellebrite [54] benutzt werden. Dieser bietet einem eine Benutzeroberfläche, die es ermöglicht eine Sicherung von einem APFS-Container durchzuführen [45] [53]. Diese Sicherung wird als AFF4-Image [2] abgespeichert.

Neben der Methode der Sicherung entscheiden die Sicherheitsfeatures auch, in welcher Art die Sicherung vorliegt. Während bei einem Intel-Rechner ohne T2-Chip die Datensicherung durch alle Sicherungsmethoden durchgeführt und dabei immer der gesamte APFS Container gesichert werden kann, funktionieren bei Rechnern mit T2, M1 und M2-Chip nicht mehr alle Methoden. Beim T2-Chip kann noch mithilfe des Festplattendienstprogramms eine Sicherung vom Daten-Volume durchgeführt werden, beim M1 und M2-Chip geht dies nicht mehr. Sowohl bei Rechnern mit T2, M1 oder M2-Chip kann bei bekanntem Passwort eine Sicherung mithilfe eines externen Bootmediums durchgeführt werden, wie zum Beispiel mit dem DigitalCollector von Cellebrite. Durch ihn wird eine Sicherung vom entschlüsselten APFS-Container durchgeführt. Somit liegt je nach Art der Sicherung entweder ein Image vom APFS-Container, vom Daten-Volume oder vom Benutzerordner vor.

2.2.2 Datenaufbereitung von gesicherten Apple-Rechnern

Es gibt viele verschiedene Programme mit unterschiedlichen Schwerpunkten, die die Daten von gesicherten Rechnern aufbereiten können. Beispiele dafür sind X-Ways Forensics [65] und Magnet Forensics Axiom [62]. Ziel dieser Programme ist es, die Daten, die sich auf dem Image befinden, zu extrahieren und bereitzustellen. Je nachdem, was das Ziel der Datenaufbereitung ist, eignet sich eine Software besser als eine andere.

Die Programme lesen die Images ein und stellen mithilfe von File Carving und Parsing die Daten wieder her [38] [66]. Beim Daten Wiederherstellen werden auch gelöschte Dateien wiederhergestellt [38] [66]. Sie bieten die Möglichkeit vorhandene Kommunikation von E-Mails oder Messengern wiederherzustellen und Inhalte daraus zu extrahieren [38] [66]. Auch Informationen über den Browserverlauf können erlangt werden [38] [66]. Ebenfalls können Informationen über das System inklusive angeschlossener Geräte ermittelt werden [38] [66]. Die gefundenen Dateien werden im rekonstruierten Dateibaum oder nach Kategorien sortiert dargestellt [38] [66]. Dort können die gefundenen Daten angeschaut und wichtige Daten markiert werden [38] [66]. Mithilfe von Filtern können die Dateien nach bestimmten Informationen durchsucht werden [38] [66]. Da von jeder Datei ein Hash-Wert gebildet wird, kann auch mithilfe von Hash-Sets nach bestimmten Dateien gesucht werden [38] [66]. Ebenfalls können Suchlisten verwendet werden, um bestimmte Themen zu finden [38] [66].

Neben der Möglichkeit der Datenextraktion bieten einzelne Programme die Möglichkeit bestimmte bootfähige Systeme zu virtualisieren. Passmark bietet mit ihrer kostenpflichtigen Software OSForensics [43] die Möglichkeit gesicherte macOS-Images in einer virtuellen Maschine zu booten [43]. Jedoch muss dafür das Image die gesamte Festplatte und nicht nur eine Partition oder ein Volume beinhalten [43]. Dabei werden die Image-Formate E01, RAW, Split Images, Virtual Machine Disk (VMDK) und Virtual Hard Disk (VHD) unterstützt [43]. Als Betriebssysteme werden macOS High Sierra, Windows XP bis Windows 10 und manche Linux Distributionen unterstützt [43].
Eine weitere kostenpflichtige Software zur Virtualisierung von gesicherten Systemen ist Forensic Explorer™ von GetData Forensics [17]. Diese ermöglicht es alle Windows Versionen, Linux EXT, und macOS Versionen ohne APFS-Dateisystem zu booten [17]. Jedoch wird auch hier ein bootfähiges System benötigt und Images von APFS-Containern oder logischen Volumes werden nicht unterstützt [17].

Es konnten keine Informationen über kostenlose Programme oder Methoden gefunden werden, die es ermöglichen bootfähige oder nicht bootfähige Sicherungen von Apple Systemen zu virtualisieren.

2.3 Recovery-Mode

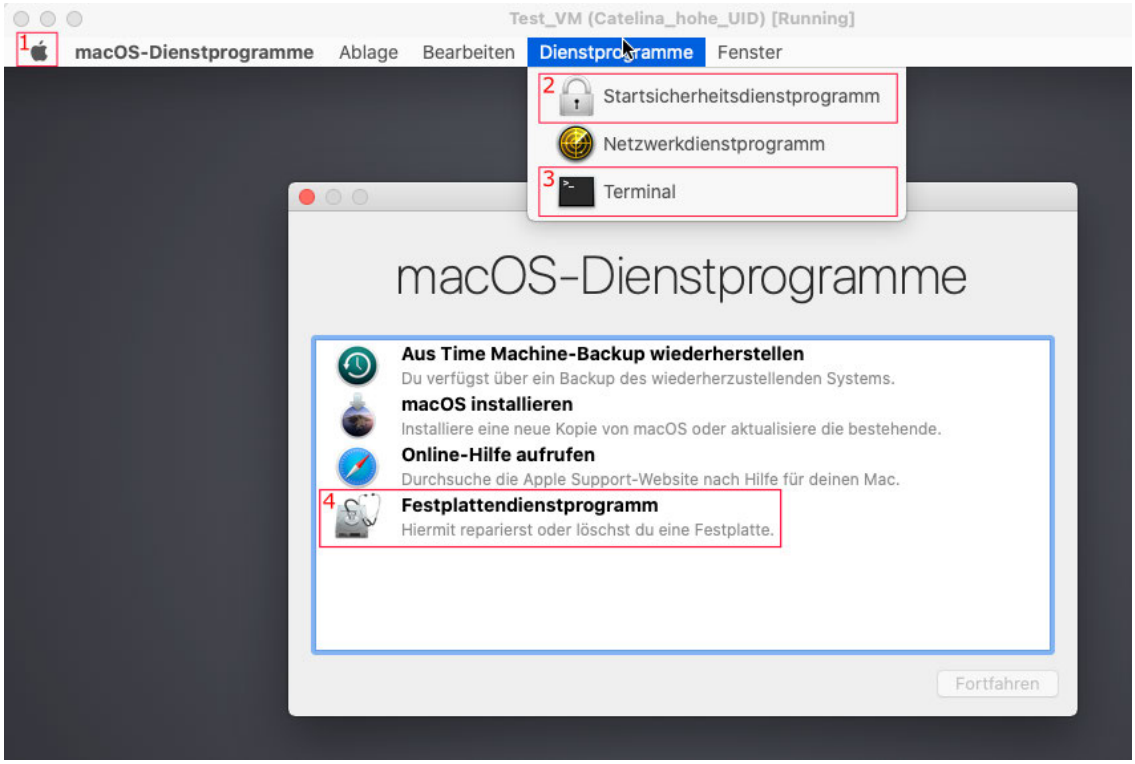


Abbildung 2.3: Darstellung des Recovery-Modes mit seinen unterschiedlichen Dienstprogrammen und Auswahlmöglichkeiten. (Quelle: Eigene Darstellung)

Für die Virtualisierung der gesicherten Apple-Systeme wird der Recovery-Mode benutzt. Dieser ist eine Apple eigene Startmethode, die es ermöglicht Veränderungen an den Festplatten und dem Rechner vorzunehmen [57]. In Abbildung 2.3 wird der Aufbau des Recovery-Modes dargestellt. Er bietet verschiedene Dienstprogramme und die Möglichkeit, das Startvolume über das Apple-Logo (Abb. 2.3, 1) zu wählen. Neben dem Festplattendienstprogramm ist auch das Terminal (Abb. 2.3, 3) wichtig für die Virtualisierung von gesicherten Apple-Systemen. Mithilfe des Punktes Startsicherheitsdienstprogramm (Abb. 2.3, 2) können die Richtlinien für das Booten von externen Festplatten geändert werden [27].

Das Festplattendienstprogramm (Abb. 2.3, 4) bietet verschiedene Möglichkeiten, um eine Festplatte, ein APFS-Container oder ein Volume zu bearbeiten [16]. In Abbildung 2.4 wird der Aufbau des Festplattendienstprogramms dargestellt.

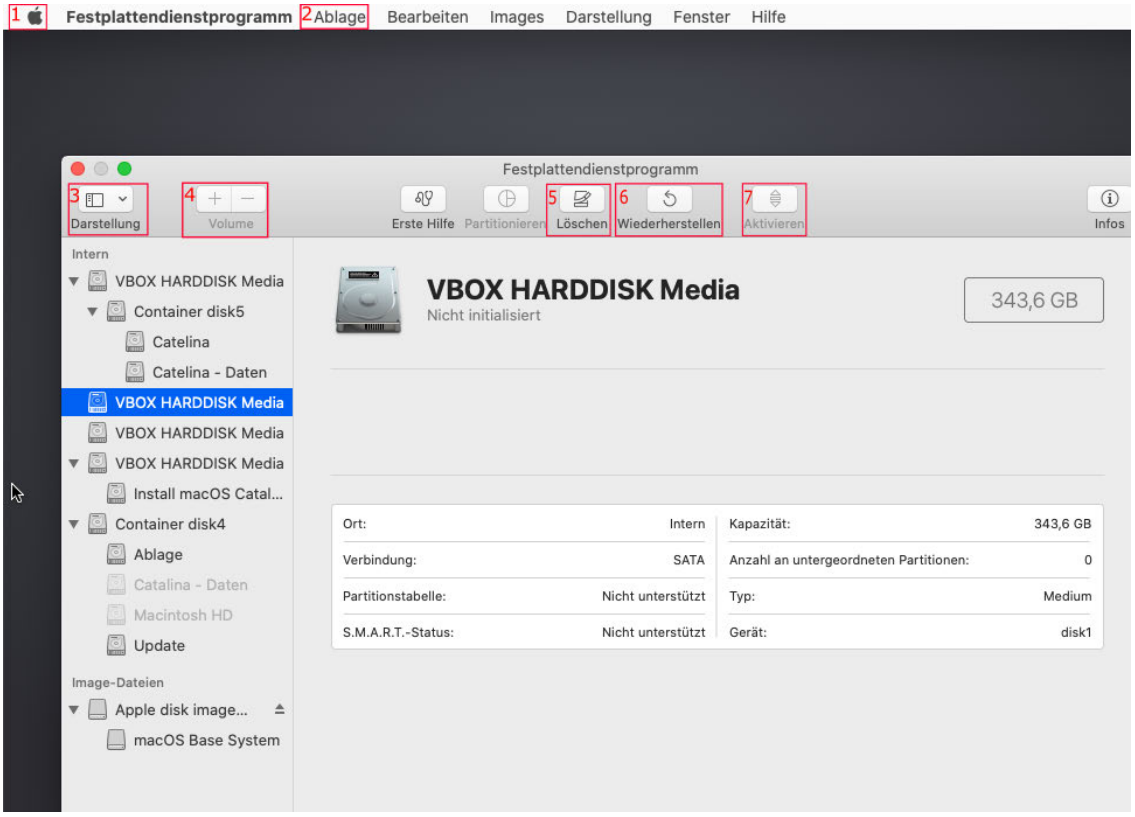


Abbildung 2.4: Darstellung des Festplattendienstprogramms im Recovery-Mode. 1: Startvolume auswählen, 2: Image Erstellen, 3: Ändern der Anzeige der Festplatten, 4: Volume Löschen / Hinzufügen 5: Formatieren, 6: Volumes etc. Wiederherstellen, 7: Volumes Aktivieren. (Quelle: Eigene Darstellung)

Über das Apple-Logo (Abb. 2.4, 1) kann das Startvolume ausgewählt oder die Maschine ausgeschaltet werden. Mithilfe des Punktes Ablage (Abb. 2.4, 2) kann ein leeres Image oder ein Image von einem Ordner erstellt werden. Durch den Punkt Darstellung (Abb. 2.4, 3) kann zwischen der Darstellung „alle Geräte anzeigen“ oder nur „Volumes anzeigen“ gewechselt werden. Der Punkt Löschen (Abb. 2.4, 4) bietet die Möglichkeit

ein Volume, ein APFS-Container oder eine Festplatte neu zu formatieren. Mithilfe von Wiederherstellen (Abb. 2.4, 5) können Festplatten, APFS-Container oder Volumes von anderen Festplatten, APFS-Container oder Volumes wiederhergestellt werden. Mithilfe von +/- können Volumes entfernt oder hinzugefügt werden. Über den Punkt Aktivieren (Abb. 2.4, 6) können verschlüsselte Volumes entschlüsselt und gemountet werden.

Diese unterschiedlichen Möglichkeiten der Bearbeitung der Festplatten, werden benötigt, um die Virtualisierungen durchführen zu können.

2.4 Virtualisierung

Im Jahr 1960 wurde eines der ersten Virtualisierungsprogramme entwickelt, mit dem jeder Benutzer ein isoliertes System verwenden konnte und dennoch waren alle Systeme in einer gemeinsamen, auf gleicher Zeit basierten Computerumgebung [14]. Seit diesem Zeitpunkt nahm die Entwicklung von Virtualisierungssoftware deutlich zu. Heutzutage abstrahieren und transformieren Virtualisierungsprogramme verschiedene physische Ressourcen von einem Computer, wie CPU, Speicher, Festplatten und Netzwerkadapter, in ein oder mehrere verschieden konfigurierte Computerumgebungen [9]. Dadurch ermöglichen sie Betriebssysteme in einer unabhängigen Umgebung zu verwenden, ohne andere Umgebungen zu beeinflussen [9]. Deshalb ist es möglich, in einer Umgebung zu experimentieren, ohne dabei das eigentliche System zu zerstören.

Im Jahr 2010 veröffentlichten Zhang et. al. [68], dass es möglich ist mithilfe von Live View [33], VMware Workstation [60] und VMware Virtual Disk Development Kit [59] gesicherte Betriebssysteme zu virtualisieren [68]. Dabei kann es aber zu einem „Blue Screen of Death“ kommen und somit die Virtualisierung fehlschlagen [68]. Live View ist eine Open-Source-Software, die es ermöglicht Windows 2008, Vista, 2003, XP, 2000, NT, Me und 98 und manche Linux Systeme zu virtualisieren [33].
Mittlerweile gibt es weitere, meistens kostenpflichtige Programme, die es ermöglichen, gesicherte bootfähige Systeme zu virtualisieren [17] [43]. Dies ist vor allem mit gesicherten Windows Systemen möglich. Auch für Linux gibt es einige Programme und Methoden. Bei gesicherten Systemen von Apple scheint dies nur eingeschränkt möglich zu sein, denn es können nur wenige Programme bestimmte macOS Versionen virtualisieren [17] [43].

Zu den bekanntesten Virtualisierungsprogrammen, die auf Apple-Systemen laufen, gehören VirtualBox [61], VMWare Fusion [58] und Paralells Desktop [41]. In dieser Arbeit wurde VirtualBox verwendet, da die anderen beiden Programme kostenpflichtig sind.

2.4.1 Virtualisierungssoftware Oracle VM VirtualBox

VirtualBox ist eine Open Source Virtualisierungssoftware und virtualisiert einen AMD64 oder Intel64 Prozessor [55]. Momentan funktioniert VirtualBox unter Windows, Linux, macOS und Solaris [55]. Es bietet die Möglichkeit mehrere virtuelle Maschinen gleichzeitig laufen zu lassen [55]. Ist eine Virtuelle Maschine (VM) einmal installiert und eingerichtet, kann die VM selbst und ihre Festplatten als Container betrachtet werden, was es ermöglicht, die VMs einzufrieren, zu kopieren und zu sichern [55]. Die einzelnen virtuellen Festplatten können aus der VM entfernt und in eine andere hinzugefügt werden [55]. Dies bietet die Möglichkeit Testumgebungen zu erschaffen, in denen Dinge ausprobiert werden können, ohne das ursprüngliche System zu zerstören [55]. Durch die Möglichkeit Snapshots zu erstellen, kann die VM zu unterschiedlichen Sicherungspunkten des Systems und somit zu einem funktionierenden Zeitpunkt zurückkehren [55].
Um zwischen der Virtualisierung und dem Rechner, auf dem die VM läuft, unterscheiden zu können, spricht man vom Host-System, beim Rechner und Gast-System bei der Virtualisierung [55]. Jede VM besitzt einen Ordner, wo alle dazugehörigen Daten, wie die Datei mit der .vbox Dateiendung, die Festplatten und der Nvram, gespeichert werden [55]. Diese Ordner befinden sich normalerweise in dem Ordner VirtualBox VMs, welcher im Home-Verzeichnis des Host-Systems erstellt wird [55].
Neben der visuellen Möglichkeit mithilfe des Oracle VM VirtualBox Managers bietet VirtualBox mithilfe von *VBoxManage* eine kommandozeilenbasierte Möglichkeit an, die VMs zu konfigurieren [55]. Dabei beinhaltet *VBoxManage* alles, was die grafische Benutzeroberfläche kann, aber noch vieles mehr [55]. So hat man zum Beispiel die Möglichkeit spezifische Einstellungen der CPU der VM zu verändern. Dies wird benötigt, um ein macOS Gast-System zum Starten zu bekommen.

2.4.2 Oracle VM VirtualBox Manager

Der Oracle VM VirtualBox Manager ist ein sehr wichtiger Bestandteil von VirtualBox. In Abbildung 2.5 ist der Aufbau des VM VirtualBox Managers dargestellt. Er listet die Eigenschaften der VMs auf und bietet die Möglichkeit die einzelnen VMs zu verwalten [55]. Über den Punkt Neu (Abb. 2.5, 3) kann eine neue VM erstellt werden, wie in Abschnitt 3.1 erklärt, und über den Punkt Hinzufügen (Abb. 2.5, 4) kann eine bestehende VM hinzugefügt werden [55]. Ein wichtiger Punkt ist der Punkt Ändern (Abb. 2.5, 5). Über diesen kann die Konfiguration der VM verwaltet werden, sowie virtuelle Festplatten hinzugefügt oder entfernt werden [55]. Der Punkt Werkzeuge (Abb. 2.5, 1) bietet einem die Möglichkeit, die Erweiterungspakete, die Medien, das Netzwerk, die Cloud und die Aktivitäten zu verwalten [55]. Dies geht über das Drei-Punkte-Menü, wenn Werkzeuge ausgewählt wird. Über das Drei-Punkte-Menü bei der VM (Abb. 2.5, 2) können die Sicherungspunkte, die Logs, die Aktivität und der Dateimanager verwaltet werden [55].

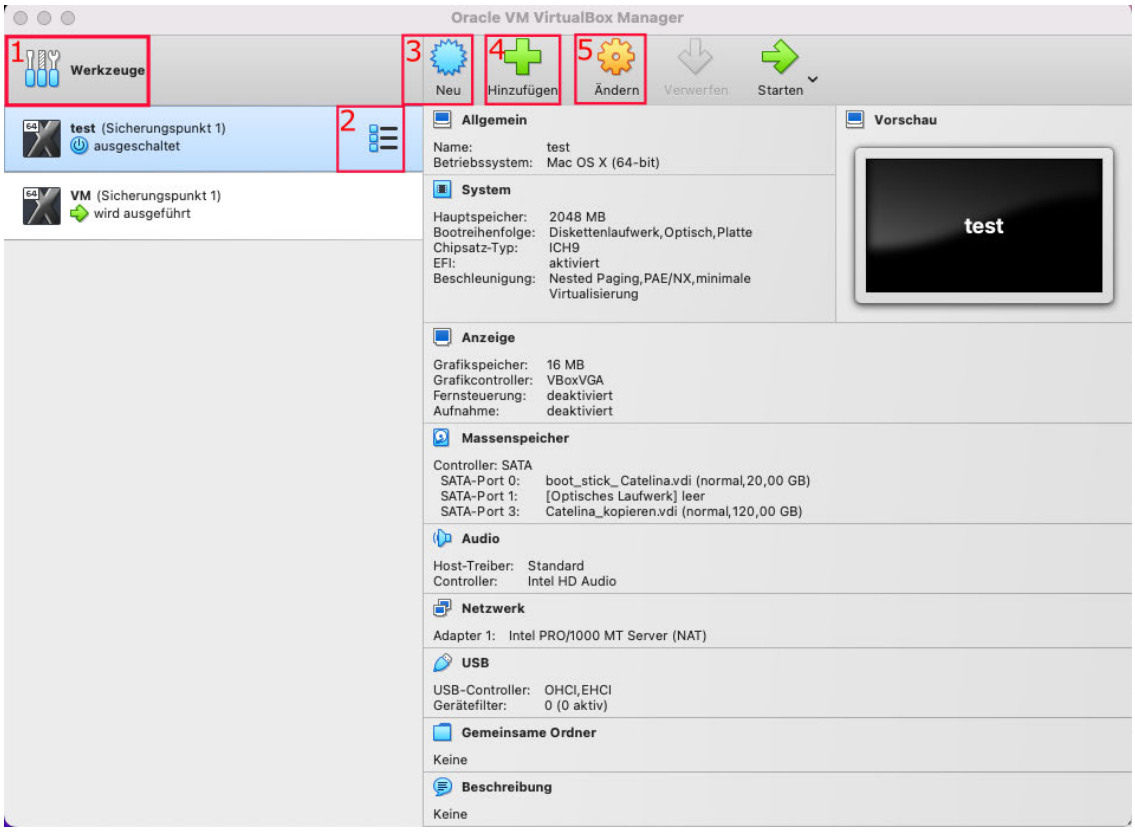


Abbildung 2.5: Aufbau des Oracle VM VirtualBox Manager auf dem Host-System. 1: Werkzeuge, 2: weitere Einstellungen VM, 3: neue VM erstellen, 4: vorhandene VM hinzufügen, 5: Einstellungen VM und Festplatten hinzufügen. (Quelle: Eigene Darstellung)

2.4.3 Sicherungspunkte VM VirtualBox

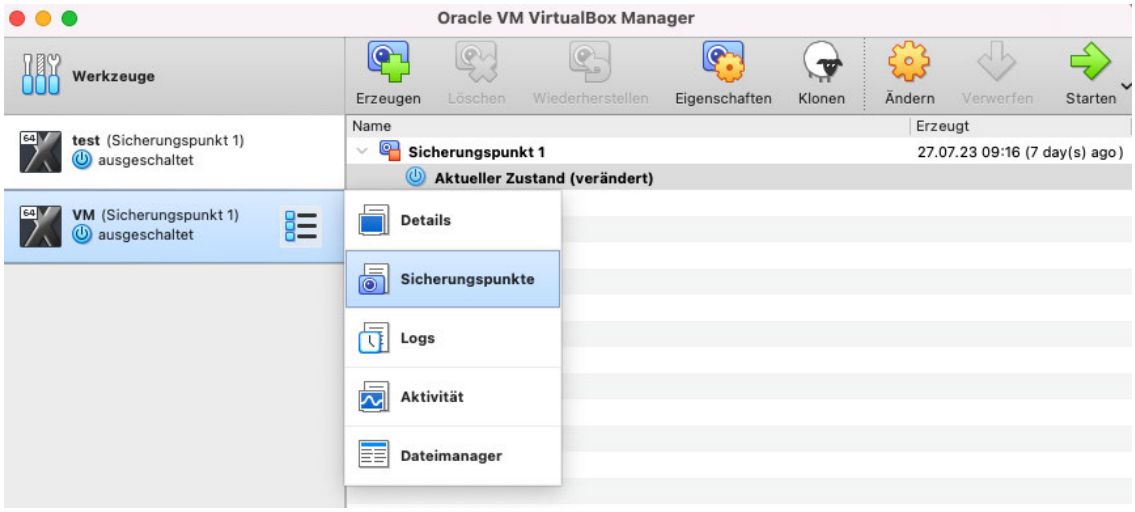


Abbildung 2.6: Darstellung der Einstellung Sicherungspunkte in Oracle VM VirtualBox Managers. (Quelle: Eigene Darstellung)

VirtualBox bietet die Möglichkeit Sicherungspunkte anzulegen. Diese speichern das System zu einem bestimmten Zeitpunkt der VM, zu welchem zu jederzeit zurückgekehrt werden kann [55]. Diese können mithilfe des VirtualBox Managers erstellt werden. Dafür muss auf die drei Punkte am Rand der VM geklickt und dort Sicherungspunkte ausgewählt werden (Abb. 2.6) [55]. Abbildung 2.6 stellt den Aufbau des Punktes Sicherungspunkte da. Dort können neue Sicherungspunkte erzeugt und alte wiederhergestellt oder gelöscht werden.

3 Methodik und Durchführung

In diesem Kapitel wird auf die Entwicklung der einzelnen Methoden für unterschiedliche Sicherungsarten eingegangen. Dafür wurde zunächst eine VM für ein macOS Gastsystem erstellt und anschließend die Datensicherung in eine für die VM passende Datei umgewandelt. Anschließend konnten unterschiedliche Ansätze für verschiedene Sicherungsarten entwickelt werden.

3.1 Vorbereitung: VirtualBox macOS VM einrichten

Im Folgenden wird erklärt, wie eine VM mit einem macOS Betriebssystem als Gastsystem erstellt und modifiziert wurde. Das Betriebssystem Ventura benötigt zur Installation AVX2. Der Host-Rechner (siehe Abschnitt 3.4) besitzt dieses nicht. Deshalb wurden nur Betriebssysteme bis Ventura virtualisiert.

Zur Erstellung der VM wurde der Oracle VM VirtualBox Manager verwendet. Bevor die VM erstellt wurde, wurde zunächst das VirtualBox Extension Pack installiert, um Komplikationen im weiteren Verlauf zu verhindern. Nach dem Installieren des Extension-Packs wurde eine neue VM mithilfe des VirtualBox Managers erstellt. Die spezifischen Einstellungen sind im Anhang A dargestellt.
Apple hat für die Installation von macOS Systemen Sicherheitsfeatures eingebaut. Aus diesem Grund mussten ein paar Konfigurationseinstellungen mithilfe des Terminals und des *VBoxManage* Befehls (Anhang A) geändert werden.
Für das Einrichten der VM wurde ein Bootstick benötigt. Um die Bootsticks mit dem macOS Betriebssystem zu erstellen, musste zunächst das gewünschte Betriebssystem aus dem App-Store heruntergeladen werden und anschließend mithilfe des Terminals aus einem USB-Stick ein Bootstick erstellt werden. Damit die erstellten Bootsticks in der VM verwendet werden konnten, wurde mithilfe von *VBoxManage* (Anhang A) zunächst eine VMDK-Datei und anschließend aufgrund von eventuell auftretenden Fehlern eine Virtual Disk Image (VDI)-Datei erstellt.
Von dieser wurden die Lese-Schreib-Rechte mithilfe vom *chmod* Befehl (Anhang A) verändert, damit sie zur VM hinzugefügt werden konnte.

Nach dem Erstellen des Bootsticks wurde mittels des VirtualBox Managers über den Punkt Ändern der Bootstick als Festplatte hinzugefügt. Anschließend wurde eine neue Festplatte erstellt und zur VM hinzugefügt, um auf dieser anschließend das Betriebssystem zu installieren. Nach dem Starten der VM musste mithilfe des Festplattendienstprogramms zunächst die Festplatte initialisiert werden, um anschließend das Betriebssystem zu erstellen.

3.2 Vorbereitung: Konvertierung der gesicherten Daten für die Arbeit mit VirtualBox

Die unterschiedlichen Datensicherungen, die in verschiedenen Image-Formaten vorliegen, konnten so noch nicht von der VM verwendet werden. Diese mussten zunächst in ein kompatibles Format umgewandelt werden. Für die verschiedenen Image-Formate ist die Umwandlung unterschiedlich. Ziel war es am Ende eine eigenständige VDI-Datei des gesicherten Rechners zu besitzen, um mit dieser in VirtualBox weiterzuarbeiten.

3.2.1 Konvertierung E01- und RAW-Images mithilfe von Xmount in VDI-Datei

Xmount ist ein kommandozeilenbasiertes Programm, das die Möglichkeit bietet ein Image im EWF, RAW oder Advanced Forensic Format (AFF) in RAW (DD), DMG, VHD, VDI oder Virtual Machine Disk (VMDK) Image umzuwandeln [67].
Mithilfe von Xmount wurden die einzelnen Datensicherungen, die als RAW-Image oder EWF-Image vorlagen, als VDI-Datei gemountet. Für ein gesplittetes RAW-Image wurde folgender Befehl verwendet

```
1 sudo xmount --in raw <Pfad zum Image>/<Imagename>.?? --out vdi
  ↪ <Pfad zum Mountpoint>/Mountpoint/
```

und für ein gesplittetes EWF-Image folgender Befehl.

```
1 sudo xmount --in ewf <Pfad zum Image>/<Imagename>.e?? --out vdi
  ↪ <Pfad zum Mountpoint>/Mountpoint/
```

Der jeweilige Befehl sorgt dafür, dass die erstellten Sicherungen, die im RAW-Format oder EWF-Format vorliegen, als VDI-Datei an einer bestimmten Stelle gemountet werden [67]. Die Wildcard-Zeichen „??“ werden angegeben, damit alle Fragmente des Images mit berücksichtigt werden [67].
Da das RAW-Image und das EWF-Image nicht verändert werden können, musste entweder eine Kopie der VDI-Datei erstellt oder der Parameter –cache <caching> verwendet werden. In diesem Fall wurde mithilfe des folgenden *Clone*-Befehls von VirtualBox eine eigene VDI-Datei erstellt, um die eigentliche Sicherung nicht zu verändern.

```
cd /Applications/VirtualBox.app/Contents
sudo VBoxManage clonehd --format vdi <Pfad zum
↪ Mountpoint>/Mountpoint/<Imagename>.vdi <Pfad Abspeicherung
↪ vdi>/<Imagename>.vdi
```

Der Befehl sorgt dafür, dass von der gemounteten VDI-Datei eine neue identische VDI-Datei an dem angegebenen Pfad abgespeichert wird [55]. Aufgrund von Apple-Sicherheitsfeatures mussten die Lese-Schreib-Rechte der VDI-Datei mithilfe des *chmod* Befehls geändert werden, damit die Datei als Festplatte zu einer VM hinzugefügt werden kann. Dieses gilt auch für die in den folgenden Abschnitten erstellten VDI-Dateien.

3.2.2 Konvertierung von DMG und CDR Dateien in VDI-Datei

Bei der Sicherungsart „Image von Ordner“ entsteht eine DMG-Datei oder eine Compact Disc Recordable (CDR) Datei. Die CDR-Datei kann wie ein RAW-Image mit Xmount gemountet werden, die DMG-Datei jedoch nicht. Beide können aber mithilfe von *hdiutil* [20] gemountet werden. In dieser Arbeit wurde das Image mithilfe von dem folgenden *hdiutil*-Befehl als Festplatte gemountet.

```
    hdiutil attach <Pfad zum image>/<Imagename>.cdr -nomount
```

```
    hdiutil attach <Pfad zum image>/<Imagename>.dmg -nomount
```

Dabei wurden mithilfe des Parameters *-nomount* die einzelnen Volumes nicht gemountet, da es ansonsten zu Problemen bei der Erstellung der VMDK-Datei gekommen ist. Mithilfe von *VBoxManage* wurde wie folgt zunächst eine VMDK und anschließend eine VDI-Datei erstellt.

```
    cd /Aplications/VirtualBox.App/Contents
    sudo VBoxManage createmedium disk --filename=<Pfad
    ↪ vmdk>/<Imagename>.vmdk --variant=RawDisk --format=VMDK
    ↪ --property RawDrive=/dev/<Diskidentifier>
    sudo VBoxManage clonehd --format vdi <Pfad
    ↪ vmdk>/<Imagename>.vmdk <Pfad Abspeicherung
    ↪ vdi>/<Imagename>.vdi
```

Der Befehl *createmedium* sorgt dafür, dass aus der angeschlossenen Festplatte am Pfad /dev/<Diskidentifier> eine VMDK-Datei erstellt wird [55]. Die VDI-Datei wurde erstellt, damit eine unabhängige Datei von der eigentlichen Sicherung in der VM

verwendet werden konnte. Dafür erstellt der *clonehd*-Befehl an dem angegebenen Pfad aus der VMDK-Datei eine VDI-Datei [55]. Beim Erstellen der VMDK-Datei aus einer DMG-Datei trat die Fehlermeldung „VERR_ACCESS_DENIED“ auf, wodurch das Image zunächst in eine CDR-Datei umgewandelt werden musste.

3.2.3 Konvertierung von AFF4-Image in VDI-Datei

Zwar bietet Xmount die Möglichkeit AFF Images zu benutzten, jedoch funktioniert dies nicht für die Weiterentwicklung AFF4. Um eine AFF4 Datei nutzen zu können, musste diese zunächst in ein anderes Image-Format konvertiert werden. Mithilfe von X-Ways Forensics [65] wurde das Image in ein EWF Image und anschließend, wie in Unterabschnitt 3.2.1 dargestellt, in eine VDI-Datei umgewandelt.

3.3 VirtualBox VM im Recovery Mode starten

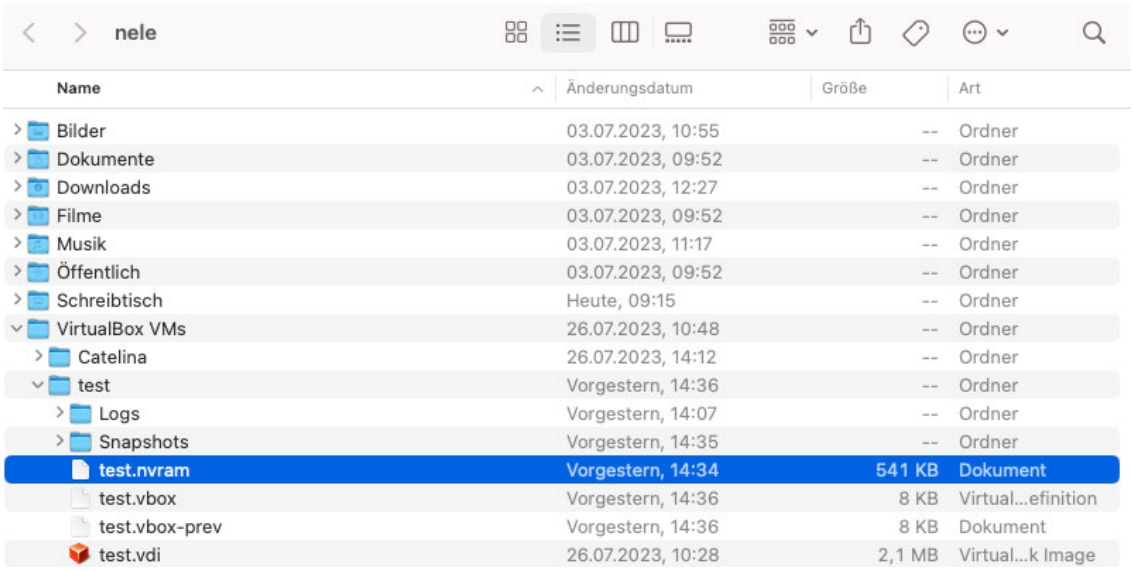


Abbildung 3.1: Darstellung des Speicherortes des Nvram der VM. Er befindet sich im Userordner unter VirtualBox VMs unter dem Namen der VM. (Quelle: Eigene Darstellung)

Der Recovery-Mode wird normalerweise beim Anschalten durch gedrückt Halten der cmd-Taste und R-Taste oder durch das Drücken und Halten der Anschalttaste gestartet [57]. Dies funktionierte bei der VM nicht. Um die VM im Recovery-Mode zu starten, musste zunächst ein passender Bootstick als VDI-Datei eingelegt und der Nvram gelöscht werden. Dieser befindet sich als Datei im Userverzeichnis des Ordner „VirtualBox VM“, in dem sich wiederum der Ordner mit dem Namen der VM, die gestartet werden soll, befindet, wie in Abbildung 3.1 zu erkennen ist. Nach dem erfolgreichen Löschen des Nvrams konnte die VM normal gestartet werden und befand sich

anschließend im Recovery Mode. Um aus dem Recovery Mode herauszukommen, musste die VM einmal aus dem Recovery-Mode gestartet werden. Beim nächsten Starten der VM, startete sie automatisch das Betriebssystem.

3.4 Entwicklung der Methoden zur Virtualisierung von gesicherten Apple-Systemen

In diesem Kapitel werden verschiedene Methoden zur Virtualisierung von fünf unterschiedlichen Sicherungsarten von verschiedenen Apple-Konfigurationen vorgestellt. Dabei wird auf unterschiedliche Lösungsansätze inklusive ihrer Probleme eingegangen. Als Hostrechner diente ein MacPro mit der Modellnummer A1481 und der EMC-Nummer 2630, der mit dem Betriebssystem Monterey arbeitet. Die Virtualisierung erfolgte mithilfe der Software Oracle VM VirtualBox in der Version 7.0.8. Als Image wird im Folgenden die erstellte VDI-Datei bezeichnet und nicht die eigentliche Sicherung.

3.4.1 Verwendete Computersysteme für die Sicherungen

Tabelle 3.1: Virtualisierungsmethoden nach Sicherungsarten und Rechnermodell geordnet.

Chip	Modellnr.	EMC	FileVault	Sicherungsart	Methode
Intel ohne T2	A1534	2991	ja	dd-Befehl: *gesamte Festplatte*	1
				dd-Befehl: *APFS-Container*	2
Intel mit T2	A1989	3214	ja	Festplattendienstprogramm: *Daten-Volume*	4
				Festplattendienstprogramm: *Ordner*	5
				DigitalCollector: *APFS-Container*	3
M1	A2338	3578	nein	DigitalCollector: *APFS-Container*	3

Für die Entwicklung der unterschiedlichen Methoden wurden fünf Sicherungen von drei unterschiedlichen Rechnern verwendet. Es wurde ein MacBook mit Intel-Chip ohne T2-Chip, ein MacBook Pro mit Intel-Chip und T2-Chip und ein MacBook Pro mit M1-Chip verwendet. In der Tabelle 3.1 werden die einzelnen Rechner und ihre unterschiedlichen Sicherungsarten dargestellt. Die Nummer in der Spalte Methode steht für die Methode, für welche die Datensicherung zur Entwicklung verwendet wurde.

Für die Entwicklung der Methoden wurden zwei unterschiedliche Sicherungen eines MacBook Modells A1534 mit der EMC-Nummer 2991 herangezogen. Dieses MacBook verfügt über eine 500 GB SSD-Festplatte und einen APFS-Container, dessen Aufbau in Abbildung 3.2 dargestellt wird. Der APFS-Container umfasst die Volumegruppe Volumes von Catalina (Catalina und Catalina-Daten), das Ablage-Volume und das Macintosh HD-Volume. Das Betriebssystem von Catalina ist Monterey, während Macintosh HD Mojave verwendet. Bei beiden Systemen ist FileVault aktiv. Die Sicherungen wurde im Recovery-Mode mithilfe des *dd*-Befehls durchgeführt. Die eine Sicherung umfasst die gesamte Festplatte. Die andere Sicherung beinhaltet den gesamten APFS-Container.

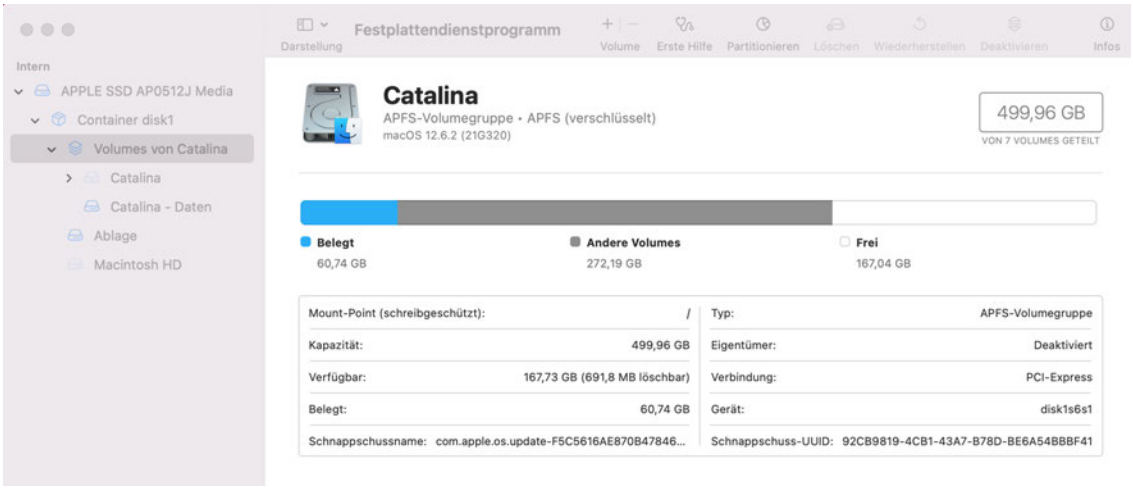


Abbildung 3.2: Festplattendienstprogramm unter dem Betriebssystem macOS Monterey auf dem gesicherten Rechner. Der APFS-Container besitzt die Volumegruppe Volumes von Catalina und die Volumes Ablage und Macintosh HD. (Quelle: Eigene Darstellung)

Darüber hinaus wurden Sicherungen von einem MacBook Pro mit der Modellnummer A1989 und EMC-Nummer 3214 sowie einem T2-Chip erstellt. Dieses MacBook verfügt über eine 256 GB SSD-Festplatte und einen APFS-Container, der in Abbildung 3.3 zu erkennen ist. Der APFS-Container enthält die Volumes Macintosh HD, Macintosh HD - Daten und Mojave. Das Betriebssystem für das Macintosh HD-Volume ist Catalina, während für das Mojave-Volume Mojave verwendet wird. Bei dem Macintosh HD-Volume ist FileVault aktiv. Die Sicherung des Daten-Volumes erfolgte im Recovery-Mode mithilfe des Festplattendienstprogramms. Zusätzlich wurde im

Recovery-Mode mithilfe des Festplattendienstprogramms eine Sicherung der einzelnen Ordner User, Applications, Library und System vom Daten-Volume erstellt. Des Weiteren wurde eine Sicherung des gesamten APFS-Containers mithilfe eines externen Bootsticks und DigitalCollector durchgeführt.

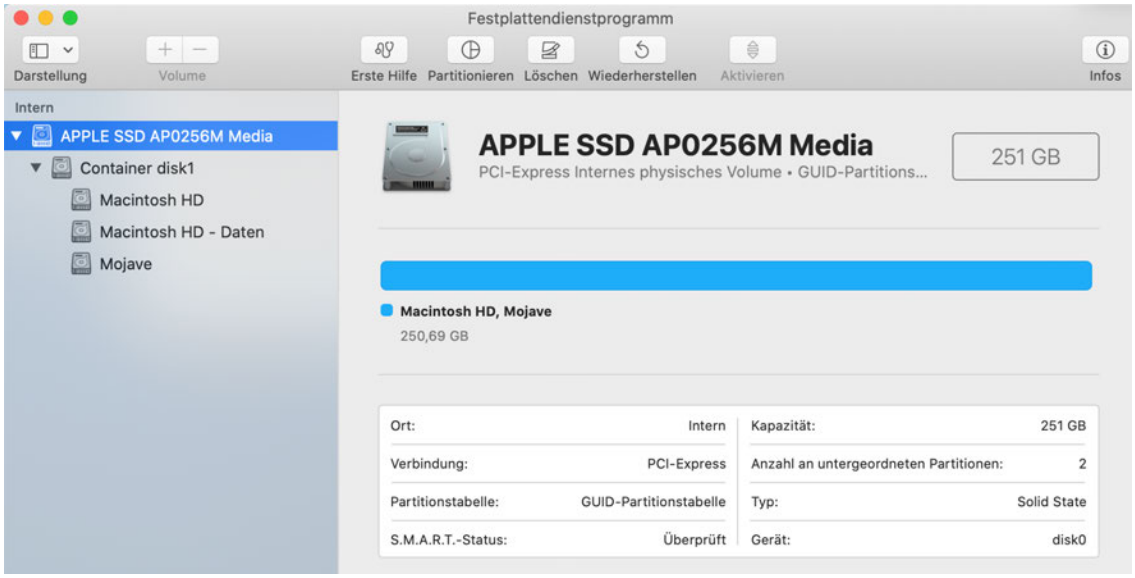


Abbildung 3.3: Das Festplattendienstprogramm auf einem MacBook Pro mit T2-Chip, welche vom Macintosh HD-Volume gestartet wurde. Der APFS-Container beinhaltet die Volumes Macintosh HD, Macintosh HD - Daten und Mojave. (Quelle: Eigene Darstellung)

Als Letztes wurde eine Sicherung von einem MacBook Pro mit der Modellnummer A2338 und EMC-Nummer 3578 sowie einem M1-Chip durchgeführt.

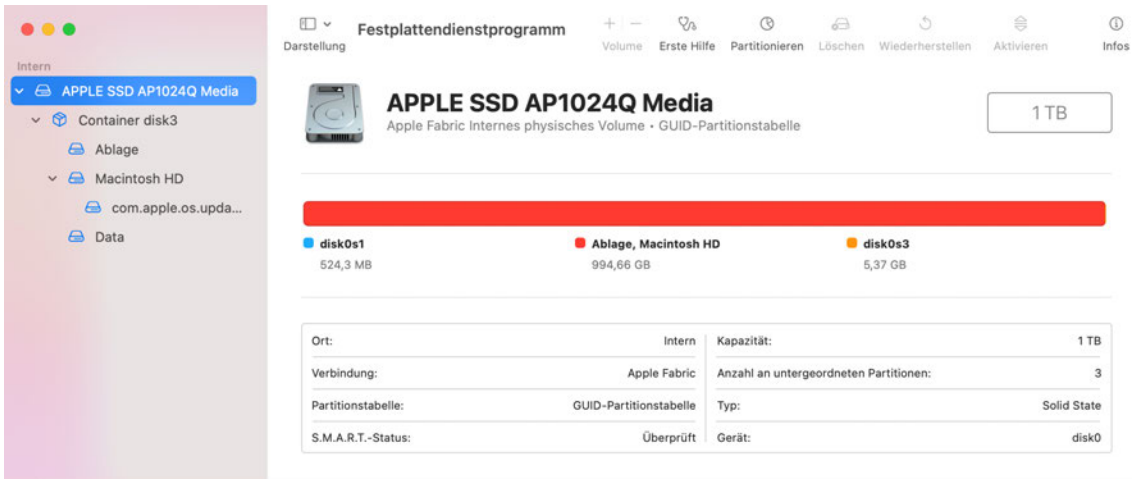


Abbildung 3.4: Festplattendienstprogramm auf einem MacBook Pro mit M1-Chip. Der APFS-Container besitzt die Volumes Ablage, Macintosh HD und Data. (Quelle: Eigene Darstellung)

Dieses MacBook Pro verfügt über eine 1 TB SSD-Festplatte. Der APFS-Container enthält die Volumes Ablage, Macintosh HD und Data, und das Betriebssystem ist Big Sur, wie in Abbildung 3.4 zu erkennen ist. FileVault ist nicht aktiviert. Die Datensicherung wurde ebenfalls mithilfe eines externen Bootsticks und DigitalCollector durchgeführt.

3.4.2 Methode 1: Virtualisierung von der gesicherten Festplatte von einem Rechner mit Intel Chip ohne T2-Chip

Bei einem Rechner mit Intel Chip ohne T2-Chip konnte die gesamte Festplatte mithilfe des *dd*-Befehls gesichert werden. Obwohl die gesamte Festplatte gesichert wurde, entstanden Probleme bei dem Versuch die Sicherung zu virtualisieren.

Für die Virtualisierung der gesicherten Festplatte wurde eine konfigurierte Apple-VM, ohne dass das Betriebssystem schon installiert war, benötigt. Dazu wurde ein virtueller Bootstick mit dem passenden Betriebssystem benötigt. Die VDI-Datei der Sicherung der gesamten Festplatte wurde zusammen mit dem Bootstick zur VM hinzugefügt. Anschließend wurde die VM im Recovery-Mode gestartet. Es trat das Problem auf, dass die gesicherte Festplatte zwar im Festplattendienstprogramm angezeigt wurde, aber der APFS-Container nicht erkannt wurde, wie in Abbildung 3.5 zu erkennen ist.

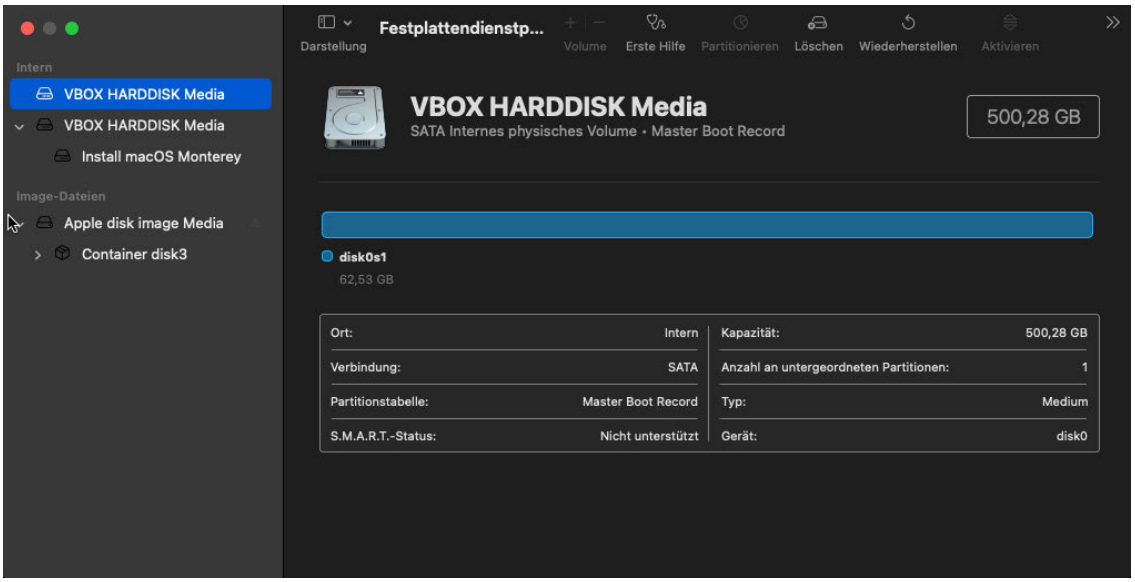


Abbildung 3.5: Das Festplattendienstprogramm im Recovery-Mode bei dem Versuch die gesamte Festplatte eines MacBooks mit Intel-Chip und ohne T2-Chip zu virtualisieren. Der APFS-Container wird in diesem nicht erkannt. (Quelle: Eigene Darstellung)

Bei der Betrachtung der Aufteilung der Festplatte mithilfe des Terminals fiel auf, dass das Partitionsschema der Festplatte nicht als Globally Unique Identifier (GUID) erkannt wurde. In Abbildung 3.6 sind die einzelnen Festplatten mit ihren Partitionen

aufgelistet. Disk 0 ist die Sicherung, bei der als Partitionsschema FDisk_parition_sheme angegeben wurde, obwohl es ein GUID Schema sein müsste. Dazu stand bei 1: lediglich ein Hexadezimalwert und nicht EFI als Type und es fehlte 2: wo eigentlich Apple_APFS stehen müsste.

```
-bash-3.2# diskutil list
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *500.3 GB   disk0
   1:                       0xEE                         62.5 GB    disk0s1
                  (free space)                          437.7 GB   -

/dev/disk1 (internal, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        *21.5 GB    disk1
   1:                        EFI EFI                     209.7 MB   disk1s1
   2:                 Apple_HFS Install macOS Monterey   21.1 GB    disk1s2

/dev/disk2 (disk image):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        +1.1 GB     disk2
   1:                Apple_APFS Container disk3          1.1 GB     disk2s1

/dev/disk3 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +1.1 GB     disk3
                                 Physical Store disk2s1
   1:                APFS Volume macOS Base System       919.7 MB   disk3s1
   2:                APFS Volume Preboot                 80.0 MB    disk3s2
```

Abbildung 3.6: Auflistung aller Festplatten inklusive ihres Partitionsschemas und den Partitionen. Bei der Festplatte 0, der Sicherung, wird das falsche Partitionsschema und keine falschen Partitionen angezeigt. (Quelle: Eigene Darstellung)

Aus dem Grund, dass die Festplatte nicht richtig erkannt wurde, war es nicht möglich, die gesamte Festplatte zu virtualisieren.

3.4.3 Methode 2: Virtualisierung eines gesicherten APFS-Containers von einem Rechner mit Intel Chip ohne T2-Chip

Im Folgenden wird auf die Virtualisierung eines APFS Containers von einem Intel Rechner ohne T2-Chip erklärt und auf die auftretenden Probleme eingegangen.

Für die Virtualisierung wurde eine VM benötigt, die für Apple-Gast-Systeme konfiguriert wurde. Das Betriebssystem war noch nicht installiert. Der VM wurde ein virtueller Bootstick als Festplatte hinzugefügt. Dabei war darauf zu achten, dass die korrekte macOS-Version des virtuellen Bootsticks ausgewählt wurde, damit keine Probleme beim Entschlüsseln der verschlüsselten Volumes mit FileVault auftraten. Besitzt der APFS-Container mehrere System-Volumes musste der Bootstick die höhere Version haben.

Neben dem Bootstick wurde das Image als VDI-Datei hinzugefügt. Dazu wurde eine neue virtuelle Festplatte in der Größe des APFS-Containers erstellt und ebenfalls zur VM hinzugefügt.
Nachdem der VM alle notwendigen virtuellen Festplatten hinzugefügt wurden, wurde diese gestartet. Die VM befand sich nun im Recovery Mode. In diesem wurde das Festplattendienstprogramm ausgewählt. Dort wurde die neu erstellte Festplatte ausgewählt und initialisiert. Anschließend wurde mithilfe des Punktes Wiederherstellen im Festplattendienstprogramm die einzelnen Volumes des APFS-Containers auf der neuen Festplatte wiederhergestellt. Dafür mussten alle verschlüsselten Volumes entschlüsselt werden, da ansonsten die Fehlermeldung in Abbildung 3.7 beim Wiederherstellen auftrat.

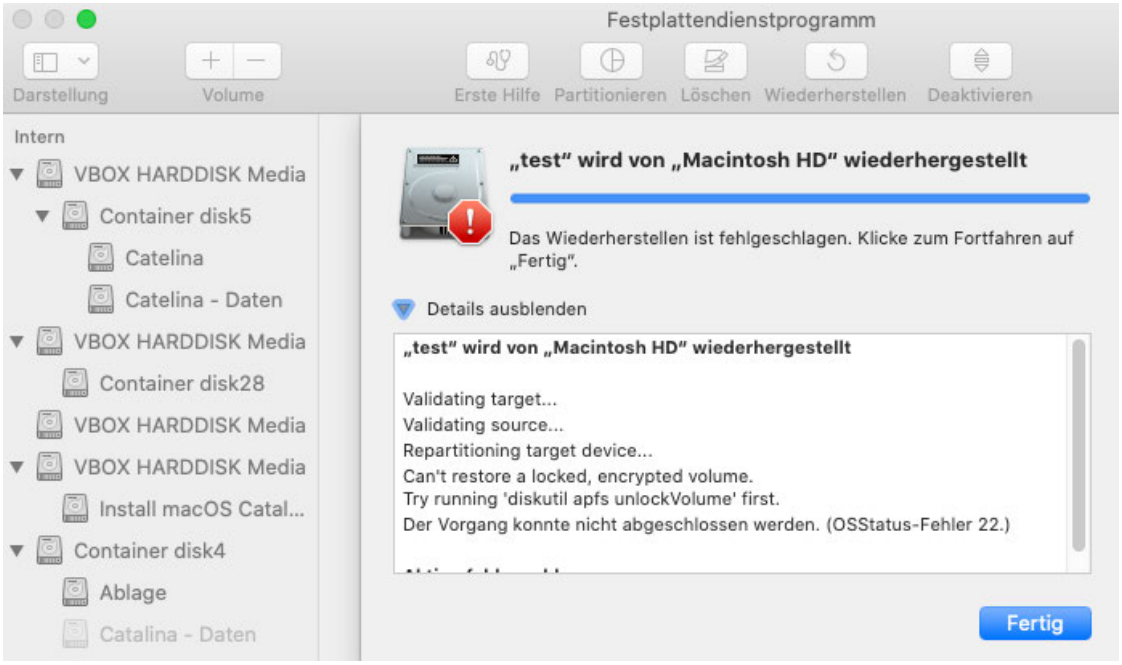


Abbildung 3.7: Fehlermeldung beim Versuch Macintosh HD auf der neu initialisierten Festplatte herzustellen. (Quelle: Eigene Darstellung)

Die Volumes wurden mithilfe des Terminals entschlüsselt. Im Terminal wurde dafür der Befehl *diskutil* verwendet.

```
1  diskutil apfs list
2  diskutil apfs UnlockVolume disk4s1
```

Der erste Befehl listet alle möglichen APFS-Container und ihre Volumes auf. Aus diesem musste das verschlüsselte Volume herausgesucht und dann mithilfe des zweiten Befehls entschlüsselt werden. Nach erfolgreichem Entschlüsseln der Volumes konnten diese auf der neuen Festplatte wiederhergestellt werden. Dabei war zu beachten, dass wenn mehrere Volumes wiederhergestellt werden sollten, bei jedem weiterem Volume nach dem Ersten, der Haken „*Auf neuem Volume wiederherstellen*“ aktiviert

wurde. Wurde dies nicht getan, wurden die bereits wiederhergestellten Volumes gelöscht und nur das neue wiederhergestellte Volume angezeigt.

Nach dem erfolgreichen Wiederherstellen des Containers konnte das gesicherte System als Startvolume verwendet werden. Ohne das Wiederherstellen des Containers auf einer neuen initialisierten Festplatte wurde das Startvolume des APFS-Containers zwar angezeigt, aber beim Versuch von diesem zu starten war eine Fehlermeldung aufgetreten. Das Volume konnte nicht als Startvolume ausgewählt werden. Nach dem erfolgreichen Starten des wiederhergestellten APFS-Containers tauchte der Anmeldebildschirm des gesicherten Rechners und nach erfolgreichem Anmelden der Nutzerbildschirm auf. Somit war die Virtualisierung ein Erfolg und der Bootstick und das Image können entfernt werden.

3.4.4 Methode 3: Virtualisierung von einem mit DigitalCollector gesichertem APFS-Container

Die Datensicherung von den Rechnern mit T2-Chip oder M1-Chip wurde mithilfe von DigitalCollector durchgeführt. Obwohl die Daten nicht mehr verschlüsselt sind, gestaltet sich die Virtualisierung der Datensicherung als problematisch und konnte nicht durchgeführt werden. Im Folgenden werden die auftretenden Probleme der Virtualisierung vom APFS-Container mit aktivem FileVault und ohne aktivem FileVault behandelt.

Ansatz 1: Virtualisierung des gesamten APFS-Containers mithilfe des Festplattendienstprogrammes

Bei dem Versuch einen APFS-Container mit aktivem FileVault zu virtualisieren trat dasselbe Problem auf wie beim Virtualisieren des APFS Containers von einem Rechner ohne Sicherheitschip. Das Startvolume des Containers konnte nicht direkt zum Starten verwendet werden und musste somit auf einer neuen Festplatte wiederhergestellt werden.

Bei dem Versuch den APFS-Container zu virtualisieren trat das Problem auf, dass die Daten-Volumes als verschlüsselt angezeigt wurden, unabhängig davon ob FileVault aktiviert war oder nicht.
Bei dem Rechner ohne aktivem FileVault war zwar das System-Volume entschlüsselt und wurde im Festplattendienstprogramm, wie in Abbildung 3.8, als gemountet angezeigt, die anderen Volumes wurden jedoch als deaktiviert angezeigt.

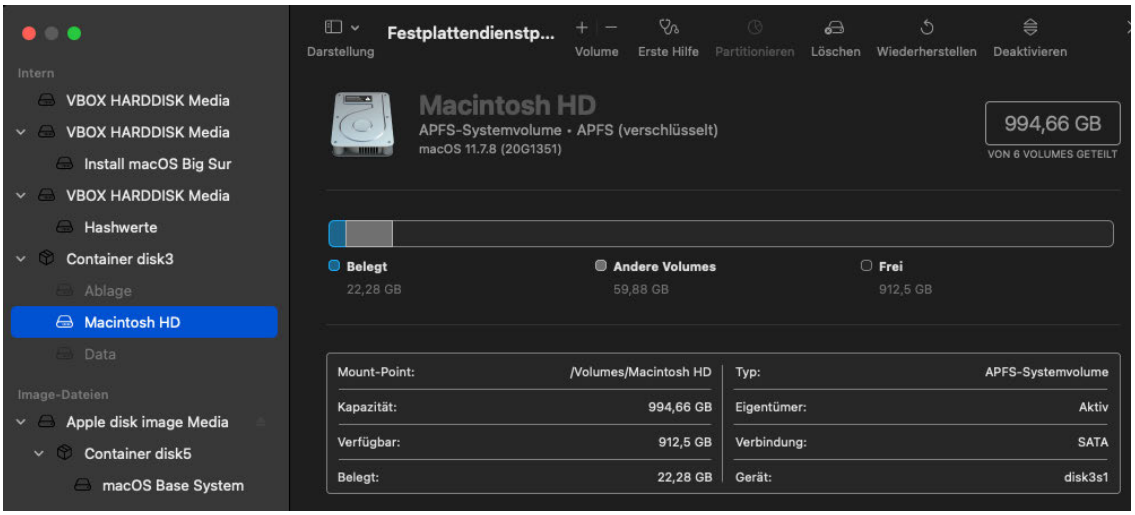


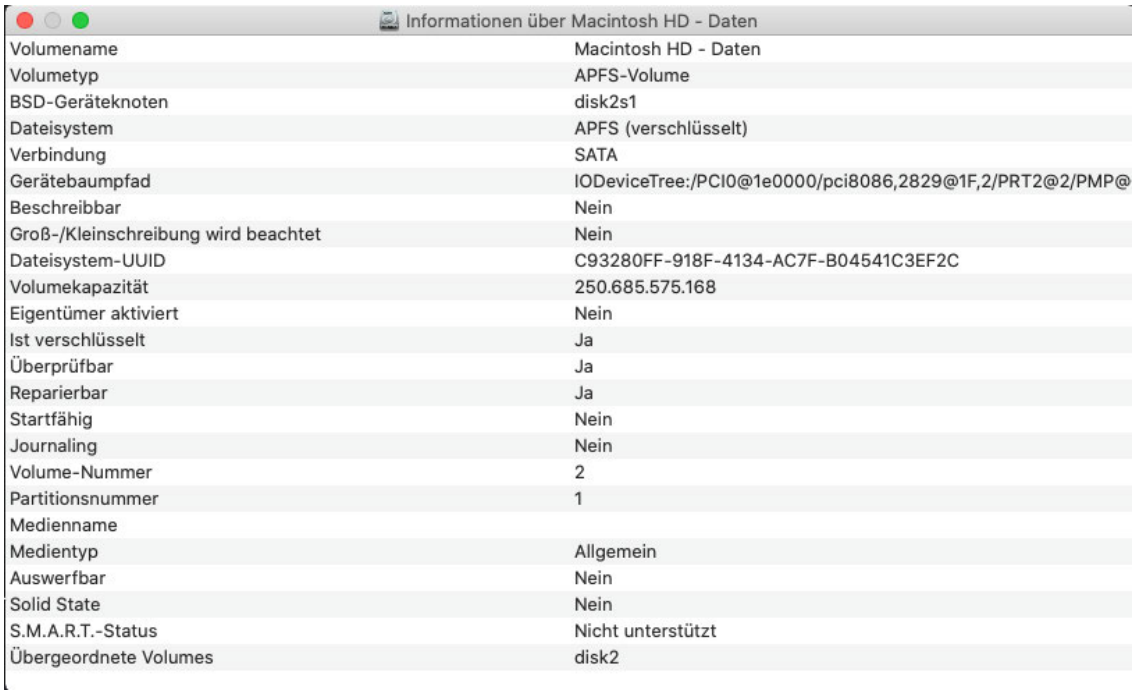
Abbildung 3.8: Darstellung des Festplattendienstprogramms bei einer Sicherung von einem APFS-Container mit DigitalCollector von einem Rechner ohne aktivem FileVault. Das System-Volume der Sicherung des APFS-Containers des Rechners ist aktiviert, alle anderen nicht. (Quelle: Eigene Darstellung)

Somit mussten die Volumes zunächst entschlüsselt werden, bevor sie auf der neuen Festplatte wiederhergestellt werden konnten. Das Entschlüsseln stellte jedoch ein Problem dar und war nicht möglich. Es wurde versucht, das Volume mittels des Terminals zu entschlüsseln. Jedoch kam trotz korrekter Eingabe des Passwortes die Meldung, dass das Passwort nicht korrekt sei. Beim Anschauen mithilfe des Terminals der User, die das Volume entschlüsseln können, wurden keine aufgeführt, wie in Abbildung 3.9 zu erkennen ist.

```
    Capacity Consumed:          12935499776 B (12.9 GB)
    FileVault:                  Yes (Locked)
-bash-3.2# diskutil apfs listCryptoUsers disk2s1
No cryptographic users for disk2s1
-bash-3.2#
```

Abbildung 3.9: Auszug aus dem Terminal in dem zuerkennen ist, dass FileVault aktiv ist, aber es keine User gibt, die das Recht haben das Volume zu entschlüsseln. (Quelle: Eigene Darstellung)

Auch die Eigenschaften des Volumes im Festplattendienstprogramm in Abbildung 3.10 zeigten, dass keine Owner für das Volume aktiviert waren. Somit war es unmöglich das Volume zu entschlüsseln, da kein Nutzer vorhanden war, der dafür die Rechte hatte. Daraus resultierend konnte der APFS-Container nicht wiederhergestellt und somit nicht virtualisiert werden.



Informationen über Macintosh HD - Daten

Volumename	Macintosh HD - Daten
Volumetyp	APFS-Volume
BSD-Geräteknoten	disk2s1
Dateisystem	APFS (verschlüsselt)
Verbindung	SATA
Gerätebaumpfad	IODeviceTree:/PCI0@1e0000/pci8086,2829@1F,2/PRT2@2/PMP@
Beschreibbar	Nein
Groß-/Kleinschreibung wird beachtet	Nein
Dateisystem-UUID	C93280FF-918F-4134-AC7F-B04541C3EF2C
Volumekapazität	250.685.575.168
Eigentümer aktiviert	Nein
Ist verschlüsselt	Ja
Überprüfbar	Ja
Reparierbar	Ja
Startfähig	Nein
Journaling	Nein
Volume-Nummer	2
Partitionsnummer	1
Medienname	
Medientyp	Allgemein
Auswerfbar	Nein
Solid State	Nein
S.M.A.R.T.-Status	Nicht unterstützt
Übergeordnete Volumes	disk2

Abbildung 3.10: Darstellung der Eigenschaften des Daten-Volumes. Zu erkennen ist, dass es verschlüsselt ist und keine Eigentümer aktiviert sind. (Quelle: Eigene Darstellung)

Ansatz 2: Extraktion des Daten-Volumes mithilfe von Inspector und Virtualisierung davon

Ein Ansatz die Probleme mit der Verschlüsslung zu umgehen wäre, lediglich das Daten-Volume zu virtualisieren. Dafür wurde mithilfe von Inspector [13], dem zugehörigen Datenaufbereitungsprogramm zu DigitalCollector [45], das AFF4 Image eingelesen und das Daten-Volume als Ordner exportiert. Problematisch dabei war, dass die versteckten Dateien mittels Unterstrich vor dem Dateinamen abgespeichert wurden und somit eine andere Bezeichnung als die eigentlichen Dateien haben. Das führte dazu, dass beim Versuch das Daten-Volume wie in Unterabschnitt 3.4.5 zu virtualisieren, die Installation sich aufgehängt hat und die versteckten Dateien doppelt existierten. Sie waren einmal mit Unterstrich vor dem Namen und einmal ohne Unterstrich vor dem Namen vorhanden. Dementsprechend war es nicht möglich durch diesen Ansatz das Daten-Volume zu virtualisieren.

Ansatz 3: Extraktion des Daten-Volumes mithilfe vom Aff4Imager und Virtualisierung davon

Eine weitere Möglichkeit war, das Daten-Volume mithilfe des Terminal-Programms AFF4Imager [10] zu extrahieren. Dies funktionierte jedoch ebenfalls nicht. Beim Eingeben des Befehls

```
aff4imager -e '*/data' --export_dir /tmp/export/ /tmp/test.aff4
```

war nichts passiert. Es gab keine Ausgabe im Terminal und die es wurden keine Daten extrahiert. Es war somit nicht möglich, ein AFF4-Image, das mit DigitalCollector erstellt wurde, zu virtualisieren.

3.4.5 Methode 4: Virtualisierung vom gesicherten Daten-Volume

Das Daten-Volume ist nur ein Teil des APFS-Containers und beinhaltet lediglich die System-Informationen, die verändert worden sind. Deshalb trat bei der Virtualisierung das Problem auf, dass das Daten-Volume mit den fehlenden Volumes des APFS-Containers verknüpft werden musste, um ein funktionierendes System zu erstellen. Beim Verknüpfen des Daten-Volumes mit den restlichen Volumes des APFS-Containers musste darauf geachtet werden, dass keine Vermischung der Daten der einzelnen Volumes entstand.

Ansatz 1: Wiederherstellung der Daten des gesicherten Daten-Volumes auf dem Daten-Volume der vollständig eingerichteten VM

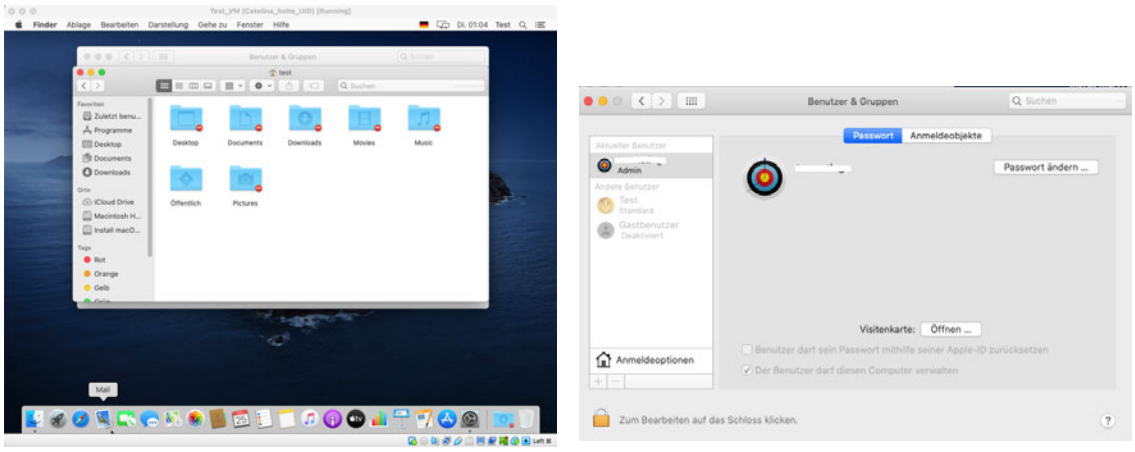
Ein Ansatz war es, eine vollständig eingerichtete VM mit dem passenden Betriebssystem für die Virtualisierung zu nutzen. Beim Verknüpfen des Daten-Volumes mit den restlichen Volumes des APFS-Containers musste darauf geachtet werden, dass keine Vermischung der Daten der einzelnen Volumes entsteht.
Eine Idee war es, ebenfalls das Festplattendienstprogramm und den Punkt Wiederherstellen zu nutzen, um das gesicherte Daten-Volume auf dem Daten-Volume der VM wiederherzustellen. Dies funktionierte jedoch nicht, da sich das Daten-Volume und das System-Volume der VM in einer Volumegruppe befinden und somit beim Wiederherstellen beide gelöscht wurden. Demzufolge fehlte das System-Volume und es konnte nicht mehr vom APFS-Container aus gestartet werden.

Ansatz 2: Kopieren der Daten des gesicherten Daten-Volumes auf das Daten-Volume der vollständig eingerichteten VM

Eine weitere Möglichkeit war mithilfe des Terminals im Recovery-Mode die Daten des gesicherten Daten-Volumes auf das Daten-Volume der VM zu kopieren und die vorhandenen Daten zu überschreiben.

```
cp -Rap /Volumes/Macintosh\ HD\ -\Daten/ /Volumes/Catalina\ -\
↪  Daten/
```

Wurden die Daten des Daten-Volumes der VM vor dem darauf Kopieren nicht gelöscht, entstand eine Vermischung der Nutzer des gesicherten Rechners und der VM. Hatten die Nutzer der VM und des gesicherten Rechners dieselbe ID, tauchte nur ein Nutzer als Anmeldeoption auf. In den Systemeinstellungen unter Benutzer&Gruppen wurden dennoch alle angezeigt. Bei unterschiedlicher ID wurden zwar beide Nutzer angezeigt, aber beim Anmelden des Nutzers der VM sind einige Benutzerordner gesperrt gewesen und der Nutzer wurde auf Standard zurückgesetzt, wie in Abbildung 3.11 zu erkennen ist.



(a) (b)

Abbildung 3.11: Probleme die durch den Kopiervorgang des Daten-Volumes entstehen. Abbildung (a) zeigt, dass die Userordner gesperrt sind, obwohl sie zum Home-Ordner gehören. Abbildung (b) zeigt, dass der Nutzer dem VM nur noch ein Standardnutzer ist. (Quelle: Eigene Darstellung)

Die Daten des Daten-Volumes der VM konnten mittels des Befehls

```
1 rm -Rf /Volumes/Catalina\ -\ Daten/
```

gelöscht werden. Wichtig dabei war, dass das System-Volume vor dem Kopieren unmountet wurde, da ansonsten die Fehlermeldung „*Operation not permitted*“ auftrat. Da bei dem Kopiervorgang einige Fehlermeldungen auftraten, funktionierte die Virtualisierung immer noch nicht fehlerfrei. Deshalb musste das Daten-Volume mithilfe des Festplattendienstprogramms formatiert werden und anschließend die Daten darauf kopiert werden. Problematisch dabei war, dass die Volumegruppe des Daten-Volumes und des System-Volumes durch das Formatieren aufgelöst wurde. Um das System dennoch starten zu können, musste das Betriebssystem mithilfe des Recovery-Modes neu installiert werden. Dafür musste das System-Volume zunächst entfernt werden. Nach erfolgreicher Installation war die Virtualisierung erfolgreich abgeschlossen.

Ansatz 3: Wiederherstellung der Daten des gesicherten Daten-Volumes auf neuer Festplatte und Installation vom Betriebssystem

Eine bessere Methode war es, lediglich eine konfigurierte Apple VM, ohne ein bereits installiertes Betriebssystem zu nutzen. Dadurch wurde im Vergleich zur vorherigen Lösung die Zeit des Installierens des Betriebssystems der VM gespart.

Für die Virtualisierung wurde eine neue virtuelle Festplatte benötigt, die die Größe des Daten-Volumes plus 30 GB besitzt. Dazu musste der passende Bootstick und das Image zur VM hinzugefügt werden. Nach erfolgreichem Hinzufügen der virtuellen Festplatten wurde die VM im Recovery-Mode gestartet. Die neue Festplatte wurde mithilfe des Festplattendienstprogramms initialisiert und anschließend das gesicherte Daten-Volume darauf wiederhergestellt. Um Verwirrungen zu vermeiden, wurde die VM ausgeschaltet und das gesicherte Daten-Volume entfernt. Anschließend konnte mithilfe des Recovery-Modes das Betriebssystem auf dem Daten-Volume installiert werden. Nach erfolgreicher Installation tauchte der Anmeldebildschirm des gesicherten Rechners auf und nach erfolgreicher Anmeldung war der Startbildschirm des Nutzers mit seinen gesamten Daten zusehen.

Somit war es möglich das Daten-Volume es APFS-Containers eines gesicherten Rechners zu virtualisieren.

3.4.6 Virtualisierung von gesicherten Ordnern

Aufgrund dessen, dass es nicht immer möglich ist ein Image vom Daten-Volume zu erstellen, existiert auch die Möglichkeit nur die einzelnen Ordner zu sichern und diese in der VM wieder zusammenzubauen und zu virtualisieren. Wichtig dabei ist, dass die Ordner vom Daten-Volume gesichert werden und nicht die, die im Finder angezeigt werden. Dazu sollten alle Ordner Users, Applications, Library und System gesichert werden und nicht nur der Nutzerordner.
Bei der Virtualisierung mithilfe der einzelnen Ordner traten weitere Probleme auf, da nun nicht mal mehr das Daten-Volume vorhanden war. Somit mussten die Ordner mit Strukturen von einem vorhandenen Daten-Volume und System kombiniert werden.

Ansatz 1: Users, Applications, System und Library Ordner auf Festplatte kopieren und Betriebssystem installieren

Ein erster Ansatz war es, die Ordner mithilfe des Terminals im Recovery-Mode auf eine neue initialisierte Festplatte zu kopieren und auf dieser anschließend das Betriebssystem zu installieren. Dabei kam es jedoch zu Fehlermeldung in Abbildung 3.12

und die Installation bricht ab. Die Fehlermeldung sagt, dass das Installationsprogramm Teile des Volumes als macOS System erkennt und dadurch Probleme hat das Betriebssystem zu installieren. Somit war es so nicht möglich, die Ordner zu virtualisieren.

Abbildung 3.12: Abbildung zeigt den Fehler, der bei der Installation auf dem Volume mit gesicherten Ordnern auftritt. Es werden Teile des Volumes als macOS erkannt. (Quelle: Eigene Darstellung)

Ansatz 2: Users, Applications und System durch Kopieren der Ordner auf Daten-Volume der VM virtualisieren

Für die Virtualisierung wird eine vollständig eingerichtete VM mit dem passenden Betriebssystem benötigt.
Beim Zusammenfügen der VM mit den einzelnen gesicherten Ordnern trat ebenfalls das Problem der Vermischung der Daten auf. Wurden die Daten mithilfe des Terminals im Recovery-Mode auf das Daten-Volume kopiert, kam es beim Kopieren zu Fehlermeldungen und eine Vermischung der User-Daten der erstellten VM mit denen des gesicherten Rechners.

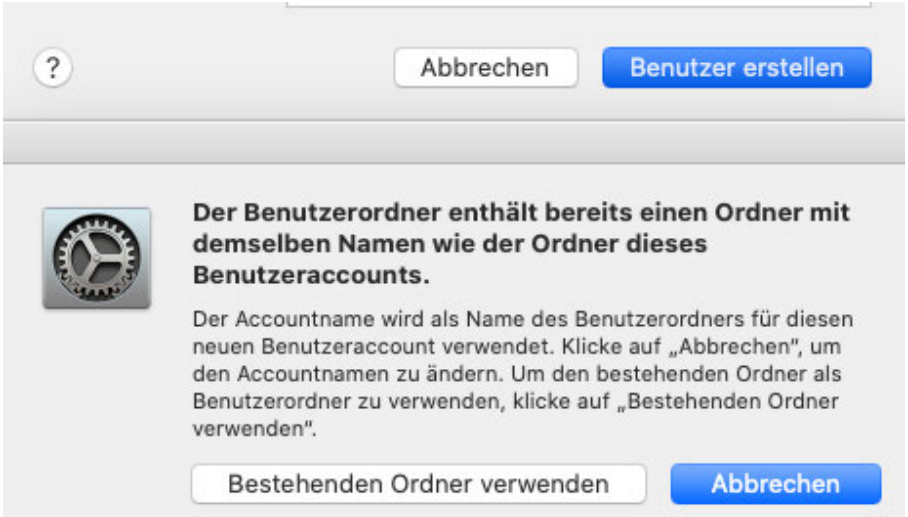


Abbildung 3.13: Beim Erstellen des neuen Nutzers mit demselben Nutzernamen den Nutzerordner übernehmen. (Quelle: Eigene Darstellung)

Dazu existierte der Nutzer des gesicherten Rechners nicht und es musste zunächst ein neuer Nutzer hinzugefügt werden. Es konnte zwar der Nutzerordner des gesicherten Rechners ausgewählt werden, wie in Abbildung 3.13, und der Nutzer mit diesem erstellt werden, jedoch funktionierten die Apps nicht.

Aus diesem Grund mussten die sich überschneidenden Daten zunächst auf dem Daten-Volume der VM mithilfe des Terminals gelöscht werden. Dafür musste im Festplattendienstprogramm das System-Volume entfernt werden. Wichtig beim Löschen war, dass nicht die Volumegruppe gelöscht wurde, sondern nur das System-Volume. Ansonsten wurde das Daten-Volume mit gelöscht. Anschließend musste mithilfe des Terminals der User-Ordner, der System-Ordner und der Applications-Ordner entfernt werden, damit keine Vermischung entsteht. Der Ordner Library konnte nicht entfernt werden, da er nicht gelöscht werden konnte.

```
1 rm -Rf /Volumes/Macintosh\ HD\ -\ Daten/Users
2 rm -Rf /Volumes/Macintosh\ HD\ -\ Daten/Applications
3 rm -Rf /Volumes/Macintosh\ HD\ -\ Daten/System
```

Danach konnten die gesicherten Ordner auf das Daten-Volume kopiert werden. Der Ordner Library wurde nicht kopiert, da beim Kopieren Fehlermeldungen entstanden, dass der Ordner Library nicht existiert.

```
1 cp -Rap /Volumes/Users/ /Volumes/Macintosh\ \HD/User
2 cp -Rap /Volumes/Applications/ /Volumes/Macintosh\ \HD/Aplications
3 cp -Rap /Volumes/System/ /Volumes/Macintosh\ \HD/System
```

Nach dem Kopieren wurde das Betriebssystem neu installiert. Die Installation hing sich bei „Noch ungefähr 9 Minuten ...“ auf und deshalb musste der Nvram gelöscht und das System erneut installiert werden. Nach der Installation tauchte der Nutzer der VM auf, aber nicht der Nutzer des gesicherten Rechners.
Aus diesem Grund musste ein neuer Nutzer angelegt werden, der den vorhandenen Nutzerordner nutzt. Dieser beinhaltet die Daten des Nutzers des gesicherten Rechners. Anschließend konnte der Nutzer der VM entfernt werden und die Virtualisierung war abgeschlossen.

Ansatz 3: Users, Applications, System und Library Ordner und Daten vom Daten-Volume auf neue Festplatte kopieren und Betriebssystem installieren

Der Nachteil an dem Ansatz 2 ist, dass der Ordner Library nicht virtualisiert wurde. Ein Ansatz den Ordner Library ebenfalls mit zu virtualisieren war, auf eine neue Festplatte die gesicherten Ordner und die fehlenden Dateien des Daten-Volumes vom Daten-Volume der VM zu kopieren.

Die Festplatte musste dafür die Größe der gesicherten Ordner plus die Größe des Daten-Volumes und ungefähr weitere 30 GB besitzen. mithilfe des Festplattendienstprogramms musste die Festplatte initialisiert werden. Das APFS-Volume wurde Macintosh HD genannt. Anschließend wurden mithilfe des Terminals zunächst die gesicherten Ordner auf die Festplatte kopiert

```
cp -Rap /Volumes/Users/ /Volumes/Macintosh\ \HD/User
cp -Rap /Volumes/Applications/ /Volumes/Macintosh\ \HD/Aplications
cp -Rap /Volumes/Library/ /Volumes/Macintosh\ \HD/Library
cp -Rap /Volumes/System /Volumes/Macintosh\ \HD/System
```

und danach die fehlenden Daten vom Daten-Volume der VM. Eine Möglichkeit dafür war das ganze Daten-Volume auf das andere Daten-Volume zu kopieren.

```
cp -Ranp /Volumes/Catalina\ -\ Daten/ /Volumes/Macintosh\ \HD
rm -Rf /Volumes/Macintosh\ \HD/Users/test
```

Dabei wurde der Parameter -n benutzt, damit keine Daten überschrieben werden. Bei dieser Möglichkeit traten jedoch Fehlermeldungen beim Kopieren auf und der User Ordner der VM befand sich zum Beispiel ebenfalls im Ordner Users. Somit waren die Daten der gesicherten Ordner mit den gleichen Ordnern des Daten-Volumes der VM vermischt.
Deshalb mussten alle Ordner inklusive versteckte Dateien und Ordner des Daten-Volumes einzeln auf das Volume der neuen Festplatte kopiert werden. Dabei mussten die Ordner Application, Library, System und Users weggelassen werden.

```
cp -Rap /Volumes/Catalina\ -\ Daten/.Spotlight-V100/
 ↪  /Volumes/Macintosh\ \HD/
cp -ap /Volumes/Catalina\ -\ Daten/.TempReceipt.bom
 ↪  /Volumes/Macintosh\ \HD/
cp -Rap /Volumes/Catalina\ -\ Daten/.TemporaryItems/
 ↪  /Volumes/Macintosh\ \HD/
cp -Rap /Volumes/Catalina\ -\ Daten/.fseventsd /Volumes/Macintosh\
 ↪  \HD/
cp -Rap /Volumes/Catalina\ -\ Daten/.installer-compatibility
 ↪  /Volumes/Macintosh\ \HD/
cp -Rap /Volumes/Catalina\ -\ Daten/Volumes /Volumes/Macintosh\
 ↪  \HD/Volumes
cp -Rap /Volumes/Catalina\ -\ Daten/cores /Volumes/Macintosh\
 ↪  \HD/cores
cp -Rap /Volumes/Catalina\ -\ Daten/home /Volumes/Macintosh\
 ↪  \HD/home
```

```
cp -Rap /Volumes/Catalina\ -\ Daten/mnt /Volumes/Macintosh\ \HD/mnt
cp -Rap /Volumes/Catalina\ -\ Daten/opt /Volumes/Macintosh\ \HD/opt
cp -Rap /Volumes/Catalina\ -\ Daten/private /Volumes/Macintosh\
 ↪  \HD/private
cp -Rap /Volumes/Catalina\ -\ Daten/sw /Volumes/Macintosh\ \HD/sw
cp -Rap /Volumes/Catalina\ -\ Daten/usr /Volumes/Macintosh\ \HD/usr
```

Anschließend wurde die Festplatte der eingerichteten VM entfernt, um Verwirrungen zu vermeiden und das Betriebssystem wurde auf dem Volume der Festplatte installiert. Die Installation hing sich jedoch ebenfalls bei „Noch ungefähr 9 Minuten ...“ auf und konnte auch nicht durch ein erneutes Installieren vollendet werden.

Somit war es nicht möglich den Ordner Library mit zu virtualisieren, sondern es können nur die Ordner Applications, User und System virtualisiert werden.

3.5 Skript zum Berechnen und Vergleichen von Hashwerten

Datenintegrität spielt in der IT-Forensik eine wichtige Rolle. Aus diesem Grund wurden zum Überprüfen der Unversehrtheit der Daten die Hashwerte der einzelnen Dateien und daraus ein Hashwert für alle Dateien berechnet. Dazu wurde der Hashwert mit folgendem Befehl in der VM im Terminal des Recovery-Modes für die Sicherung vor und nach der Virtualisierung bestimmt.

```
find -s <pfad> -type f -exec md5 {} \;
 ↪  >>/Volumes/Hashwerte/checksum.txt
```

Um die Hashwerte abzuspeichern, wurde eine neue virtuelle Festplatte erstellt, auf welcher die Hashwerte in einer Datei abgespeichert wurden.

Damit von den Dateien aus dem APFS-Container der Hashwert berechnet werden konnten, mussten die Volumes des APFS-Containers erst an einem neuen Mountpoint gemountet werden. Dafür wurden die Volumes mithilfe des Befehls

```
diskutil apfs UnlockVolume disk4s1 -nomount
```

entschlüsselt und durch den Parameter -nomount nicht gemountet. Anschließend wurde ein gemeinsamer Mountpoint mithilfe von *mkdir* erstellt, der für jedes Volume einen eigenen mit *mkdir* erstellten Mountpoint besaß. An diesen Mountpoints wurde jedes Volume mithilfe von *diskutil mount* als ReadOnly gemountet. Anschließend

wurde mithilfe des *find* Befehls rekursiv nach allen Dateien beginnend im Mountpoint mit allen unteren Mountpoints gesucht und auf jede gefundene Datei der *md5* Befehl angewendet.

```
1 find -s /Volumes/Mountpoint -type f -exec md5 {} \; >>
  ↪ /Volumes/Hashwerte/checksum.txt
```

Die Dateien mussten zunächst mit dem *find*-Befehl gesucht werden, da der md5-Befehl lediglich auf einzelne Dateien funktioniert und nicht auf Ordner. Aus diesem Grund wurden auch die einzelnen Mountpoints erstellt, um nur einen *find*-Befehl für das gesamte Volume verwenden zu müssen. Die mit dem md5-Befehl erstellten Ausgaben wurden in einer Textdatei abgespeichert, die auf der neu erstellten Festplatte abgespeichert wurden.
Damit die Dateien der Hashwerte anschließend auf dem Host-Rechner mithilfe eines Pythonskriptes miteinander verglichen konnten, mussten diese zunächst aus der VM heraus geholt werden. Dafür wurde auf dem Hostrechner mithilfe des Festplattendienstprogrammes ein leeres Image erstellt und dieses anschließend wie in Unterabschnitt 3.2.2 in eine VDI-Datei umgewandelt. Auf diese wurde dann mithilfe des Terminals im Recovery-Mode der Festplatte die erstellten Textdateien mit den Hashwerten abgespeichert. Anschließend wurde das erstellte Image auf dem Hostrechner gemountet und die Daten herunter kopiert.

Für den Vergleich der Hashwerte wurde eine Pythonskript (Anhang B) entwickelt, welches die Ausgabe der einzelnen md5 Befehle, die in den Textdateien von vor und nach der Virtualisierung abgespeichert wurden, einliest und die Hashwerte und den dazugehörigen Pfad extrahiert und in einer Liste pro Textdatei speichert.
Es wurde eine angepasste Methode des Mergesorts [40] geschrieben, die die Liste der Hashwerte mit dem dazugehörigen Pfad nach den Hashwerten für ein schnelleres Vergleichen sortiert.
Dazu wurde eine Methode entwickelt, die die Listen in Teillisten aufteilt, um mithilfe von Multi-Processing die Hashwerte vergleichen zu können. Für das eigentliche Vergleichen wurde ebenfalls eine Methode geschrieben, die nach einem passenden Hashwert der ersten Liste in der zweiten Liste sucht.
Das Pythonskript gibt die Hashwerte der beiden Listen und die Gesamtanzahl der nicht übereinstimmenden Dateien und die gesamte Anzahl an Dateien aus. Dazu erstellt das Skript eine Textdatei, in der alle Hashwerte inklusive Pfad abgespeichert werden, die nicht übereinstimmen.

Das Ergebnis des Skriptes wurde benutzt, um im folgenden Kapitel die Datenintegrität der einzelnen Methoden zu bewerten.

4 Ergebnisse und Diskussion

In diesem Kapitel wird dargestellt, welche Methoden mit welchen Ansätzen funktioniert haben und wie diese anschließend bewertet werden. Dazu wird auf die Datenintegrität der einzelnen Methoden eingegangen. Danach werden die entwickelten Methoden mit anderen Möglichkeiten der Datenaufbereitung verglichen.

4.1 Ergebnis der verschiedenen Ansätze der unterschiedlichen Sicherungsarten

In dem vorherigen Kapitel wurden unterschiedliche Methoden durch unterschiedliche Ansätze entwickelt. Tabelle 4.1 zeigt, welche Ansätze bei welcher Sicherungsart funktionieren und somit valide Methoden sind, um die Sicherungsart zu virtualisieren.

Tabelle 4.1: Übersicht über den Erfolg der einzelnen Ansätze der verschiedenen Sicherungsarten. Der „✓“ bedeutet, dass der Ansatz funktioniert und das „-“ bedeutet, dass er nicht funktioniert.

Methode	Sicherungsart	Ansatz	Erfolg
1	Festplatte (Intel Chip ohne T2-Chip)	(3.4.2)	-
2	APFS-Container (Intel Chip ohne T2-Chip)	(3.4.3)	✓
3	APFS-Container (DigitalCollector)	Ansatz 1 (3.4.4)	-
		Ansatz 2 (3.4.4)	-
		Ansatz 3 (3.4.4)	-
4	Daten-Volume	Ansatz 1 (3.4.5)	-
		Ansatz 2 (3.4.5)	✓
		Ansatz 3 (3.4.5)	✓
5	Ordner	Ansatz 1 (3.4.6)	-
		Ansatz 2 (3.4.6)	✓
		Ansatz 3 (3.4.6)	-

Es fällt auf, dass es momentan zum einen nicht möglich ist, APFS-Container zu virtualisieren, die mit DigitalCollector [53] gesichert wurden und zum anderen nicht möglich ist, die Sicherung einer gesamten Festplatte zu virtualisieren.

Das Problem bei der Virtualisierung der Sicherung des APFS-Containers, der mit DigitalCollector gesichert worden ist, ist, dass die Daten des APFS-Containers zwar eigentlich entschlüsselt sind, die VM die einzelnen Volumes des gesicherten Containers aber nicht mounten kann, da FileVault noch als aktiviert angesehen wird. FileVault kann aber nicht deaktiviert werden, da keine Benutzer vorhanden sind, die dazu die Berechtigungen haben. Es ist nicht möglich Benutzer über die VM hinzuzufügen. Die einzige Möglichkeit, um den gesamten APFS-Container zu virtualisieren wäre, die FileVault Flag zu deaktivieren oder die Benutzerkonten zu aktiveren. Dafür müsste aber auf die Daten im Image zugegriffen werden und anschließend die Flags, die für FileVault zuständig sind, deaktiviert werden. Eine Möglichkeit auf die Daten des Daten-Volumes zuzugreifen wäre mithilfe des AFF4Imagers, jedoch kann mit diesem keine Daten extrahiert werden. Aus diesem Grund wäre es für die Zukunft sinnvoll, eine Methode zu entwickeln, die es ermöglicht FileVault zu deaktivieren oder die Benutzer wieder zu aktiveren.

Eine andere Möglichkeit war es nur das Daten-Volume des gesicherten Containers zu virtualisieren. Dafür sollte mithilfe von Inspector [13] das Daten-Volume als Ordner extrahiert werden. Problematisch dabei war, dass die versteckten Dateien als `_.<Dateiname>` gesichert werden und somit nicht mehr dieselben Dateinamen besitzen. Das führt dazu, dass das Daten-Volume nicht wie die anderen gesicherten Daten-Volumes virtualisiert werden kann, da sich die Installation nach einer gewissen Zeit aufhängt.
Das Problem bei der Virtualisierung des APFS-Containers liegt vermutlich daran, wie die Sicherung vom DigitalCollector durchgeführt und als AFF4-Image abgespeichert wurde. Aus diesem Grund konnten neben dem Versuch der Virtualisierung mithilfe des Recovery-Modes auch keine Daten mithilfe des Aff4Imagers [10] extrahiert werden, obwohl dieser klassischerweise dafür verwendet wird. Um das Problem genauer klassifizieren zu können, müsste somit verstanden werden, wie DigitalCollector die Datensicherung konkret durchführt. Dafür wäre eine tiefere Analyse der Software notwendig. Eine Möglichkeit, die Datensicherung mithilfe von Inspector [13] zu virtualisieren, wäre es, die von Inspektor erzeugten Unterstriche mithilfe eines Programms wieder zu entfernen. Diese Idee wurde jedoch im Rahmen dieser Arbeit nicht weiter verfolgt.

Bei der Virtualisierung der gesamten Festplatte liegt das Problem, dass die Festplatte nicht korrekt erkannt wird, vermutlich an der Sektorgröße, die nicht der Standardgröße von 512 Byte pro Sektor entspricht. Dazu könnte es an der Kombination aus der EFI-Partition und des APFS-Containers liegen, da die Virtualisierung von einem USB-Stick ohne EFI-Partition mit APFS-Container funktioniert.

Für die Virtualisierung eines APFS-Containers von einem Rechner mit Intel Chip ohne T2-Chip und für die Virtualisierung von gesicherten Ordnern konnte eine Methode entwickelt werden, die es ermöglicht die Sicherung zu virtualisieren. Bei der Daten-

sicherung von den Ordnern Applications, Users, Library und System funktioniert die Virtualisierung jedoch nur mit drei der vier Ordnern. Der Ordner Library kann nicht mit virtualisiert werden, da beim Kopieren Fehlermeldungen auftreten. Diese liegen vermutlich daran, dass das System-Volume und das Daten-Volume miteinander verknüpft sind und somit der Ordner vorhanden ist, obwohl er eigentlich nicht vorhanden ist. Dies führt dazu, dass der Ordner nicht entfernt werden kann, aber auch keine Daten in den vorhandenen Ordner kopiert werden können, da dieser angeblich nicht existiert.

Die Sicherung des Daten-Volumes funktioniert bei zwei der 3 erarbeitenden Ansätzen. Der Ansatz 3 ist im Vergleich zum Ansatz 2 effektiver und schneller. Dies liegt daran, dass bei Ansatz 2 zunächst in der VM das Betriebssystem installiert werden muss und anschließend nach dem Kopieren der Daten das System erneut installiert werden muss. Beim Ansatz 3 wird das Betriebssystem jedoch lediglich einmal installiert. Deshalb sollte als bevorzugte Methode für die Virtualisierung des Daten-Volumes der dritte Ansatz verwendet werden. Das Daten-Volume muss auf einem anderen Volume wiederhergestellt werden, da die Festplatte des Images nicht genügend Platz besitzt, damit das System darauf installiert werden kann. Bei beiden Ansätzen hatten die Volumes nach der Installation des Betriebssystems verschiedene Namen. Das System-Volume war das Volume mit „Daten“ im Namen und das Daten-Volume war das Volume mit „Daten - Daten“ im Namen. Dies liegt daran, dass das ehemalige Daten-Volume zum System-Volume wurde, da es als Installationsvolume ausgewählt wurde. Während der Installation wurde ein neues Daten-Volume erstellt, welches den Namen des System-Volumes plus die Endung Daten besitzt.

4.2 Bewertung der Güte der Umwandlung der Sicherungen in ein VDI-Image

Bevor überhaupt die Sicherungen virtualisiert werden können, müssen sie zunächst in eine VDI-Datei umgewandelt werden, damit sie der VM als virtuelle Festplatte dienen können. Dies war für Sicherungen des Datentyps EWF, RAW, CDR und DMG mithilfe von *xmount* und *hdiutil* möglich, jedoch konnte es einige Zeit dauern. Im Gegensatz dazu konnten die Sicherungen, die als AFF4 Image gespeichert wurden, nicht mehr einfach umgewandelt werden.
Zwar bietet der Aff4Imager [10] die Möglichkeit Datenstreams zu extrahieren, jedoch funktioniert dies nicht mit den Datensicherungen, die mithilfe der Software DigitalCollector [53] durchgeführt wurden. Der Aff4Imager nutzt osxpmem [52], ein Programm, dass es ermöglicht AFF4 Images zu erstellen und Streams daraus zu extrahieren. Das Problem dabei ist, dass es lediglich die Komprimierung zlip [1] und snappy [42] unterstützt, aber nicht lz4 [11], mit welcher das AFF4 Image komprimiert ist. Neben der Möglichkeit den Aff4Imager unter macOS zu nutzen, gibt es das äquivalente Programm auch für Windows. Dieses nutzt Winpmem [44], welches auch die lz4

Komprimierung unterstützt. Dennoch kann damit nicht auf einfache Art und Weise die Daten aus dem AFF4 Image extrahieren werden. Normalerweise kann mithilfe von *winpmem.exe* und dem Parameter *-e* einzelne Daten aus dem AFF4 Image extrahiert werden [28]. Dies funktioniert aber aufgrund des Aufbaus des Images vom DigitalCollector nicht. Normalerweise ist ein AFF4 Kategorie „physical“ vorhanden, in dem der Data Stream gespeichert ist [28]. Bei dem Image von DigitalCollector ist der Data Stream jedoch direkt gespeichert, wie in Abbildung 4.1 zu erkennen ist.

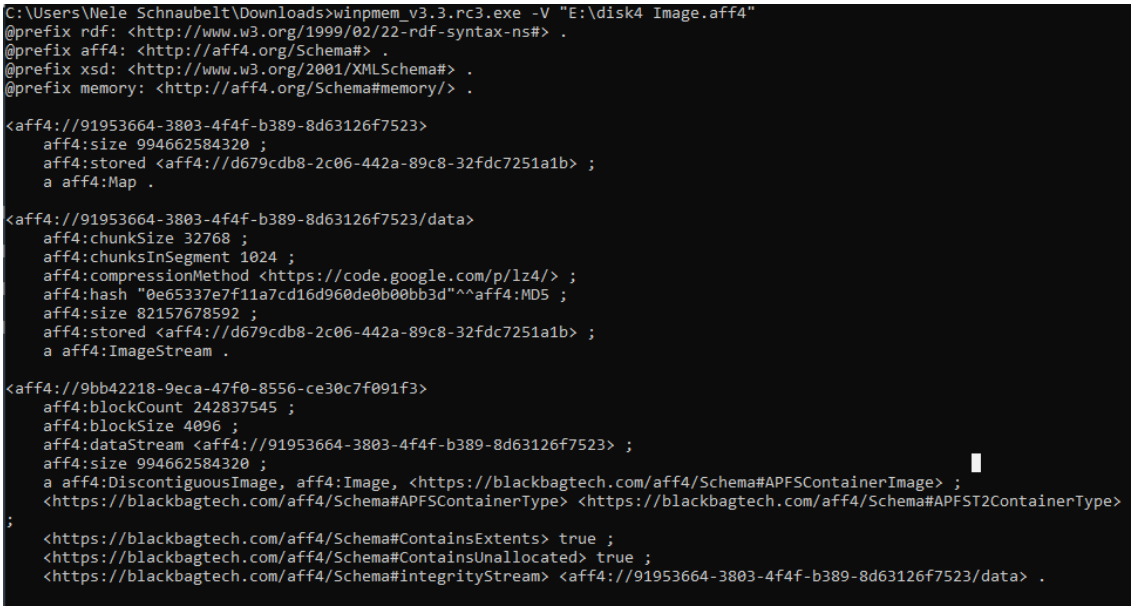


Abbildung 4.1: Aufbau des AFF4Images von einer Sicherung mit DigitalCollector von dem MacBook Pro mit M1-Chip. Es besitzt kein Physical Stream. (Quelle: Eigene Darstellung)

Aus diesem Grund muss beim *winpmem*-Befehl Stream direkt der Data-Stream ausgewählt werden. Jedoch erzeugt der Befehl keine Ausgabe und es werden keine Daten extrahiert.
Eine weitere Möglichkeit ist mithilfe von X-Ways Forensics die Daten in ein e01 Image umzuwandeln. Der Nachteil an X-Ways Forensics ist, dass es ein Programm ist, dass nur unter Windows läuft und somit ein weiterer Untersuchungsrechner zum Konvertieren benötigt wird.

4.3 Vergleich der einzelnen Methoden

Im Folgenden werden die Methoden der unterschiedlichen Sicherungsarten miteinander verglichen. Dabei wird auch auf den Datenverlust der einzelnen Sicherungen und somit der Datenverlust der Virtualisierungen eingegangen. Jedoch wird noch nicht auf die Veränderung der gesicherten Daten eingegangen. Dies wird im folgenden Abschnitt 4.4 behandelt.

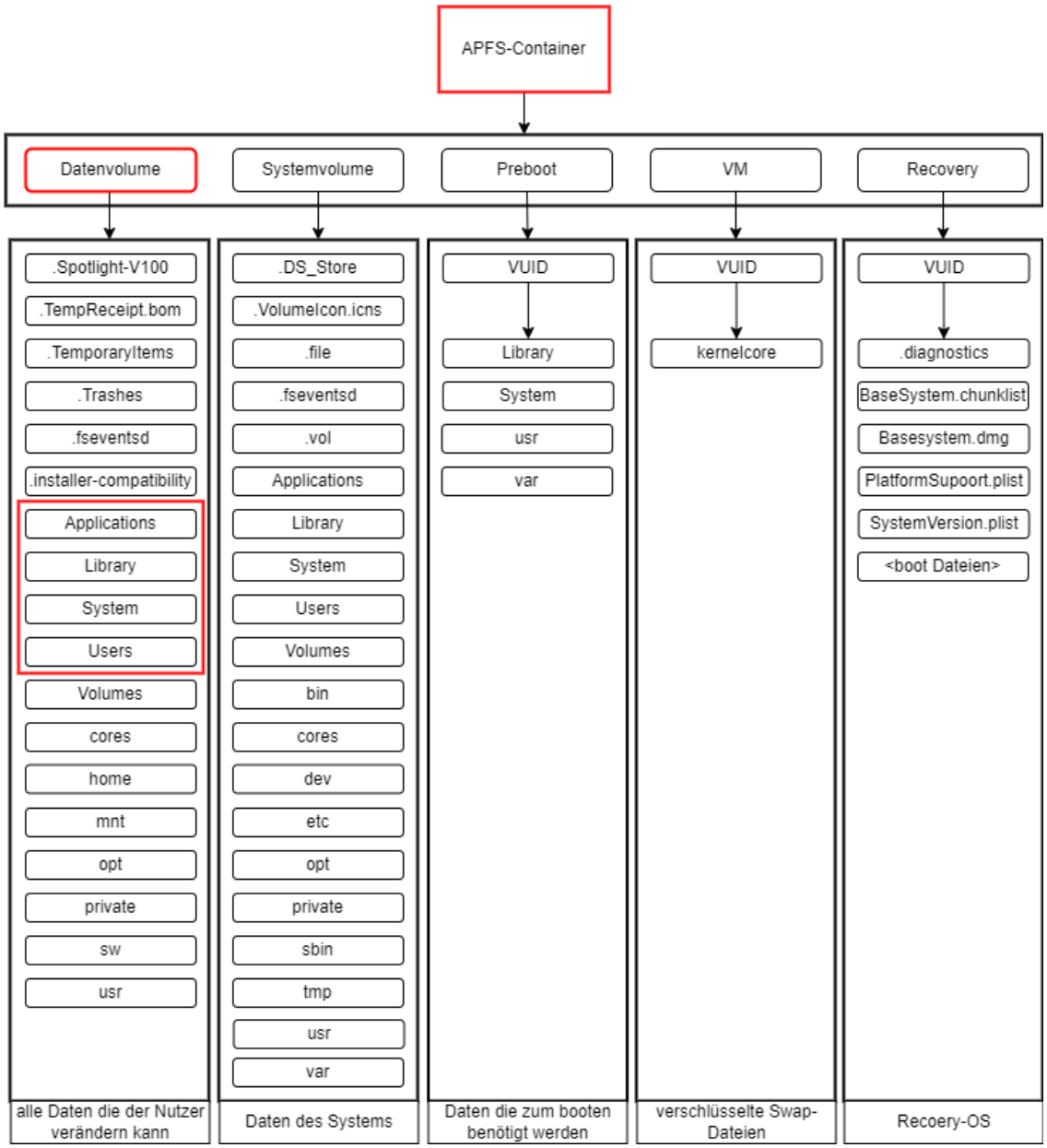


Abbildung 4.2: Darstellung des Datenverlustes der einzelnen Methoden. Der Aufbau kann je nach Betriebssystem leicht abweichen. Die rot umrandeten Felder sind Daten, die gesichert wurden. VUID: eindeutige Volume ID. (Quelle: Eigene Darstellung)

Betrachtet man die einzelnen Methoden, mussten alle erst mithilfe des Festplattendienstprogrammes und des Terminals im Recovery Mode bootfähig gemacht werden. Dadurch hat das Virtualisieren je nach Methode einige Zeit in Anspruch genommen. Wichtig bei allen Methoden ist, dass das korrekte Betriebssystem als Bootstick verwendet wird, da es ansonsten zu Problemen bei der Virtualisierung gekommen ist.

Das Virtualisieren eines APFS-Containers eines Intel-Rechners, ausgenommen einer Sicherung mit DigitalCollector, war einfach und im Vergleich zu den anderen Metho-

den nicht sehr zeitaufwändig. Lediglich musste der APFS-Container in der VM auf einer neuen initialisierten Festplatte wiederhergestellt werden. Anschließend konnte das System normal gestartet und verwendet werden. Das Starten der VM hat dennoch etwas gedauert.
Bei der Sicherung des APFS-Containers sollte kein Datenverlust gegenüber des gesicherten Rechners entstehen. In Abbildung 4.2 werden die einzelnen Volumes des APFS-Containers mit ihrem Inhalt und Funktionen dargestellt. Die rot umrandeten Felder sind die Datensicherungen, die virtualisiert wurden. Jedoch fällt beim Vergleich des APFS-Containers vor und nach dem Wiederherstellen auf, dass das VM-Volume nicht wiederhergestellt wurde. Das VM-Volume beinhaltet Swap-Dateien. Ein Swap-Speicher ist ein Teil des virtuellen Speichers, der genutzt wird, um inaktive Daten, die für den späteren Abruf verwendet werden, zu speichern [63]. Auf die Datenintegrität des Images während der Virtualisierung wird im Abschnitt 4.4 genauer eingegangen. Dazu stimmten nicht alle Informationen in der VM mit denen des gesicherten Rechners überein. Aufgrund dessen, dass die angezeigten Informationen zur Hardware direkt von der VM von dieser ausgelesen werden, werden in der VM die Informationen angezeigt, die zur Hardware der VM gehören und nicht zu denen des gesicherten Rechners. Somit stimmten die Seriennummer und die Informationen der CPU nicht mit dem des gesicherten Rechners über ein. Theoretisch bietet VirtualBox [61] die Möglichkeit an, mithilfe von VBoxManage die Seriennummer zu ändern. Jedoch ist dies nicht notwendig, da es nicht die Sichtung der Daten beeinflusst.

Rechner mit Sicherheitschips konnten nicht mehr so gesichert und virtualisiert werden wie Rechner nur mit Intel-Chip. Würde man genauso vorgehen wie bei der Sicherung des Rechners nur mit Intel-Chip, konnte FileVault in der VM nicht deaktiviert werden, da sich der Sicherheitsschlüssel zum Entsperren aus einer Schlüsselhierarchie in der Secure Enclave zusammen setzt und somit der Sicherheitschip zum Entschlüsseln benötigt wird [5]. Aus diesem Grund wurde die Sicherung mittels DigitalCollector [53] durchgeführt. Es konnte jedoch keine Methode entwickelt werden, die die Sicherung mit DigitalCollector von einem Apple-Rechner virtualisiert, da die Volumes mit und ohne aktivem FileVault, FileVault angeblich aktiviert hatten und dieses nicht deaktiviert und somit die Volumes nicht entschlüsselt werden konnten.

Die Virtualisierung eines Daten-Volumes ist deutlich zeitaufwändiger, als die eines APFS Containers. Dies liegt daran, dass nicht nur das Daten-Volume auf einem neuen Volume einer initialisierten Festplatte wiederhergestellt werden, sondern auch das Betriebssystem neu installiert werden musste. Somit besteht nicht nur die Zeit fürs Wiederherstellen, sondern auch für das Installieren.
Bei der Virtualisierung des Daten-Volumes gingen im Vergleich zum APFS-Container zusätzlich die Informationen über das System verloren, da dieses nicht mit gesichert wurden (Abb. 4.2). Dazu gehören die Informationen zum Starten des Systems, die verschlüsselten Swap-Dateien und das Recovery-OS. Die Daten vom User bleiben erhalten. Somit sollten die Daten übereinstimmen, die der Nutzer erstellt hat, jedoch

waren die Informationen über das Betriebssystem anders, da diese nun aus der Erstellung der VM stammen und nicht mehr aus dem gesicherten Rechner stammten. Alle Veränderungen, die der Nutzer vorgenommen hat, sollten dennoch vorhanden sein, da diese auf dem Daten-Volume abgespeichert werden. In Abschnitt 4.4 wird auf die exakten Unterschiede des Daten-Volumes vor und nach der Virtualisierung eingegangen.

Am aufwändigsten war die Virtualisierung von gesicherten Ordnern, da dort die einzelnen Ordner und Dateien zu einem Daten-Volume verbunden werden mussten. Somit muss der Aufwand der einzelnen Kopier- aber auch Löschvorgänge und die Zeit, die für die Installation benötigt wird, mit berücksichtigt werden.
Dadurch, dass kein vollständiges Daten-Volume vorhanden war, gingen im Vergleich zum Daten-Volume noch mehr Informationen über den gesicherten Rechner verloren (Abb. 4.2). Auf einem Daten-Volume sind grundsätzlich die Ordner und Dateien `.Spotlight-V100`, `.TempReceipt.bom`, `.TemporaryItems`, `.Trashes`, `.fsevenstd`, `.installer-compatibility`, `Applications`, `Library`, `System`, `Users`, `Volumes`, `cores`, `home`, `mnt`, `opt`, `private`, `sw` und `usr` vorhanden (Abb. 4.2). Die Ordner `System`, `Applications` und `Users` werden von den Sicherungen der Ordner ersetzt und somit ist egal, was in diesen vorhanden war. Der Ordner `Library` kann nicht mit virtualisiert werden. In diesem werden Einstellungen gespeichert, die von Programmen verwaltet werden. Da der Ordner Volumes alle angeschlossenen Volumes beinhaltet, enthält dieser keine Informationen, die für die korrekte Virtualisierung wichtig sind. Die Ordner `cores`, `home`, `mnt`, `opt`, `private` und `sw` sind leer. Der Ordner `usr` beinhaltet die Ordner `libexec`, `local` und `share`. Dabei ist der Ordner `local` ebenfalls leer. Allgemein sind im Ordner `usr` weitere Informationen über das System gespeichert, die Programmen zur Verfügung gestellt werden, vorhanden. Die versteckten Dateien `.Spotlight-V100`, `.TempReceipt.bom`, `.TemporaryItems` und `.fsevenstd` beinhalten Informationen über temporäre Dateien und Veränderungen von Dateien [23]. Der Ordner `.Trashes` beinhaltet die gelöschten Dateien des Volumes [30], aber nicht die gelöschten Dateien, die sich im Papierkorb befinden. Diese sind im Userordner gespeichert. Somit kann angenommen werden, dass alle nutzerrelevanten Informationen vorhanden sind, die sich angeschaut werden können, außer die Einstellungen in Programmen.

Bei der Virtualisierung von gesicherten Apple Rechnern muss einem bewusst sein, dass nur bei der Sicherung des APFS-Containers alle Informationen in die VM übertragen werden, lediglich die Informationen zur Hardware und zum Swap-Speicher gehen verloren. Bei allen anderen Sicherungen hat man einen Datenverlust der Informationen über das System. Dazu kommt, dass es einige Zeit in Anspruch nimmt, die Datensicherungen zu virtualisieren. Der Zeitaufwand nimmt zu, je weniger Bestandteile der Sicherung von dem gesicherten Rechner vorhanden sind.

Bei allen Methoden ist zu beachten, dass sie lediglich mit 3 Rechnern ausprobiert

und entwickelt worden sind. Es sollte zwar in der Theorie keinen Unterschied machen, welches Betriebssystem vorhanden ist, dennoch kann es sein, dass die Virtualisierung bei anderen Konfigurationen nicht funktioniert. Aus diesem Grund sollte in der Zukunft mit anderen Zusammensetzungen die verschiedenen Methoden überprüft werden.

4.4 Datenintegrität

Jede Methode scheint am Ende den Benutzer des gesicherten Rechners widerzuspiegeln. Werden die Startbildschirme der einzelnen Nutzer de VM und des gesicherten Rechners angeschaut, so sehen sie gleich aus. Auch beim Vergleich der Strukturen im Finder sieht die VM wie der gesicherte Rechner aus. Dennoch ist dies kein eindeutiger Beweis dafür, dass die Systeme wirklich gleich sind. Es könnten einzelne Dateien trotzdem nicht vorhanden sein und es ist nicht sicher, ob bei den Kopiervorgängen nicht doch eine Datei nicht mit kopiert oder verändert wurde.
Ein wichtiger Punkt in der IT-Forensik ist die Gewährleistung der Datenintegrität. Aus diesem Grund wurden die Hashwerte der Dateien des Containers, des Volumes und der Ordner bestimmt und anschließend verglichen.

Der Nachteil an dieser Methode ist, dass das Berechnen des Hashwertes sehr lange dauert, da von jeder einzelnen Datei der Hashwert berechnet und anschließend von diesen Hashwerten der übergeordnete Hashwert berechnen werden muss. Dies kann bei großen Volumes und APFS-Containern sehr lange dauern kann.

Tabelle 4.2: Hashwerte der Dateien des Images und der Dateien nach der Virtualisierung und den Anteil an veränderten Dateien in Prozent.

Methode	**Sicherung**	**Hashwert vor**	**Hashwert nach**	**Anteil veränderte Dateien**
2	APFS-Container (ohne T2)	e5ab0198fc43bb0a340e6718139f48a	dfdadfabb73d0ef7152f7fcd33db382	0,0003%
4	Daten-Volume	31aad3f3b8a8a42386b828ef816278e1	90814a19b2cbd9a3a1096cdf996eaf89	20%
5	Ordner Users	4395740b19ed7331d3755adcc35d7325	2a2959317839053509b0a11330ad274b	4 %
5	Ordner Applications	7b8d2233ee432706d2a7244374a2962b	a50d7aac24c23711a1622519a69eb975	0,03 %
5	Ordner System	2f73938f4faa4366ffd03178df33fa19	6f6ac484f49f0406a74ee38e7a001365	9 %

In Tabelle 4.2 werden die Hashwerte der unterschiedlichen Sicherungen von vor und nach der Virtualisierung dargestellt. Dazu wird der Anteil der veränderten Dateien in Prozent angeben.

Beim Vergleichen der Hashwerte fällt auf, dass kein Hashwert der Sicherung vor und nach der Virtualisierung übereinstimmt. Somit ist bei keinem die vollständige Integrität der Daten gewährleistet. Im Folgenden werden deshalb die einzelnen Dateien genauer betrachtet, die nicht übereinstimmen. In Tabelle 4.3 werden die Dateien und Ordner aufgelistet, die bei der Virtualisierung der einzelnen Sicherungen verändert wurden.

Die Hashwerte des APFS-Containers ohne T2-Chip wurden vor und nach der Wiederherstellung bestimmt. Dabei wurde das System noch nicht gestartet. Betrachtet man die Anzahl der Daten, die nicht übereinstimmen, so liegt diese bei ca. 0,0003 %. Bei diesen Dateien handelt es sich um Dateien aus dem versteckten Ordner `fseventsd`. Dieser Ordner beinhaltet Änderungen von Dateien und Ordnern, die auf einer Speicherfestplatte des Macs auftreten [7]. Dadurch, dass die Dateien sich auf einer neuen Festplatte befinden, ändern sich somit manche Einträge in dieser Datei. Dieses hat jedoch keinen Einfluss auf die Nutzung der Virtualisierung.

Tabelle 4.3: In der Tabelle werden die veränderten Dateien und Ordner der einzelnen Sicherungen dargestellt

Methode	**Sicherung**	**veränderte Ordner / Dateien**
2	APFS-Container (ohne T2)	.fseventsd
4	Daten-Volume	.DokumentRevision-V100 .Spotlight-V100 .fseventd /Library/ /Users/<Nutzer>/Library /private/etc /private/var /usr/libexec
5	Ordner Users	.Spotlight-V10 .fseventsd /Library
5	Ordner Applcations	.Spotlight-V100 .DS_Store
5	Ordner System	.Spotlicht-V100 .fseventsd /Library

Bei der Virtualisierung des Daten-Volumes wurden deutlich mehr Dateien verändert. Insgesamt stimmen rund 20 % der Dateien nicht überein. Es werden zwei Dateien aus dem versteckten Ordner `.DokumentRevision-V100` verändert. Der Ordner beinhaltet Dateiänderungen [23]. Bei den Dateien handelt es sich um `db-V1/db.sqlite-wal` und `metadata`. Durch die Veränderung dieser Dateien kann der Verlauf der Dateiänderungen an den Dateien nicht mehr korrekt sein [23]. Dies beeinflusst jedoch nicht die aktuelle Datei, da deren Informationen woanders abgespeichert wurden. Des Weiteren wird die versteckte Datei `.Spotlight-V100` und Dateien aus dem versteckten Ordner `.fseventd` verändert. Die Datei `.Spotlight-V100` beinhaltet die Konfigurationsdateien und den Spotlight Index des Volumes und `.fseventsd` überwacht das Dateisystem und teilt Veränderungen von Ordnern und Programmen mit [24].
Außerdem werden Dateien aus den Ordnern `/Library/` und aus dem Userordner `/Users/<Nutzer>/Library/` verändert. Der Ordner `Library` enthält Schriften und andere Objekte, die von Apps verwendet werden [29]. Des Weiteren werden Dateien des Ordners `/private/etc` und `/private/var` verändert und Dateien des Ordners `/usr/libexec` verändert.
Bei genauerer Betrachtung der veränderten Dateien fällt auf, dass bei allen Dateien, die verändert werden, es sich um Dateien handelt, die für das Betriebssystem relevant sind, aber keine Informationen über den User und seine aktuellen Dateien beinhalten. Dennoch gehen die Informationen verloren, welche Dateiänderungen vorgenommen wurden, da `.DokumentRevision-V100`, `.Spotlight-V100` und `.fseventd` verändert werden. Somit kann lediglich der aktuelle Stand der Dateien betrachtet werden, jedoch nicht frühere Versionen. Dazu kann der Aufbau der Apps anders sein, da Dateien aus dem Ordner `Library` verändert wurden.

Auch bei der Virtualisierung mithilfe der einzelnen Ordner, werden Dateien aus den Ordnern verändert. Bei dem Ordner Users stimmen ca. 4 % der Dateien nicht überein. Bei den Dateien handelt es sich ebenfalls um die versteckte Datei `.Spotlight-V100` und den versteckten Ordner `.fseventsd` und um Dateien aus dem Ordner `Library`. Bei dem Ordner Applications stimmen 0,03 % der Dateien nicht überein. Es wurden ebenfalls Dateien aus dem versteckten Ordner `.Spotlight-V100` und dazu noch die versteckte Datei `.DS_Store` verändert. Die Datei `.DS_Store` enthält Metadaten zu Dateien, Ordnern und Festplatten [19]. Dazu gehören Eigenschaften wie Listen- oder Symbolansicht, Symbol- und Schriftgröße des Finders [19]. Aus dem Ordner System stimmen ca. 9 % der Dateien nicht überein. Bei den Dateien handelt es sich ebenfalls um `.Spotlicht-V100` und `.fseventsd` Dateien, sowie Dateien aus dem Ordner `Library`.
Somit wurde bei den Ordnern genauso wie beim Daten-Volume lediglich für das System relevante Dateien verändert, aber keine Dateien vom Nutzer.

Die Datenintegrität ist bei keiner Virtualisierung zu 100 % gewährleistet. Bei den veränderten Dateien handelt es sich aber um Dateien, die benötigt werden, damit das System ausgeführt werden kann. Einige der Dateien werden bei der Virtualisierung

vermutlich ebenfalls weiter verändert, genauso wie bei einem Live-System Dateien des Systems verändert werden. Wichtig bei der Datensichtung in der VM ist deshalb, dass jeder einzelne Schritt dokumentiert wird, um die Veränderungen nachvollziehen zu können. Es gelten dieselben Regeln wie beim Untersuchen eines Live-Systems. Dazu sollte nicht direkt die Datensicherung verwendet werden, sondern eine Kopie davon, um das original Image nicht zu verändern.

4.5 Vergleich der durchgeführten Virtualisierung mit anderen Datenaufbereitungsprogrammen

Im Bereich der IT-Forensik gibt es viele unterschiedliche Programme, die das Ziel haben, die Daten von gesicherten Rechnern aufzubereiten. In den meisten Fällen werden dafür die Daten mithilfe von Parsing und Carving aus dem Image extrahiert und dann zur Verfügung gestellt. Sie bieten zur Sichtung der Daten unterschiedliche Ansichten und Filter an, um die Daten nach den gesuchten Informationen zu durchsuchen. Manche Programme bieten die Möglichkeit an, im beschränkten Rahmen bootfähige Images zu virtualisieren [43] [17].

Vergleicht man die hier entwickelten Methoden der Virtualisierung mit denen Methoden von anderer Software zur Datenaufbereitung, so haben beide Vor- und Nachteile. Betrachtet man die Bedienfreundlichkeit der unterschiedlichen Methoden, so ist die Virtualisierung ohne komplizierte Erklärungen zu bedienen, da sie sich genauso, wie ein Apple Rechner, verhält und man dementsprechend die Daten wie auf einem normalen Gerät sichten kann. Im Gegensatz dazu ist das Bedienen von anderer Software wie X-Ways Forensics [66] oder Axiom Forensics [62] deutlich komplizierter. Durch die verschiedenen Ansichten und Filter benötigt es gewisse Einarbeitungszeit und erfolgt in den meisten Fällen nicht intuitiv.

Die kompliziertere Bedienung folgt aus den verschiedenen Möglichkeiten, die die Programme bieten, um die Daten nach Informationen zu durchsuchen. So können Abgleiche mit Hashsets durchgeführt werden oder Stichwortlisten zur Durchsuchung genutzt werden [38] [66]. Dies ist in der Virtualisierung nicht möglich. Dort kann zum Suchen lediglich die Betriebssystem eigenen Suchmethoden verwendet werden.
In den meisten Fällen bieten die vorgestellten Programme die Möglichkeit, in einer Baumstruktur die einzelnen Ordner und Dateien zu sichten [38] [66]. Dies ist nicht so übersichtlich, wie in den Apple eigenen Dateistrukturdarstellungen, ermöglicht aber dennoch die Ordnerstruktur nachzuvollziehen.

Ein Nachteil an der Virtualisierung ist, dass durch die Sichtung der Daten Dateien verändert werden, genauso wie bei einem Live-System. Dennoch werden Daten wie Dokumente, Bilder und Ordnerstrukturen nicht ohne aktives Eingreifen verändert und können dementsprechend ohne Folgen dafür gesichtet werden. Vor allem, wenn

es sich bei der Sicherung für die Virtualisierung um keinen APFS-Container handelt, muss einem bewusst sein, welche Daten von dem gesicherten Rechner und welche aus der VM der Virtualisierung stammen. Das Problem der Veränderung der Daten besteht bei anderen Programmen nicht, da dort nicht aktiv auf das System zugegriffen wird [38] [66]. Dennoch werden von den einzelnen Dateien Hashwerte gebildet, um die Datenintegrität zu gewährleisten [38] [66]. Auch bei der Sichtung der Daten in der Virtualisierung, sollte ebenfalls vor und nach der Sichtung die Hashwerte der Dateien berechnet werden, um zu überprüfen, welche Dateien verändert wurden.

Der große Vorteil der Virtualisierung gegenüber den anderen Programmen ist, dass durch die Virtualisierung ein Eindruck vom Nutzer des Rechners erlangt werden kann, da man explizit das sieht, was der Nutzer sieht. Zusammen mit der intuitiven Bedienung kann dadurch schnell und einfach eine Sichtung erfolgen und eine Einschätzung durchgeführt werden. Aufgrund dessen, dass die Anwendungen in der VM ebenfalls vorhanden sind, können die Daten von bestimmten Programmen, wie E-Mails oder anderen Nachrichten, mit dem korrekten Programm gesichtet werden und stellen kein Problem bei der Sichtung dar, wie in den anderen Programmen. Dadurch kann es möglich sein, bestimmte Beziehungen zwischen bestimmten Dateien besser nachzuvollziehen zu können. Dies führt dazu, dass auch Personen mit weniger Computerwissen leichter Verknüpfungen schließen können und nachvollziehen können, woher welche Datei stammt und was sie beinhaltet. Dennoch muss einem bewusst sein, dass durch die Sichtung keine gelöschten Daten, die sich nicht mehr im Papierkorb befinden, gefunden werden können, was mit anderen Programmen möglich ist [38] [66].

Da die Virtualisierung gegenüber anderer forensischer Software sowohl starke Vor- aber auch Nachteile besitzt, sollte sie in Kombination mit anderer Software verwendet werden, um das bestmögliche Ergebnis zu erhalten. Die Virtualisierung kann genutzt werden, um sich schnell einen Überblick und ersten Eindruck zu verschaffen, um anschließend mithilfe von anderer Software, falls notwendig, eine tiefgehendere Untersuchung durchzuführen. Insbesondere hilft die Virtualisierung bestimmte Zusammenhänge besser zu verstehen.

Der Vorteil von in dieser Arbeit erstellen Methoden gegenüber anderen Programmen zur Virtualisierung von Images ist, dass sie kostenlos sind und auch nicht bootfähige Images virtualisieren können. Dazu können sie APFS und macOS bis inklusive der Version Monterey virtualisieren.

5 Fazit und Ausblick

In dieser Arbeit wurde sich mit der Virtualisierung von gesicherten Apple System mithilfe von VirtualBox befasst. Dabei wurde sich auf die Virtualisierung von Apple-Computern ab dem Betriebssystem Mojave bis Ventura festgelegt, die das Dateisystem APFS besitzen.
Bei der Virtualisierung von gesicherten Apple-Systemen stellen die Sicherheitsfeatures, vor allem die Sicherheitschips, ein Problem dar. Es ist möglich den APFS-Container eines Intel-Rechners ohne Sicherheitschip mithilfe von VirtualBox zu virtualisieren. Bei APFS-Containern von Rechnern mit Sicherheitschip ist die Virtualisierung nicht mehr möglich. Dies liegt daran, dass das Passwort zum Entsperren von FileVault vom Sicherheitschip abhängig ist und ohne diesen kann FileVault die Daten nicht entschlüsseln. Auch die entschlüsselte Datensicherung mit DigitalCollector kann nicht virtualisiert werden, da FileVault angeblich aktiviert ist, es aber keine Nutzer gibt, die das Recht haben das Volume zu entschlüsseln. Zur Virtualisierung des APFS-Containers wird lediglich das Festplattendienstprogramm der VM benötigt.
Bei Rechnern mit T2-Chip besteht die Möglichkeit bis Monterey ein Image vom Daten-Volume zu erstellen. Dieses kann ebenfalls virtualisiert werden, indem auf dem Daten-Volume das Betriebssystem neu installiert wird.
Ab Monterey können nur noch Images von den einzelnen Ordnern des Daten-Volumes durchgeführt werden. Mithilfe von diesen kann der gesicherte Rechner ebenfalls virtualisiert werden.
Zu beachten ist jedoch bei der Virtualisierung der gesicherten Apple-Systeme, dass die Datenintegrität nicht für alle Daten gewährleistet werden kann. Bei der Virtualisierung werden Dateien verändert, die das System benötigt. Es werden aber keine Daten des Nutzers verändert. Trotzdem gehen immer mehr Informationen des gesicherten Systems verloren, je weniger Daten zur Virtualisierung vorhanden ist.
Aus diesem Grund sollte die Virtualisierung in Kombination mit andern Datenaufbereitungsprogrammen verwendet werden, da es einen ersten Eindruck der Dateien auf dem gesicherten System übermittelt.

In Zukunft sollten die im Rahmen dieser Arbeit Methoden mit anderen Konfigurationen überprüft und sich tiefgehender mit dem Datenverlust auseinandergesetzt werden. Dazu sollte eine Methode entwickelt werden, die es ermöglicht die Daten aus dem APFS-Container, der als AFF4 Image vorliegt, zu extrahieren. Dies würde dazu führen, dass Sicherungen von APFS-Containern von Rechnern mit T2, M1 oder M2-Chip mit und ohne aktivem FileVault ebenfalls virtualisiert werden können.

Anhang A: Befehle und Einstellungen VM erstellen

Abbildung A.1: Einstellungen, um eine neue VM mit einem macOS Gastsystem zu erstellen. (Quelle: Eigene Darstellung)

Der Hauptspeicher sollte mindestens 8192 GB betragen.

Konfigurationsbefehle für die Apple-VM:

```
1 cd  /Applications/VirtualBox.app/Contents/
2 VBoxManage modifyVM "<VM-Name>" --cpuidset 00000001 000106e5
  ↪  00100800 0098e3fd bfebfbff
3 VBoxManage setextradata "<VM-Name>"
  ↪  "VBoxInternal/Devices/efi/0/Config/DmiSystemProduct" "iMac11,3"
4 VBoxManage setextradata "<VM-Name>"
  ↪  "VBoxInternal/Devices/efi/0/Config/DmiSystemVersion" "1.0"
5 VBoxManage setextradata "<VM-Name>"
  ↪  "VBoxInternal/Devices/efi/0/Config/DmiBoardProduct"
  ↪  "Iloveapple"
6 VBoxManage setextradata "<VM-Name>"
  ↪  "VBoxInternal/Devices/smc/0/Config/DeviceKey"
  ↪  "ourhardworkbythesewordsguardedpleasedontsteal(c)AppleComputerInc"
7 VBoxManage setextradata "<VM-Name>"
  ↪  "VBoxInternal/Devices/smc/0/Config/GetKeyFromRealSMC" 1
```

An der Stelle des VM-Namens wird der Name der VM eingetragen, die konfiguriert werden soll.

Bootstick VMDK und VDI-Datei erstellen:

```
1 cd  /Applications/VirtualBox.app/Contents/
2 sudo VBoxManage createmedium disk --filename=<pfad Datei
  ↪  speichern>/<Dateiname> --variant=RawDisk --format=vmdk
  ↪ --property RawDrive=/dev/<USB-Stick>
3 sudo VboxManage clonehd --format Vdi <Pfad>/<Dateiname.vmdk
  ↪  <Pfad>/<Dateiname.vdi>
```

```
1 sudo chmod 070 /Users/nele/VirtualBox\ VMs/<Dateiname>.vdi
```

Anhang B: Pythonskript zum Hashwert-Abgleich

```
import fileinput
import os
from pdb import line_prefix
from re import L
from sys import argv
import threading
import multiprocessing
import time
import hashlib

def main():
 #Erstellen der Listen für die Hashwerte und Dateinamen aus den zwei Txt-Dateien mit Hashwerten
 hashwerte=[]
 hashwerte_2=[]
 Threads=[]
 hash1=[]
 hash2 =[]
 text=[]
 zp=" "
 line_break = False
 prozesse=8
 #Überprüfen der korekten Eingabe
 if len(argv) != 3:
  print("invalid input")
 elif not (os.path.exists(argv[1])&os.path.exists(argv[2])):
  print("invalid argument")
 #Einlesen der einzelnen Hashwerte-Ausgaben und extrahieren des Dateipfades und des Hashwertes
 else:
  with open(argv[1]) as file1:
   Lines=file1.readlines()
   for line in Lines:
    if line == " ":
      continue
    if ('=' in line) == False:
     zp=line.rstrip(line[-1])
```

```
36      line_break = True
37      continue
38     if line_break:
39      line=zp+line
40     text= line.rsplit("=",1)
41     hash=text[1]
42     hash=hash.split()
43     hash=hash[0]
44     text=text[0].split("MD5")
45     pfad=text[1]
46     pfad=pfad[2:-2]
47     hashwerte.append([pfad,hash])
48     line_break=False
49
50   with open(argv[2]) as file2:
51    Lines=file2.readlines()
52    for line in Lines:
53     if line == " ":
54       continue
55     if ('=' in line) == False:
56      zp=line.rstrip(line[-1])
57      line_break = True
58      continue
59     if line_break:
60      line=zp+line
61     text= line.rsplit("=",1)
62     hash=text[1]
63     hash=hash.split()
64     hash=hash[0]
65     text=text[0].split("MD5")
66     pfad=text[1]
67     pfad=pfad[2:-2]
68     hashwerte_2.append([pfad,hash])
69     line_break=False
70
71 #Sortieren der Werte mit Hilfe des Mergesorts
72   pool =multiprocessing.Pool(processes=2)
73   t1= pool.apply_async(mergeSort,(hashwerte,))
74   t2= pool.apply_async(mergeSort,(hashwerte_2,))
75   pool.close()
76   pool.join()
77   hashwerte=t1.get()
78   hashwerte_2=t2.get()
```

```
79
80  #Hashwert über alle Hashwerte
81    pool =multiprocessing.Pool(processes=2)
82    t1= pool.apply_async(md5_liste,(hashwerte,))
83    t2= pool.apply_async(md5_liste,(hashwerte_2,))
84    pool.close()
85    pool.join()
86    print("Der Hashwert des 1. Hashsets beträgt "+t1.get())
87    print("Der Hashwert des 2. Hashsets beträgt "+t2.get())
88  #Abbruch, wenn Werte gleich sind
89    if t1.get() != t2.get():
90     print("Vergleiche Hashwerte...")
91
92     if len(hashwerte)>=len(hashwerte_2):
93     #Vergelichen der Hashwerte, wenn Liste 2 länger als Liste 1
94      pool =multiprocessing.Pool(processes=prozesse)
95      count = 0
96      for i in range(prozesse):
97       hash1.append(hashwerte[i*len(hashwerte)//prozesse:
          ↪  (i+1)*len(hashwerte)//prozesse])
98       hashvalue1, hashvalue2 =hash2_aufteilen(hash1[i],hashwerte_2)
99       hash2.append(hashvalue2)
100      hashwerte_2=hashvalue2
101      hash2[i]=hashvalue1
102      Threads.append(pool.apply_async(vergleichen,
          ↪  (hash1[i],hash2[i],)))
103    #warten bis Vergleiche fertig sind
104     pool.close()
105     pool.join()
106
107    #Schreiben der nicht korrekten Dateien sortiert in eine
       ↪  Textdatei
108     for i in range(prozesse):
109      thread=Threads[i]
110      for item in thread.get()[1]:
111        text.append(item)
112      count=count+thread.get()[0]
113     text.sort()
114
115     print("Schreibe veränderte Dateien in Textdatei...")
116     with open ("Desktop/ver_Dateien.txt", "w+") as text_file:
117      for item in text:
118       text_file.write(item+"\n")
```

```
119      text_file.write("\n"+"Es stimmen " + str(count)+" von " +
          ↪  str(len(hashwerte)) + " aus dem 1. Hashset überein"+"\n")
120     print("Es stimmen " + str(count)+" von " + str(len(hashwerte))
         ↪  + " aus dem 1. Hashset überein")
121
122    else:
123    #Vergelichen der Hashwerte, wenn Liste 2 länger als Liste 1
124     pool =multiprocessing.Pool(processes=prozesse)
125     count = 0
126     for i in range(prozesse):
127      hash1.append(hashwerte_2[i*len(hashwerte_2)//prozesse:
          ↪  (i+1)*len(hashwerte_2)//prozesse])
128      hashvalue1, hashvalue2 =hash2_aufteilen(hash1[i],hashwerte)
129      hash2.append(hashvalue2)
130      hashwerte=hash2[i]
131      hash2[i]=hashvalue1
132      Threads.append(pool.apply_async(vergleichen,
          ↪  (hash1[i],hash2[i],)))
133    #warten bis Vergleiche fertig sind
134     pool.close()
135     pool.join()
136
137   #Schreiben der nicht korrekten Dateien sortiert in eine Textdatei
138     for i in range(prozesse):
139      thread=Threads[i]
140      for item in thread.get()[1]:
141        text.append(item)
142      count=count+thread.get()[0]
143     text.sort()
144
145     print("Schreibe veränderte Dateien in Textdatei...")
146     with open ("Desktop/ver_Dateien.txt", "w+") as text_file:
147      for item in text:
148       text_file.write(item+"\n")
149      text_file.write("\n"+"Es stimmen " + str(count)+" von " +
          ↪  str(len(hashwerte_2)) + " aus dem 2. Hashset
          ↪  überein"+"\n")
150     print("Es stimmen " + str(count)+" von " +
         ↪  str(len(hashwerte_2)) + " aus dem 2. Hashset überein")
151
152 #Sortieren der Listen mit Hilfe des Mergesorts
153 def mergeSort(list):
154  if len(list) > 1:
```

```
155   mid=len(list)//2
156   left = list[:mid]
157   right = list[mid:]
158
159   mergeSort(left)
160   mergeSort(right)
161   i=0
162   j=0
163   k=0
164
165   while i < len(left) and j < len(right):
166    if left[i][1] <= right[j][1]:
167     list[k] = left[i]
168     i=i+1
169    else:
170     list[k] = right[j]
171     j=j+1
172    k=k+1
173   while i<len(left):
174    list[k] = left[i]
175    i=i+1
176    k=k+1
177   while j<len(right):
178    list[k] = right[j]
179    j =j+1
180    k=k+1
181   return list
182
183 #Vergleichen der Hashwerte der längeren Datei mit der der kürzeren
     ↪  Datei und Ausgeben des Dateinamens und des Hashwertes, welche
     ↪  keinen passenden Hashwert in der anderen Datei haben
184 def vergleichen(hashliste1, hashliste2):
185   liste=[]
186   stop=False
187   count=0
188   not_removed=True
189   for hashdatei in hashliste1:
190    for hashdatei2 in hashliste2:
191     if hashdatei[1] == hashdatei2[1]:
192      stop=True
193      count=count+1
194      del hashdatei2
195      break
```

```
196     if stop == False:
197      datei=hashdatei[0]
198      Hash=hashdatei[1]
199      liste.append("Die Datei "+str(datei)+" mit den Hash "+ Hash+ "
          ↪  wurde verändert")
200     stop=False
201    return count, liste
202
203  #Auftrennen, der 2. Hashliste in Teillisten
204  def hash2_aufteilen(hashliste1,hashliste2):
205   gefunden=False
206   for hashdatei in reversed(hashliste1):
207    for hashdatei2 in hashliste2:
208      if hashdatei[1] == hashdatei2[1]:
209          index=hashliste1.index(hashdatei)
210          counter=0
211          while hashliste1[hashliste1.index(hashdatei)][1] ==
              ↪  hashliste1[index][1]:
212           index=index-1
213           counter=counter+1
214          hashliste=hashliste2[:hashliste2.index(hashdatei2)+
              ↪  counter]
215          hashliste2=hashliste2[hashliste2.index(hashdatei2)+
              ↪  counter:]
216          gefunden=True
217          break
218      if gefunden:
219          break
220    if gefunden:
221      break
222   return hashliste, hashliste2
223
224  def md5_liste(liste):
225   hashliste=""
226   for hash in liste:
227    hashliste=hashliste+str(liste[1])
228   return hashlib.md5(hashliste.encode('utf-8')).hexdigest()
229
230  if __name__ == "__main__" :
231   main()
232
```

Literaturverzeichnis

[1] M. Adler. „Zlib“, GitHub. (20. Aug. 2023), Adresse: https://github.com/madler/zlib (besucht am 19. 08. 2023).

[2] „AFF4 -The Advanced Forensics File Format“, Aff4. (), Adresse: https://www2.aff4.org/ (besucht am 19. 09. 2023).

[3] „Apple stellt den M1 vor“, Apple. (10. Nov. 2020), Adresse: https://www.apple.com/de/newsroom/2020/11/apple-unleashes-m1/ (besucht am 12. 07. 2023).

[4] „Apple stellt den M2 Chip vor, der die bahnbrechende Performance und Funktionalität des M1 weiter voranbringt“. (6. Juni 2022), Adresse: https://www.apple.com/de/newsroom/2022/06/apple-unveils-m2-with-breakthrough-performance-and-capabilities/.

[5] *Apple T2 Security Chip Security Overview*, Okt. 2018. Adresse: https://www.apple.com/mideast/mac/docs/Apple_T2_Security_Chip_Overview.pdf (besucht am 28. 08. 2023).

[6] „Apple veröffentlicht Vorschau auf Mac OS X Lion für Entwickler“, Apple. (24. Feb. 2011), Adresse: https://www.apple.com/de/newsroom/2011/02/24Apple-Releases-Developer-Preview-of-Mac-OS-X-Lion/ (besucht am 12. 07. 2023).

[7] D. Balaban. „Wie man hohe CPU- und Speicherauslastung durch fseventsd auf dem Mac behebt“, macsecurity. (20. Aug. 2023), Adresse: https://macsecurity.net/de/view/576-fseventsd-mac-high-cpu-and-memory (besucht am 23. 08. 2023).

[8] O. Choudary, F. Grobert und J. Metz, „Infiltrate the Vault: Security Analysis and Decryption of Lion Full Disk Encryption“, 2012. Adresse: https://eprint.iacr.org/2012/374.pdf.

[9] H. T. Co, „Virtualization Technology“, in *Cloud Computing Technology*, 2023, S. 97–110, ISBN: 978-981-19-3025-6.

[10] M. Cohen. „The AFF4 Imager“, Aff4 Imager. (), Adresse: https://docs.aff4.org/en/latest/# (besucht am 19. 09. 2023).

[11] Y. Collet. „Lz4“, GitHub. (15. Sep. 2023), Adresse: https://github.com/lz4/lz4 (besucht am 19. 09. 2023).

[12] P. Craiger und P. Burke, „mac OS X Forensics“, in *Advances in digital Forensics II*, 2006, S. 159–170.

[13] „Die führende Computerdaten-Analyselösung für Windows und Mac“, Cellebrite. (), Adresse: https://cellebrite.com/de/cellebrite-inspector/ (besucht am 19. 09. 2023).

[14] F. Douglis und O. Krieger, *Virtualization*, Apr. 2013. Adresse: https://ieeexplore.ieee.org/abstract/document/6488669.

[15] „Erstellen einer Image-Datei im Festplattendienstprogramm auf dem Mac“, apple. (), Adresse: https://support.apple.com/de-de/guide/disk-utility/dskutl11888/19.0/mac/10.15 (besucht am 14. 08. 2023).

[16] „Festplattendienstprogramm auf dem Mac“, Apple. (), Adresse: https://support.apple.com/de-de/guide/disk-utility/dskutl1029/19.0/mac/10.15 (besucht am 19. 09. 2023).

[17] *Forensic Expporer User Manuel*, 9. Juni 2023.

[18] M. Gerl. „Disk-Arbitrator“, GitHub. (8. Feb. 2020), Adresse: https://github.com/aburgh/Disk-Arbitrator (besucht am 19. 09. 2023).

[19] B. Gruber. „.DS_Store: So funktioniert der Gedächnisbaustein von macOS“, Macwelt. (29. Dez. 2018), Adresse: https://www.macwelt.de/article/966501/ds-store-osx-finder-terminal.html (besucht am 23. 08. 2023).

[20] „hdiutil“, ss6. (), Adresse: https://ss64.com/osx/hdiutil.html (besucht am 31. 07. 2023).

[21] N. Hery-Moßmann. „macOS Versionen: Alle Betriebssysteme im Überblick“, Chip. (29. Nov. 2022), Adresse: https://praxistipps.chip.de/macos-versionen-alle-betriebssysteme-im-ueberblick_48546 (besucht am 11. 08. 2023).

[22] „Hinzufügen oder Löschen von APFS-Volumes mithilfe des Festplattendienstprogramms auf dem Mac“, Apple. (), Adresse: https://support.apple.com/de-de/guide/disk-utility/dskua9e6a110/21.0/mac/12.0 (besucht am 18. 07. 2023).

[23] hoakley. „Document Versioning“, eclecticlight. (19. Feb. 2018), Adresse: https://eclecticlight.co/2018/02/19/document-versioning/ (besucht am 23. 08. 2023).

[24] hoakley. „Hidden files and folders on APFS volumes“, eclecticlight. (21. Sep. 2021), Adresse: https://eclecticlight.co/2021/12/09/hidden-files-and-folders-on-apfs-volumes/ (besucht am 23. 08. 2023).

[25] „iMac Pro, der leistungsstärkste Mac aller Zeiten ist ab heute erhältlich“, Apple. (14. Dez. 2017), Adresse: https://www.apple.com/de/newsroom/2017/12/imac-pro-the-most-powerful-mac-ever-available-today/ (besucht am 12. 07. 2023).

[26] „Informationen zum schreibgeschützten Systemvolume in macOS Catalina“, Apple. (24. Okt. 2022), Adresse: https://support.apple.com/de-de/HT210650 (besucht am 18. 07. 2023).

[27] „Informationen zum Startsicherheitsdienstprogramm auf einem Mac mit dem Apple T2 Security Chip“, Apple. (), Adresse: https://support.apple.com/de-de/HT208198 (besucht am 19. 09. 2023).

[28] „Inspecting AFF4 volumes“, winpmem. (), Adresse: https://winpmem.velocidex.com/docs/inspecting/ (besucht am 23. 08. 2023).

[29] „Integrierte Ordner des Mac“, Apple. (), Adresse: https://support.apple.com/de-de/guide/mac-help/mchlp1143/10.15/mac/10.15 (besucht am 23. 08. 2023).

[30] L. Klein. „Mac Papierkorb finden, anzeigen und sicher entleeren“, macube. (6. Mai 2022), Adresse: https://www.macube.com/de/how-to/securely-clean-up-trash-mac.html#:~:text=Im%20Finder%20befindet%20sich%20der,%C3%96ffnen%20Sie%20den%20Apple%20Papierkorb (besucht am 23. 08. 2023).

[31] *Laptops: Apple (MacBook) owners in Germany*, Aug. 2023. Adresse: https://www.statista.com/study/93335/laptops-apple-macbook-owners-in-germany/ (besucht am 22. 09. 2023).

[32] *Leitfaden „IT-Forensik“*, Bonn, März 2011. Adresse: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf?__blob=publicationFile&v=1 (besucht am 19. 09. 2023).

[33] „Live View“, SourceForge. (), Adresse: https://liveview.sourceforge.net/ (besucht am 14. 08. 2023).

[34] „macOS 13: Ventura“, GitHub. (11. Juli 2023), Adresse: https://dortania.github.io/OpenCore-Install-Guide/extras/ventura.html#table-of-contents (besucht am 31. 07. 2023).

[35] „macOS High Sierra liefert verbesserte Technologien für Speicher, Video und Grafik“, Apple. (5. Juni 2017), Adresse: https://www.apple.com/de/newsroom/2017/06/macos-high-sierra-delivers-advanced-technologies-for-storage-video-and-graphics/ (besucht am 18. 07. 2023).

[36] „macOS Ventura bringt leistungsstarke Produktivitätstools und neue Features zur Zusammenarbeit, die das Mac Erlebnis besser machen denn je“, Apple. (6. Juni 2022), Adresse: https://www.apple.com/de/newsroom/2022/06/macos-ventura-brings-powerful-productivity-tools-new-continuity-features-to-mac/ (besucht am 31. 07. 2022).

[37] „Macs mit Apple-Prozessoren: Neuer Boot-Manager, Ersatz für Target-Disk-Modus, Booten von externen Volumes weiter möglich“, Mac Gadget. (25. Juni 2020), Adresse: https://www.macgadget.de/News/2020/06/25/Macs-mit-Apple-Prozessoren-Neuer-Boot-Manager-Ersatz-fuer-Target-Disk-Modus-Booten (besucht am 08. 08. 2023).

[38] *Magnet Axiom User Guide*, 7. Juli 2023.

[39] K. Mastwijk. „ewfacquire(1) - Linux man page“, die.net. (1. Sep. 2010), Adresse: https://linux.die.net/man/1/ewfacquire (besucht am 08. 08. 2023).

[40] „Merge Sort – Data Structure and Algorithms Tutorials“, geeksforgeeks. (), Adresse: https://www.geeksforgeeks.org/merge-sort/ (besucht am 19. 09. 2023).

[41] „Nutzen Sie Windows auf dem Mac. Einfach. Leistungsstark. Perfekt integriert.“, Parallels. (), Adresse: https://www.parallels.com/de/products/desktop/ (besucht am 19. 09. 2023).

[42] R. O'Grady. „Snappy“, GitHub. (12. Juli 2023), Adresse: https://github.com/google/snappy (besucht am 19. 09. 2023).

[43] *OSForensics*, Apr. 2023. Adresse: https://www.osforensics.com/downloads/OSF_help.pdf#search=%22booting%20a%20forensic%20image%22.

[44] „Overview-winpmem memory imager“, velocidex. (), Adresse: https://winpmem.velocidex.com/docs/usage/ (besucht am 19. 09. 2023).

[45] *Product Overview: Cellebrite Digital Collector*. Adresse: https://cellebrite.com/wp-content/uploads/2021/04/ProductOverview_DigitalCollector_LTR_2021_web.pdf (besucht am 14. 08. 2023).

[46] N. Reddy, „Mac OS Forensics“, in *Practical Cyber Forensics*, 2019, S. 101–132.

[47] „Rolle von APFS (Apple File System)“, Apple. (13. Mai 2022), Adresse: https://support.apple.com/de-de/guide/security/seca6147599e/web (besucht am 18. 07. 2023).

[48] A. Schulz. „Mac: Geschichte und Entwicklung des beliebten Computers von Apple“, ingame. (28. Okt. 2022), Adresse: //www.ingame.de/hardware/mac-computer-macbook-apple-macintosh-91880762.html (besucht am 18. 07. 2023).

[49] *Sicherheit der Apple-Plattformen*, 2022. Adresse: https://help.apple.com/pdf/security/de_DE/apple-platform-security-guide-d.pdf.

[50] D. Sladovic, D. Topolcic und D. Delija, „Overwiev of Mac system security and its impact on digital forensics process“, **presentedat**MIPRO 2020, Opatija, Croatia, Okt. 2020. (besucht am 20. 08. 2023).

[51] D. Smith, *Apple macOS and iOS System Administration*. 2020, ISBN: 978-1-4842-5819-4.

[52] J. Stuettgen. „OSXPMem - Mac OS X Physical Memory acquisition tool“, Google Code. (), Adresse: https://code.google.com/archive/p/pmem/wikis/OSXPmem.wiki (besucht am 19. 09. 2023).

[53] „The Only Solution for Live and Targeted Computer Data Collection“, cellebrite. (), Adresse: https://enterprise.cellebrite.com/digital-collector/ (besucht am 14. 08. 2023).

[54] „Transformation für das Zeitalter digitaler Ermittlungen“, Cellebrite. (), Adresse: https://cellebrite.com/de/startseite/ (besucht am 19. 09. 2023).

[55] *User Manual Version 7.0.10*. Adresse: http://download.virtualbox.org/virtualbox/UserManual.pdf.

[56] „Verschlüsseln der Mac-Daten mit FileVault“, Apple. (), Adresse: https://support.apple.com/de-de/guide/mac-help/mh11785/mac (besucht am 12. 07. 2023).

[57] „Verwenden von macOS-Wiederherstellung auf einem Intel-basierten Mac“, Apple. (), Adresse: https://support.apple.com/de-de/guide/mac-help/mchl338cf9a8/11.0/mac/11.0 (besucht am 19. 09. 2023).

[58] „VMware Fusion“, vmware. (), Adresse: https://www.vmware.com/de/products/fusion.html (besucht am 19. 09. 2023).

[59] „VMware Virtual Disk Development Kit (VDDK)“, vmware developer. (), Adresse: https://developer.vmware.com/web/sdk/6.0/vddk (besucht am 19. 09. 2023).

[60] „VMware Workstation Pro“, vmware. (), Adresse: https://www.vmware.com/de/products/workstation-pro.html (besucht am 19. 09. 2023).

[61] „Welcome to VirtualBox.org!“, VirtualBox. (), Adresse: https://www.virtualbox.org/ (besucht am 13. 09. 2023).

[62] „Wiederherstellung und Analyse aller Beweismittel in einem Fall“, Magnet Forensics. (), Adresse: https://www.magnetforensics.com/de/products/magnet-axiom/ (besucht am 19. 09. 2023).

[63] A. Wong. „How Much Swap Memory Is Mac Using“, Iboysoft. (10. Juli 2023), Adresse: https://iboysoft.com/wiki/swap-memory-mac.html (besucht am 23. 08. 2023).

[64] „Working with APFS Volume Groups“, Carbon Copy Cloner. (20. Juli 2022), Adresse: https://bombich.com/kb/ccc6/working-apfs-volume-groups (besucht am 18. 07. 2023).

[65] „X-Ways Forensics: Integrierte Software für Computerforensik“, X-Ways. (23. Mai 2015), Adresse: https://www.x-ways.net/forensics/index-d.html (besucht am 19. 09. 2023).

[66] *X-Ways Forensics/ WinHex Manuel*, Juli 2023. Adresse: http://www.x-ways.net/winhex/manual.pdf.

[67] „Xmount“, It-Forensic Wiki. (), Adresse: https://it-forensik.fiw.hs-wismar.de/index.php/Xmount (besucht am 31. 07. 2023).

[68] L. Zhang, D. Zhang und L. Wang, „Live Digital Forensics in a Virtual Machine“, **presentedat**International Conference on Computer Application and System Modeling, 2010. (besucht am 23. 08. 2023).

Eidesstattliche Erklärung

Hiermit versichere ich – Nele Christin Schnaubelt – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 26. September 2023 [redacted]

Ort, Datum Nele Christin Schnaubelt