Hochschule Wismar
University of Applied Sciences
Technology, Business and Design
Fakultät für Ingenieurswissenschaften, Bereich EuI

IT-Forensik-Projekt II

„FORENSIK-SOFTWARE – TEST VON FUNKTIONALITÄTEN AUF DATEIIDENTIFIKATIONS- & DATEIREKONSTRUKTIONSEBENE IN NTFS & APFS-DATEISYSTEMEN“

(LAUT AUSGEGEBENER AUFGABENSTELLUNG)

Eingereicht am: 19.06.2022
Studiengang: IT-Forensik | Modul: IT-Forensik-Projekt II

Aufgabenstellung

Erstellung einer (ingenieur)wissenschaftlichen Aufarbeitung einer IT-Forensischen Fragestellung. Dabei liegt der Fokus auf zwei Aspekten:

1) Bearbeitung und Lösungsfindung für ein IT-Forensisches Problem
2) Wissenschaftliche Aufarbeitung in einer schriftlichen Arbeit (inkl. Vertraut machen mit den Formalitäten und Richtlinien und somit Vorbereitung auf die Bachelor Thesis)

Die Fragestellung soll aus der Praxis bzw. dem Tätigkeitsumfeld und/oder den Studieninhalten stammen.

Neben der schriftlichen Ausarbeitung ist ein 15-minütiger Vortrag zur IT-Forensischen Fragestellung Bestandteil der Prüfungsleistung. Im Optimalfall soll hier auch die Demonstration der Lösung des IT-Forensischen Problems enthalten sein.

Kurzreferat

Diese Arbeit beschäftigt sich mit der Evaluierung der grundlegenden Funktionen, die zur forensischen Aufarbeitung digitaler Spurenträger Einsatz findet. Diese Funktionen umfassen das Erkennen von Dateien anhand ihrer Signaturen, Auffinden und Identifizieren von versteckten und das Wiederherstellen gelöschter Dateien, sowie deren Dokumentation zu Beweiszwecken. Während die Dokumentationsfunktionen zuverlässig die aufbereiteten Inhalte der Programme wiedergaben, konnte jedoch festgestellt werden, dass keines der untersuchten Programme mit absoluter Zuverlässigkeit bei der Erstellung dieser Inhalte arbeitet. Die Spanne der hierbei dokumentierten Schwächen reicht von dem Nichterkennen des Verstecktstatus von Dateien bis hin zum fehlerhaften Inerpretieren von $MFT Einträgen.

Abstract

This thesis deals with the evaluation of the basic functions that are used for the forensic processing of digital evidence media. These functions include recognizing files by their signatures, finding and identifying hidden files, recovering deleted files, and documenting them for evidentiary purposes. While the documentation functions reliably reproduced the prepared content of the programs, it was found that none of the programs examined worked with absolute reliability in creating this content. The range of weaknesses documented here ranged from not recognizing the hidden status of files to incorrectly interpreting $MFT entries.

Inhaltsverzeichnis

Literaturverzeichnis

[1] Abcdef: Datei Format – File Format. Online im Internet, URL: https://de.abcdef.wiki/wiki/File_format#Magic_number, 29.09.2021, letzter Zugriff 17.06.2022

[2] Amr Amin: Inside NTFS: Discovering The Master File Table (MFT)– PART2. Online im Internet, URL: https://www.bluekaizen.org/inside-ntfs-discovering-the-master-file-table-mft-part2/, 31.12.2015, letzter Zugriff 17.06.2022

[3] Ashutosh Dixit, Know IT Like Pro: Understanding NTFS Architecture, MFT and Basics of NTFS forensics. Online im Internet, URL: https://knowitlikepro.com/understanding-ntfs-architecture-mft-and-basics-of-ntfs-forensics/, unbekannt, letzter Zugriff 17.06.2022

[4] Basis Technology: Autopsy User Documentation 4.19.3. URL: http://sleuthkit.org/autopsy/docs/user-docs/4.19.3/, 17.06.2022, letzter Zugriff 17.06.2022

[5] Brian Carrier: File System Forensic Analysis. 1. Auflage. Boston: Pearson Education, 2005

[6] Daniel Garrier: Understanding Deleted Files, Unallocated Space, and Their Impact on E-Discovery. Online im Internet, URL: https://www.thomsonreuters.com/en-us/posts/legal/understanding-e-discovery/, Stand 28.12.2017, letzter Abruf am 17.06.2022

[7] Dave Hull, SANS: NTFS: Attributes Part One. Online im Internet, URL: https://www.sans.org/blog/ntfs-attributes-part-one/, 24.12.2009, letzter Zugriff 17.06.2022

[8] Developer.Apple: Apple File System Reference, Online im Internet. URL: https://developer.apple.com/support/downloads/Apple-File-System-Reference.pdf, Stand 22.06.2020, letzter Zugriff 27.05.2022

[9] Dirk Labudde, Michael Spranger: Forensik in der digitalen Welt – Moderne Methoden der forensischen Fallarbeit in der digitalen und digitalisierten realen Welt. 1. Auflage, Berlin: Springer-Verlag, 2017

[10] Hartmut Ernst , Jochen Schmidt & Gerd Beneken: Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis – Eine umfassende, praxisorientierte Einführung. 7., erweiterte und aktualisierte Auflage. Wiesbaden: Springer Vieweg, 2020

[11] Helmut Herold, Bruno Lurz, Jürgen Wohlrab, Matthias Hopf: Grundlagen der Informatik. 3. aktualisierte Auflage. Hallbergmoos: Paerson Deutschland GmbH, 2017

[12] IT-Forensik WIKI: Dateiheader. Online im Internet, URL: https://it-forensik.fiw.hs-wismar.de/index.php/Dateiheader, Stand 28.07.2019, letzter Abruf am 17.06.2022

[13] Jessie Richardson: Volume Vs. Partition – What's The Difference?. Online im Internet, URL: https://www.alphr.com/volume-vs-partition/, Stand 06.06.2019, letzter Abruf am 17.06.2022

[14] Joep: The NTFS $Bitmap file. Online im Internet, URL: https://www.disktuna.com/the-ntfs-bitmap-file/, 28.09.2016, letzter Zugriff 17.06.2022

[15] Jonas' blog IT security & forensics: Comparison of APFS file recovery tools. Online im Internet, URL: https://blog.cugu.eu/post/comparison-of-file-recovery-in-apfs/, 15.10.2018, letzter Zugriff 31.05.2022

[16] Joseph Moronwi: The NTFS Master File Table (MFT). Online im Internet, URL: https://digitalinvestigator.blogspot.com/2022/03/the-ntfs-master-file-table-mft.html, 21.03.2022, letzter Zugriff 17.06.2022

[17] Laura Pfeiffer: Forensische Analyse des Apple File System (APFS). Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät für Allgemeine und Digitale Forensik, Bachelorarbeit, 2017, URL: https://monami.hs-mittweida.de/frontdoor/deliver/index/docId/9899/file/Bachelor+Analyse+APFS.pdf, .10.2017, letzter Zugriff 31.05.2022

[18] Lorenz Kuhlee & Victor Völzow: Computer Forensik Hacks. 1. Auflage. Köln: O'Reilly Verlag, 2012

[19] Michael Wager: DATA RUNS (RUN-LISTS) IN NTFS FILESYSTEMS. Online im Internet, URL: https://mwager.de/cyber_security/2022/01/27/ntfs-mft-example/, 27.01.2022, letzter Zugriff 17.06.2022

[20] Microsoft.Docs: Übersicht über NTFS. Online im Internet, URL: https://docs.microsoft.com/de-de/windows-server/storage/file-server/ntfs-overview, 02.04.2022, letzter Zugriff 17.06.2022

[21] Mike Cohen: Recovering deleted NTFS Files with Velociraptor. Online im Internet, URL: https://velociraptor.velocidex.com/recovering-deleted-ntfs-files-with-velociraptor-1fcf09855311, 15.11.2019, letzter Zugriff 17.06.2022

[22] NTFS.com: NTFS File Attributes. Online im Internet, URL: http://www.ntfs.com/ntfs-files-types.htm, unbekannt, letzter Zugriff 17.06.2022
[23] NTFS.com: NTFS Partition Boot Sector. Online im Internet, URL: www.ntfs.com/ntfs-partition-boot-sector.htm, unbekannt, letzter Zugriff 17.06.2022
[24] NTFS.com: NTFS System Files. Online im Internet, URL: http://www.ntfs.com/ntfs-system-files.htm, unbekannt, letzter Zugriff 17.06.2022
[25] Peter Fischer & Peter Hofer: Lexikon der Informatik. 15. überarbeitete Auflage. Heidelberg: Springer-Verlag, 2011
[26] Sabercomlogica: The Path to our File. Online im Internet, URL: https://sabercomlogica.com/en/the-path-to-our-file/, unbekannt, letzter Zugriff 17.06.2022
[27] Sascha Kersken: IT-Handbuch für Fachinformatiker – Der Ausbildungsbegleiter. 8. aktualisierte Auflage, Bonn: Rheinwerk Computing, 2018
[28] Stefan Lengauer: Sektorgrößen von Datenträgern. Online im Internet, URL: https://www.thomas-krenn.com/de/wiki/Sektorgr%C3%B6%C3%9Fen_von_Datentr%C3%A4gern, unbekannt, letzter Zugriff 17.06.2022
[29] Support.Apple (Dateisystem): Im Festplattendienstprogramm auf dem Mac verfügbare Dateisystemformate. Online im Internet, URL: https://support.apple.com/de-de/guide/disk-utility/dsku19ed921c/mac, unbekannt, letzter Zugriff 27.05.2022
[30] Support.Apple (Rolle APFS): Role of Apple File System. Online im Internet, URL: https://support.apple.com/de-de/guide/security/seca6147599e/web, 13.05.2022, letzter Zugriff 27.05.2022
[31] Support.Apple: Einführung in FileVault. Online im Internet, URL: https://support.apple.com/de-de/guide/deployment/dep82064ec40/web, Stand 27.10.2021, letzter Zugriff 27.05.2022
[32] Wikipedia: Data cluster. Online im Internet, URL: https://en.wikipedia.org/wiki/Data_cluster, 20.02.2022, letzter Zugriff 17.06.2022
[33] Wikipedia: Dateinamenserweiterung. Online im Internet, URL: https://de.wikipedia.org/wiki/Dateinamenserweiterung, 10.05.2022, letzter Zugriff 17.06.2022
[34] Wikipedia: NTFS. Online im Internet, URL: https://en.wikipedia.org/wiki/NTFS, 14.06.2022, letzter Zugriff 17.06.2022

Abkürzungsverzeichnis

APFS	Apple File System
bspw.	beispielsweise
bzw.	beziehungsweise
CPU	Central Processing Unit
etc.	et cetera
gem.	gemäß
GPU	Graphics Processing Unit
i. d. R.	in der Regel
inkl.	inklusive
MBR	Master Boot Record
MFT	Master File Table
NTFS	New Technology File System
OS	Operating System
RAM	Random Access Memory
S.	Seite
u. a.	unter anderem
z. B.	zum Beispiel

Bilderverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Beschreibung des Szenarios

Bei polizeilichen Durchsuchungsaktionen ist es meist üblich, die Wohnung nicht gewaltsam zu öffnen, sondern durch einen Bewohner eingelassen zu werden. Dies gibt potenziellen Beschuldigten die Zeit, Spuren auf EDV-Geräten zu vernichten oder zu verschleiern. Umsichtige Täter achten auch bereits im Vorfeld darauf, mögliche Spuren nicht offensichtlich auf ihren Gerätschaften liegen zu haben. Dieser Umstand macht es umso wichtiger, dass die eingesetzten Programme in der Digitalen Forensik mit möglichen Verschleierungstaktiken umgehen und digitale Spuren zuverlässig und korrekt identifizieren, wiederherstellen und dokumentieren können.

Zu den wichtigsten dieser Funktionen zählen die Erkennung von Dateisignaturen, das Feststellen versteckter und auch die Rekonstruktion gelöschter Dateien auf verschiedenen Dateisystemen. Die Evaluierung dieser Funktionalitäten soll im Mittelpunkt der folgenden Ausarbeitung stehen.

1.2 Projektziel

Projektziel ist die Evaluierung und Gegenüberstellung der Funktionen auf Dateisignatur-Ebene, sowie allgemeiner Dateirekonstruktionsfunktionen von drei Forensik-Programmen in der jeweils aktuellen Version. Da die Dokumentation vorgefundener, relevanter Daten ein fester Bestandteil der forensischen Arbeit ist, sollen eventuell enthaltene Reportingfunktionalitäten ebenfalls auf ihre Korrektheit überprüft werden.

Die zu evaluierenden Funktionen, auf denen der Fokus dieser Projektarbeit liegt, werden im Kapitel 4.3 Festlegung der zu evaluierenden Funktionen beschrieben.

Im Rahmen der Toolauswahl wurden zwei kostenpflichtige Programme und ein kostenloses Open Source Tool berücksichtigt. Dabei soll festgestellt werden, ob es zu kostenpflichtigen Tools gleichwertige Ergebnisse liefern kann. Die Auswertung soll mit dem New Technology File System von Microsoft und dem Apple File System von Apple durchgeführt werden.

Mit dem Projektergebnis soll eingeschätzt werden können, ob die Funktionen auf Dateisignatur-Ebene und die Dateiwiederherstellungs-Funktionen einwandfrei funktionieren, welches Tool das am besten geeignet für diese Aufgaben ist und ob ggf. auch Open Source Software eine Alternative sein kann.

1.3 Projektbegründung

Kriminelle können mit einfachsten Handgriffen wie beispielsweise (bspw.) das Löschen von Dateien, Ändern vom Dateiformat im Dateinamen oder der „verstecken-Funktionalität“ vom Betriebssystem relevante Dateien einfach tarnen. Umso wichtiger ist für die Kriminalitätsbekämpfung das am besten geeignete Tool zu verwenden, welches ohne Einschränkungen wahrheitsgemäße und vollumfängliche Auswertungen liefert und auf das sich blind verlassen werden kannSie müssenGenau hier setzt die vorliegende Projektarbeit an. Sie soll durch Gegenüberstellung als Entscheidungsgrundlage dienen sowie die korrekte Funktionsweise überprüfen. Da es auf dem Markt bereits einige bewährte Tools gibt, kann auf eine Eigenentwicklung verzichtet werden und eine Evaluation der bereits etablierten Tools stattfinden.

1.4 Projektabgrenzung

In diesem Projekt werden folgende Punkte nicht betrachtet:

- Verschlüsselung
- Eingebettete Dateien
- Analysen nach Keyword-Suchen im Dateinamen
- Erstellen und Auswerten eigener Dateisignaturen

2 Projektphasen und Projektplanung

Aufgrund des überschaubaren Projektumfangs und der zur Verfügung stehenden Zeit wird ein sequenzielles, wasserfallähnliches Vorgehensmodell, angewendet. Dabei entspricht jede Projektphase einem Kapitel in dieser Dokumentation:

1) Projektplanung
2) Vorbereitung
3) Planung der Durchführung der Auswertungen

4) Durchführung der Auswertungen
5) Gegenüberstellung, Bewertung und Vergleich
6) Fazit (inklusive Lessons Learned und Ausblick)

3 Kontext und Einordnung

3.1 Definition Dateisystem

Bei einem Dateisystem handelt es sich um ein dauerhaftes Ablage- und Verwaltungssystem von Dateien auf Sekundärspeichern. Es ist verantwortlich, die auf dem jeweiligen Speichermedium abgelegten Daten/Dateien in geeigneter Form zugänglich zu machen, ohne dass die Nutzer sich um die Details der internen Datenorganisation kümmern müssen. Je nach Ausformung des Dateisystems können hierbei nicht nur die Dateiinhalte, sondern auch Zusatzinformationen (z. B. Dateiname, Zugriffsrechte, Zeitstempel, etc.) abgelegt werden. Metadaten zu Dateien umfassen im Apple File System u. a. die MACB Zeitstempel.

- Modified Time (inhaltliche Änderung)
- Accessed Time (letzter Zugriff)
- Changed Time (letzte Attributsänderung)
- Birth Time (Erstelldatum)

Im Microsoft New Technology File System gibt es die Changed Time nicht.

Ein Dateisystem dient als Vermittlungsebene zwischen dem Betriebssystem und den in Dateien gespeicherten Informationen. Darüber hinaus bietet ein Dateisystem auch einen Schutzmechanismus der gewährleistet, dass Dateien nur von berechtigten Nutzern gelesen und/oder verändert werden können [10, S. 338]/[11, S. 449]/[25, S. 207].

3.1.1 New Technology File System (NTFS)

NTFS ist ein proprietäres, von Microsoft entwickeltes Dateisystem, das erstmals 1993 unter Windows NT 3.1 zum Einsatz kam und bis heute das übliche Dateisystem bei Microsoft Windows Betriebssystemen. Aktuell liegt NTFS in der Version 3.1 vor.

Je nach Konfiguration des NTFS Dateisystem werden Partitionsgrößen von 16 TB bis 8 PB unterstützt. NTFS bietet die Möglichkeit Benutzerzugriffe auf Dateien zu reglementieren sowie Daten zu verschlüsseln.

Das Herzstück dieses Dateisystems bildet die Master File Table (MFT) in der über sämtliche Dateien und Ordner Buch geführt wird [20]/[34].

Während das Partitionsschema von der Partitionstabelle (z. B. Master Boot Record (MBR)) festgelegt wird, hängt die Art wie die zu speichernden Dateien auf der jeweiligen Partition verwaltet werden vom eingesetzten Dateisystem ab. Im Fall von NTFS geschieht dies durch Verwaltungsdateien. Die untenstehende Tabelle führt einige Systemdateien und deren Funktionen auf [23].

Tabelle 1: NTFS Systemdateien und ihre Funktionen

Systemdatei	Funktion
$Boot	Bootsektor (enthält u. a. den BIOS Parameter Block)
$MFT	Master File Table (Datenbank, die einen Eintrag für jede im Dateisystem gespeicherte Datei enthält)
$MFTMirr	Kopie der $MFT
$Bitmap	Datenbank in der über die genutzten Cluster Buch geführt wird
$BadClus	Datenbank in der über als defekt erkannte Cluster Buch geführt wird
$LogFile	Transaktionslog des Dateisystems

Der Ausgangspunkt des NTFS Dateisystems ist mit der Systemdatei $Boot im ersten Sektor einer NTFS Partition zu finden. Hier identifiziert sich das Dateisystem an Byte-Offset 3 selbst als NTFS. In der Datei werden die grundlegenden Parameter des Dateisystems festgelegt, wie etwa der Speicherort (Clusternummer) der beiden weiteren Systemdateien $MFT und $MFTMirr. Die $MFT enthält zwar für jede auf dem Dateisystem gespeicherte Datei einen Eintrag über deren Speicherort, jedoch kann sie nicht auf sich selbst verweisen, um gefunden zu werden. Erst der Pointer in der $Boot Systemdatei ermöglicht das

Auffinden der $MFT und so das Funktionieren des Dateisystem als Ganzes. Als weitere Einträge sind hier die Festlegung von „Bytes pro Sektor“ und „Sektoren pro Cluster“ zu nennen.

Das Ende der Datei wird immer mit den „Magic Bytes“ **55 AA** am Ende des Sektors bestimmt (Footer) [23].

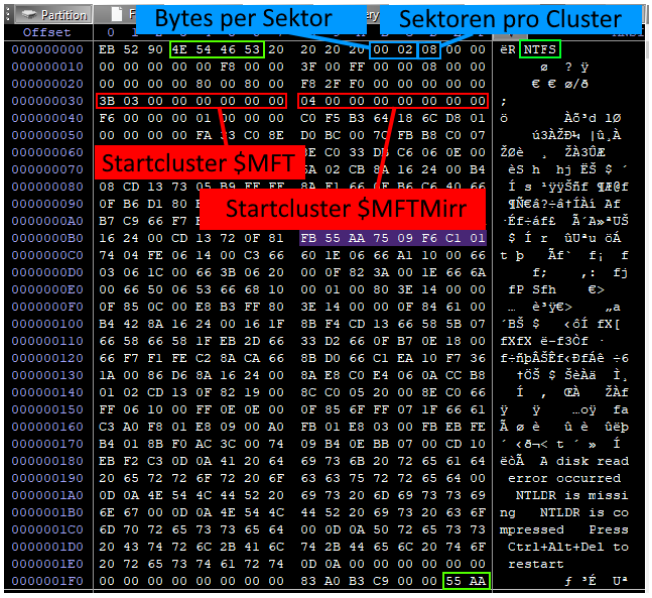


Bild 1: $Boot Systemdatei im Hexeditor

Die eigentliche Dateiverwaltung des Dateisystems übernimmt die Systemdatei „$MFT“. In dieser werden, ähnlich wie in einer Datenbank, Informationen zu jeder und jedem im Dateisystem gespeicherten Datei und Ordner hinterlegt. Welchen Raum ein Dateieintrag einnimmt wird in der $Boot bestimmt. Bei Microsoft beträgt dieser Wert jedoch standardmäßig 2 Sektoren. Die ersten 26 Einträge (0-25) sind hierbei für NTFS Systemdateien oder zukünftige Nutzung reserviert. Erst nach diesen Einträgen folgen Dateieinträge, die nicht NTFS selbst betreffen [3]/[23]/[24].

Die zwei Sektoren eines $MFT Eintrags werden unterteilt in einen Header und 13 mögliche Attribute (Blocks), von denen jedoch nicht alle Verwendung finden müssen. Die Attribute selbst verfügen wieder über einen eigenen Headerbereich. Welche Attribute in einem $MFT Eintrag aufgenommen werden, hängt etwa ab, ob der Eintrag eine Datei oder einen Dateiordner betrifft und die im Attribut dargestellten Daten hängen wieder davon ab, ob das Attribut resident

oder nicht resident in der $MFT liegt. Jeder Eintrag wird durch die Zeichenfolge FILE eingeleitet und durch die Bytefolge FF FF FF FF beendet (Footer) [22].

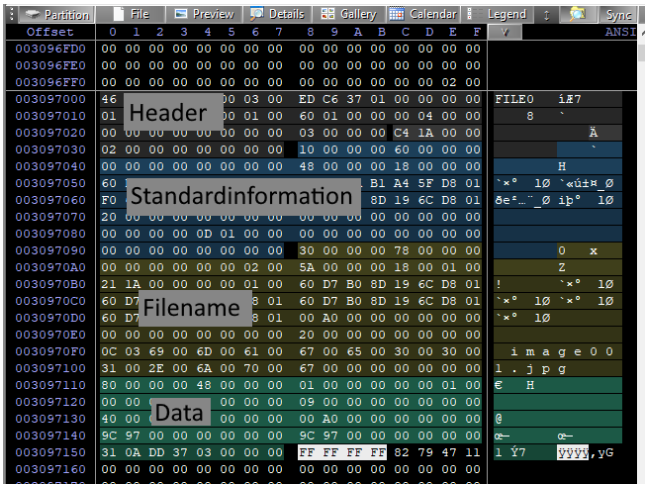


Bild 2: Dateieintrag einer Bilddatei in der $MFT

Der **Header** des Eintrags enthält neben der üblichen Startsignatur (FILE) u. a. Informationen über die Art und den Zustand der hinterlegten Datei oder des Ordners, einen Verweis auf den entsprechenden Eintrag des Transktionsprotokolls ($LogFile), eine laufende Nummer des $MFT Eintrags und eine Beschreibung des $MFT Eintrags selbst (z. B. die genutzte logische Größe des Eintrags) [16].

Im Attribut „**Standardinformation**“ sind u. a. Zeitstempelwerte der Datei / des Ordners hinterlegt aber auch Attribute wie „Hidden“, „Archive“ oder „System“.

Im Attribut „**Filename**“ wird u. a. der Datei-/Ordnername des Objekts und ein Pointer auf das Vaterverzeichnis hinterlegt.

Im „**Data**“ Block eines $MFT Eintrags werden die Inhalte der zu speichernden Datei hinterlegt, falls sie klein genug ist, um in den restlichen Eintragsbereich zu passen. Andernfalls folgen im Anschluss an den Dataheader sog. „Data Runs“ (auch: „Run Lists“), in dem auf die Speicherbereiche, die die Datei außerhalb der $MFT einnimmt, verwiesen wird.

Jedes Attribut wird durch sein erstes Byte identifiziert (z. B. Data: 0x80) [5, S. 307 ff]/[7]/[19]/[22].

Im Folgenden sollen die Details zu vier Attributarten nähere Betrachtung finden:

Details “$MFT Header”

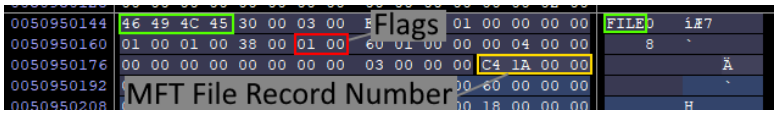


Bild 3: Header eines Dateieintrags in einer $MFT

Der Eintrag, der die Flags bestimmt kann vier Werte annehmen:

0x0000: Gelöschte Datei

0x0001: Datei

0x0002: Gelöschter Dateiordner

0x0003: Dateiordner

Details Attribut „Standard Information“

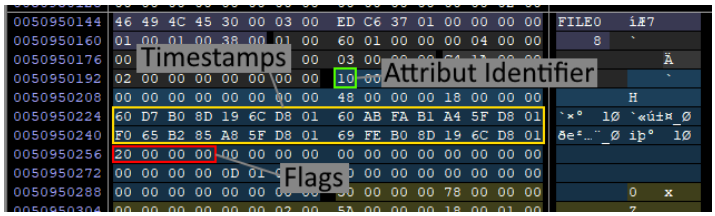


Bild 4: Standard Information Attribut in einer $MFT

Bei den Zeitstempeln, die im Attribut „Standard Information“ vorhanden sind, handelt es sich um Erstellungszeitpunkt, Änderungszeitpunkt, MFT Änderungszeitpunkt, letzter Zugriffszeitpunkt. Jeder Zeitstempel wird durch 8 Byte repräsentiert.

Der Flag-Bereich bestimmt Dateiattribute wie „Versteckt“, „System“ oder „Archiv“:

Tabelle 2: NTFS Flags

Flag	Beschreibung
0x0001	Schreibgeschützt
0x0002	Versteckt
0x0004	System

0x0020	Archiv
0x0040	Gerät
0x0080	Normal
0x0100	Temporär
0x0200	Sparse Datei
0x0400	Reparse Punkt
0x0800	Kompromittiert
0x1000	Offline
0x2000	Nicht geindext
0x4000	Verschlüsselt

Die Kombination mehrerer Flags ist möglich: 0x0006 → Versteckt + System [2]

Details Attribut „Data“

Bild 5: Data Attribut in einer $MFT

Das $MFT Attribut „Data“ wird durch den Identifier 0x80 eingeleitet. Das „Resident Flag“ gibt an, ob sämtliche Inhalte der Datei im „Data“ Attribut in der $MFT abgelegt wurden (Wert 0x00) oder ob diese aus Platzgründen auf andere Bereiche der Festplatte ausgelagert werden mussten (Wert 0x01).

Die beiden Werte für Dateigrößen geben die physische Dateigröße (ein Vielfaches der Clustergröße) und die tatsächliche Größe der Datei in Byte auf dem Datenträger an.

Ist der „Data“ Eintrag nicht resident, folgt am Ende des „Data“ Attributs eine oder mehrere Run-Lists. Diese gibt an, wie viele Cluster von der Datei einge-

nommen werden und wo diese liegen. Wird eine Datei fragmentiert (in mehreren Einzelteilen) auf dem Datenträger gespeichert, werden mehrere Run-Lists miteinander verkettet.

Run-Lists

31 0A DD 37 03

Das erste Byte gibt an, wie die nachfolgenden Bytewerte zu interpretieren sind. Es wird dabei eine Zweiteilung vorgenommen. Der erste Bytewert (hier 3 → Clusteradresse 0x0337DD) gibt an, wie viele nachfolgende Bytes den Offset zum Startcluster der Datei repräsentieren sollen. Der Zweite Bytewert gibt an, wie viele der nachfolgenden Bytes die Anzahl der durch die belegten Cluster repräsentieren (hier 1 -> 0x0A Bytes belegt).

Während der Offset der ersten Run-List wird vom Partitionsbeginn gezählt wird, verweisen mögliche weitere Run-Lists auf den Beginn der vorhergehenden Run-List. Aus diesem Grund wird hier als Datenformat ein Signed Integer verwendet, um negative Zahlen darstellen zu können [26].

Löschung von Dateien

Das Löschen von Dateien wird durch das einfache Ändern des Flag Wertes im Header des $MFT Dateieintrags durchgeführt. Eine vorhandene Datei (Wert 0x01) wird durch Ändern des Wertes in 0x00 gelöscht:

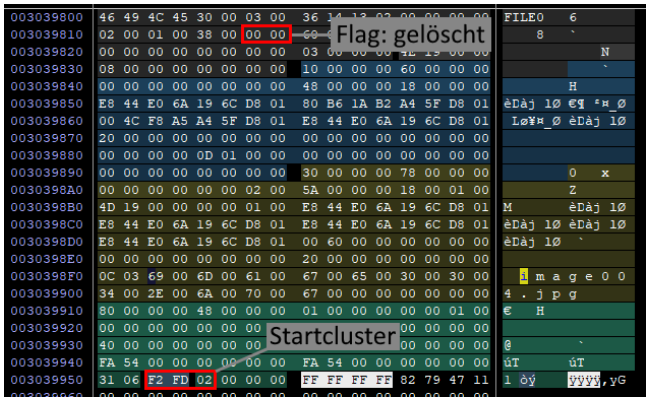


Bild 6: Dateieintrag einer gelöschten Bilddatei in der $MFT

Bei nicht residenten „Data“ Einträgen werden zusätzlich die zum Speichern der Datei genutzten Cluster in der NTFS Systemdatei $Bitmap als „frei“ gekennzeichnet:

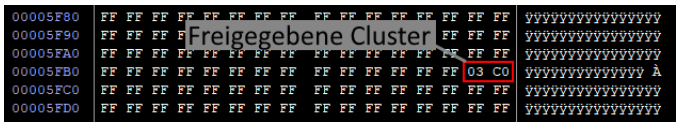


Bild 7: Freigegebene Cluster einer gelöschten Datei in der NFTS Systemdatei $Bitmap

Sowohl der $MFT Eintrag als auch die als „frei“ markierten Cluster können im Anschluss wieder für neue Daten verwendet werden.

Die Dateiinhalte im $MFT Attribut “Data” (resident) sowie die ausgelagerten (nicht residenten) Dateiinhalte auf der Festplatte bleiben hingegen unberührt, bis der $MFT Eintrag und die freigegebenen Cluster wiederverwendet werden [14]/[18, S. 45]/[21].

3.1.2 Apple File System (APFS)

Das Apple File Format ist der Nachfolger des Hierarchical File System Plus, wurde 2017 eingeführt und ist der aktuelle Standard auf allen Apple Plattformen. Neue Funktionalitäten sind bspw. eine Optimierung auf Flash/SSD-Speicher, Copy on Write, Klonen von Dateien, Snapshots, bzw. Versionierung oder das Teilen freien Speichers zwischen Volumes.

Das APFS besteht aus zwei Schichten:

- Container-Schicht (erste Schicht): Organisiert die Dateisystem-Schicht wie u. a. Speicherplatz Management und speichert allgemeine Informationen wie bspw. Snapshots sowie Metadaten des Volumes oder den Verschlüsselungsstatus. Die Größe wird in Blöcken gemessen. Ein Container ist einer Partition gleichzusetzen und kann mehrere Volumes, auch Dateisysteme genannt, verwalten.
- Dateisystem-Schicht (zweite Schicht): Diese Schicht besteht aus den Datenstrukturen die Informationen über die Verzeichnisstruktur, Metadaten der Dateien und den Dateiinhalt speichern. Die Größe wird in Bytes gemessen. Dateisystem-Objekte bestehen aus mehreren Einträgen, die als Key-Value-Paare in einem Binärbaum gespeichert sind. Bspw. besteht

ein Verzeichnis-Objekt aus einem Inode-Eintrag und einigen Verzeichnis Einstiegspunkten und einem erweiterten Attribut-Eintrag. Ein Eintrag an sich besteht aus einem Objektidentifier (eindeutig im gesamten Container), mit dem man das Objekt in der Binärbaum-Struktur finden kann.

Im Gegensatz zum New Technology File System von Microsoft gibt es keine vergleichbare Datei wie die Master File Table ($MFT), die jeweils einen Eintrag für jede Datei in dem NTFS Dateisystem Volume enthält. Die im APFS implementierte hierarchische Binärbaumstruktur übernimmt diese Aufgabe und enthält die zugehörigen Metadaten [8].

Ein weiterer Unterschied sind die unterschiedlichen Arten von gelöschten Dateien, denn neben normalen Dateien und Dateien im Papierkorb gibt es noch alte Versionen von Dateien, die durch Konvertieren des Dateisystems wiederhergestellt werden können. Darüber hinaus gibt es noch teilweise unreferenzierte Dateien (keine Referenzen mehr vom Root-Dateisystem vorhanden) und Fragmente, die allerdings nur noch über File Carving rekonstruiert werden können. Wie bereits erwähnt, bietet APFS eine Versionierung von Dateien an, in der auch alte und gelöschte Dateien zu finden sind. Die Versionierung ist durch Checkpoints umgesetzt. Dabei enthält der aktuelle Container-Superblock eine Referenz auf einen Checkpoint, der wiederrum auf den vorherigen Superblock mit dem alten Stand des Dateisystems referenziert. Dieser kann dann wiederrum auf weitere Checkpoints referenzieren. Aus dieser Struktur ergibt sich eine über Checkpoints verknüpfte Kette an Container-Superblocks. Dazu trägt auch die Copy-on-Write-Funktion des APFS Dateisystems bei, denn hier wird jeder Block kopiert, bevor Änderungen durchgeführt werden. So werden beim Speichern neuer Dateiinhalte Inkonsistenzen im Speicherzustand eliminiert, in denen ein Teil der Blöcke den alten Inhalt der Datei besitzen, andere Blöcke bereits den aktuellen Inhalt. Gleichzeitig resultiert diese Methode in eine Vielzahl an forensisch interessanten Artefakten/Versionen einer Datei [15].

Anders als das New Technology File System ist das APFS Dateisystem ein Pool Layered Dateisystem. Für ein Praxisbeispiel bzgl. Zugriff auf das APFS-Dateisystem via Sleuth Kit siehe Anlage Kapitel **Fehler! Verweisquelle konnte**

nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden..

3.1.3 Sektor

Als Sektor bezeichnet man eine vorgenommene Einteilung eines Datenspeichers in Bereiche einer bestimmten Größe. Während früher 512 Byte als Standardgröße zu nennen waren, werden heutige Datenspeicher (> 2 TB) mit einer Sektorengröße von 4096 Byte ausgeliefert, enthalten jedoch eine Emulationsschicht, die die Kompatibilität zu älteren Betriebssystemen und Software herstellt [28].

3.1.4 Cluster

Als Cluster bezeichnet man die kleinste logische Speichereinheit, die zur Datenspeicherung zugewiesen werden kann. Die Größe eines Clusters beträgt immer ein Vielfaches eines Sektors bei einer Mindestgröße von einem Sektor [32].

3.2 Definition Datei

Eine Datei ist ein Datenpaket bzw. eine Sammlung von zusammengehörenden Daten. Sie besitzen einen Namen, sind auf einem Speichermedium (z. B. Festplatte als Sekundärspeicher) abgelegt und werden durch ein Dateisystem verwaltet. Mit einer Datei können also Daten dauerhaft gespeichert werden, vor allem weil der Arbeitsspeicher klein ist und nach dem Herunterfahren bereinigt wird. Eine Datei wird durch ihren Dateinamen eindeutig bezeichnet, der als Suffix optional eine Dateierweiterung (bspw. .jpg) besitzt. Dateiname und -erweiterung sind durch einen Punkt voneinander getrennt. Durch die angegebene Dateierweiterung weiß das Betriebssystem mit welchem Anwendungsprogramm die Datei geöffnet werden soll. Die Dateierweiterung kann anders als die Dateisignatur einfach durch Anpassung des Suffixes im Dateinamen vom Nutzer verändert werden. Dadurch öffnet das Betriebssystem die Datei mit einem falschen Programm oder es kann der Inhalt einer Datei verschleiert werden. Die Daten innerhalb einer Datei können in unterschiedlichster Form organisiert sein (bspw. als lineare Bytefolge, auch Byte-Stream genannt). Neben normalen Dateien wie Textdateien, ausführbare Dateien oder Bilddateien sind noch u. a.

Verzeichnisse sowie Verweise/Links/Verknüpfungen auf andere Dateien als Unterarten zu nennen [11, S. 449]/[25, S. 206]/[27, S. 288, S. 316].

3.3 Definition Dateisignatur (inkl. Dateiheader und -footer)

Dateierweiterungen im Dateinamen können einfach von den Nutzern ohne großes Vorwissen geändert werden. Anders verhält es sich mit der Dateisignatur/Magischen Zahl/Dateityp, die in den Datei-Internas, nämlich den Dateiheadern, zu finden ist. Hierbei handelt es sich um bestimmte Bitfolgen die bei allen Dateien desselben Dateityps identisch sind. Sie identifizieren eindeutig eine Datei und geben Aufschluss über den eigentlichen Inhalt der Datei. So startet etwa eine JPG Datei mit den hexadezimalen Werten FF D8 FF und endet mit FF D9.

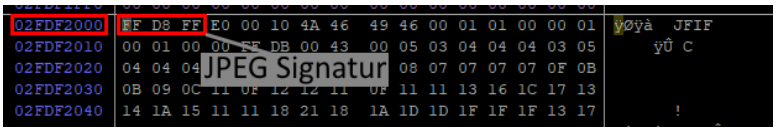


Bild 8: Header Signatur einer JPEG Bilddatei

Die Dateisignatur ist unabhängig von der Dateierweiterung und kann bspw. über einen Hex-Editor eingesehen und verändert werden, nicht aber durch Umbenennung des Dateinamens. Der Startwert ist in der Regel (i. d. R.) im Dateiheader zu finden, der Endwert im Dateifooter. Der Dateiheader ist am Beginn einer Datei zu finden, enthält Metadaten über die Datei und ist für eine ordnungsgemäße und schnelle Verarbeitung durch Programme notwendig. Im Dateiheader von Bilddateien wie einer JPG-Datei sind bspw. Informationen über GPS-Koordinaten, Blendeinstellungen oder Kamerainformationen enthalten. Auch wenn nicht alle Dateiarten, wie bspw. Textdateien (.txt), einen Header und Footer besitzen (Inhalt beginnt hier mit dem 1. Textzeichen), kann so doch mit hoher Wahrscheinlichkeit die Art der Datei bestimmt werden. Dieser Vorteil wird gerade beim File Carving relevant, denn hier werden gelöschte Dateien anhand Ihrer Dateisignatur wiederhergestellt. Dabei gehen allerdings Informationen wie der Dateiname verloren [1]/[9, S. 137 ff]/[33]/[12].

3.4 Definition Allocated Space

Unter Allocated Space versteht man Speicherbereiche auf Datenträgern, die bereits mit Daten beschrieben wurden [6]/[25, S. 37].

3.5 Definition Unallocated Space

Unter Unallocated Space versteht man Speicherbereiche auf Datenträgern, die noch keiner Bestimmung zugewiesen wurden. Dies können etwa nicht partitionierte Bereiche der Festplatte oder auch nicht zur Speicherung von Dateien genutzte Bereiche innerhalb einer Partition sein [6]/[13]/[25, S. 37].

3.6 Definition File Carving

Carving bezeichnet das Suchen von Dateien auf Datenträgern, die sich nicht mehr aus den Daten des vorhandenen Dateisystems beziehen lassen. Das kann etwa der Fall sein, wenn ein Datenträger mit einem frischen Dateisystem bespielt (formatiert) wurde. Hierbei wird der Datenspeicher (oder davon angefertigte forensische Abbilder) byteweise nach relevanten Datenspuren in Form von File Headern durchsucht. Ist ein solcher File Header gefunden worden, erfolgt die Suche nach dem zugehörigen File Footer. Aus den dazwischenliegenden Bereichen kann so die Ursprungsdatei rekonstruiert werden. Dieses Verfahren hat jedoch die Nachteile, dass hierbei keinerlei Metadaten (z. B. Zeitstempel, Zugriffsrechte) der wiederhergestellten Datei verfügbar sind. Zusätzlich dazu stößt das Verfahren an seine Grenzen, falls die zu rekonstruierende Datei nicht am Stück auf den Datenträger, sondern auf verschiedene Speicherbereiche verteilt gespeichert wurde [9, S. 137 ff]/[18, S. 99 ff].

4 Vorbereitung

4.1 Auswahl Auswertungstools

Essoll die geplante Evaluierung für die beiden Forensikprogramme X-Ways Forensics (aktuelle Version) und Magnet Forensics AXIOM (aktuelle Version) vorgenommen werden. Beides sind Standardprogramme zur forensischen Auswertung in der vorliegenden Behörde. Neben den proprietären Programmen soll ein kostenloses Open Source Tool im Vergleich berücksichtigt werden. Dabei soll geprüft werden, ob das Open Source Tool vergleichbare Ergebnisse wie die kostenpflichtigen Tools liefern kann. Dabei wurde Autopsy ausgewählt, da es

einen großen Funktionsumfang besitzt, eine gute Dokumentation verfügbar, es nutzerfreundlich, sehr beliebt und weit verbreitet ist sowie eine große Community hat. Es kann zur Dateisystemanalyse und Dateiextraktion verwendet werden. Autopsy ist die grafische Oberfläche für das Kommandozeilen Tool Sleuth Kit, das auch aus dem Studium bekannt ist und dessen Komplexität in einer übersichtlichen Oberfläche kapselt. Daneben nutzt es einige Drittanbietertools wie bspw. PhotoRec für File Carving. Im Gegensatz zu vielen spezialisierten Tools ist Autopsy breit aufgestellt. Alternativen sind bspw. Oxygen, Sift.

4.2 Auswahl der auszuwertenden Dateisysteme

Bei der Evaluierung der zu Grunde liegenden Dateisysteme wurde NTFS als Standarddateisystem im Microsoft Windows Umfeld und APFS als aktuelles Dateisystem im Apple Umfeld ausgewählt.

4.3 Festlegung der zu evaluierenden Funktionen

Folgende grundlegenden Programmfunktionen sollen einer Prüfung auf ihre Zuverlässigkeit und Korrektheit unterzogen werden:

- Auffinden von Dateien, die durch Dateisystemfunktionen vom Nutzer eines Betriebssystems nicht auffindbar sind (“verstecken” Funktionen)
- Erkennung des korrekten Dateiformats trotz veränderter Dateierweiterung
- Auffinden von gelöschten Dateien
- Reportingfunktionen in Bezug auf die oben genannten Szenarien
- Zusätzlich soll ein allgemeiner Vergleich über die Handhabung und Nutzerfreundlichkeit der geprüften Programme angestellt werden.

4.4 Auswahl der Dateiformate

Aufgrund der Häufung von Delikten im Bereich Kinderpornografie zwei gängige Mediendateiformate ausgewählt. Zusätzlich sollen auch Textdokumente aus Microsoft Word (.docx) aufgenommen werden. Als Mediendateiformate wurden das Bildformat JPEG und das Filmformat MP4 aufgrund ihrer aktuell weiten Verbreitung gewählt.

4.5 Beschreibung Systemspezifikationen

Aus Gründen der Vergleichbarkeit der Performance wurden die Auswertungen mit den Tools auf einem Rechner durchgeführt. Die Auswertungsergebnisse wurden dann für die manuelle Auswertung auf mehreren PCs verteilt. Nachfolgend sind die Angaben des Labor-Rechners zu finden, auf dem die Auswertung über die Tools erfolgte.

Tabelle 3: Systemspezifikationen

Austattung	Rechner
Central Processing Unit (CPU)/Kerne	Intel® Core™ i7-7820X (8 core)
Graphics Processing Unit (GPU)	NVIDIA GeForce RTX 2060
Random Access Memory (RAM)	64GB
Art Festplatte/Größe (Ablageort Images und Auswertungstools)	M.2 Corsair MP400 (8 TB für Images) M.2 Corsair MP510 (2 TB für Auswerteergebnisse)
Betriebssystem (OS)	Windows 10 Pro 21H1

4.6 Erstellung Asservate/USB-Sticks

4.6.1 Vorüberlegungen

Um eine aussagekräftige Prüfung der gewählten Programmfunktionen zu ermöglichen, bedarf es einer kontrollierten Umgebung in Form einer vorher festgelegten Datenbasis. Da diese Testumgebung möglichst realitätsgetreu sein sollte, wurden folgende Kriterien hierfür festgelegt:

- Vorhandene Dateiordnerstruktur mit diversen Unterebenen
- Vorhandene Dateien unterschiedlicher Formate außerhalb der zur Evaluierung relevanten Formate
- Relevante Testdateien willkürlich in der vorhandenen Ordnerstruktur und zwischen nicht relevanten Dateien verstreut

- Sobald die Testdateien in der Ordnerstruktur verteilt waren, sollten sie in folgender Weise manipuliert werden:
    - Verstecken mit Mitteln des Betriebssystems
    - Verändern des Dateinamens und der Dateiendung
    - Löschen von Dateien mit Mitteln des Betriebssystems
    - Kombinationen der oben genannten Manipulationen
- Je eine Datei pro gewähltem Dateiformat sollte als Referenzdatei unverändert bleiben.
- Da die Anforderung “Auffinden von gelöschten Dateien” sehr allgemein gehalten ist, wurde entschieden, diese Prüfung sowohl für einfach gelöschte Dateien (“Löschen” Funktion im Betriebssystem) als auch für Dateien in den ungenutzten Speicherbereichen einer Partition (Unallocated Clusters) durchzuführen, wie sie etwa nach dem Neuformatieren eines Datenträgers entstehen.
- Folglich waren insgesamt vier Datenspiegelungen zur Prüfung der gewählten Programmfunktionen anzufertigen:
    - NTFS mit vorhandener Ordnerstruktur und manipulierten Testdateien
    - NTFS neu formatiert, wobei o. g. Ordnerstruktur verloren ging
    - APFS mit vorhandener Ordnerstruktur und manipulierten Testdateien
    - APFS neu formatiert, wobei o. g. Ordnerstruktur verloren ging

4.6.2 Vorbereiten der Testumgebung

Um eine kontrollierte Umgebung für die geplante Evaluierung der Dateiwiederherstellungs- und signaturbasierten Funktionen der zu prüfenden Programme zu erhalten, wurde entschieden, hierfür einen USB-Stick der Größe 8GB einzusetzen.

Die geringe Größe des Datenträgers diente, u. a. zur Zeitersparnis beim Erstellen von Datenimages.

Bild 9: USB-Stick als Testhardware

Vor dem Bespielen des Datenträgers mit den Testdaten sollte dieser von möglichen Altdaten befreit werden. Hierzu wurde der gesamte Speicherbereich des USB-Sticks mit 0-Bits bespielt (wipen des Datenträgers). Dies wurde mit der Software „Mini Tool Partition Wizzard Free 12.6“ auf einem Windows 10 System realisiert.

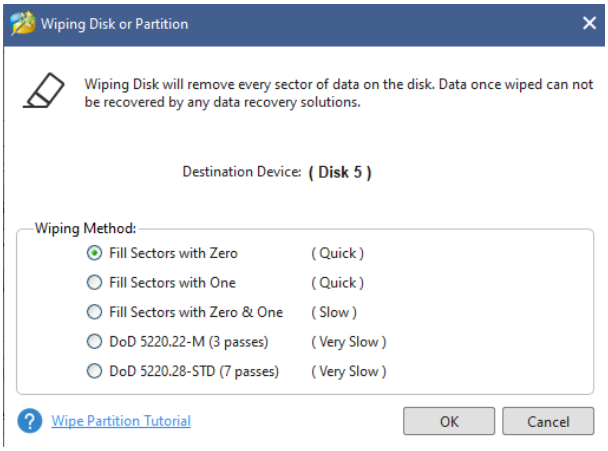


Bild 10: Auswahl der Wipe-Methode

Dieser Schritt wurde je vor der erstmaligen Formatierung des USB-Sticks mit NTFS und APFS durchgeführt.

4.6.3 Zusammenstellung der Testdaten

Um eine möglichst realitätsgetreue Umgebung für die bevorstehenden Tests zu erhalten, wurde die vollständige Ordnerstruktur eines „system32“ Ordners eines Windows 10 Systems als Basis gewählt. Der Ordner enthält nicht nur bereits vorhandene, nicht relevante Daten, sondern auch eine verzweigte Ordnerstruktur innerhalb derer die Testdateien platziert werden konnten.

Hierzu wurden je acht Bilddateien (.jpg), Filmdateien (.mp4) und Microsoft Word Dokumente (.docx) in willkürlich gewählte Unterordner eingefügt und dokumentiert.

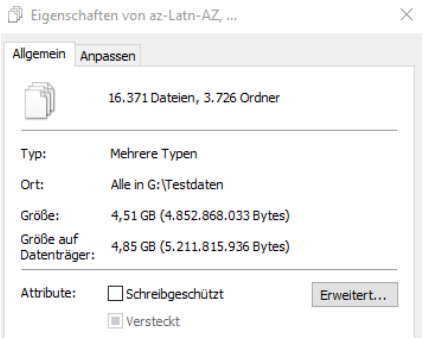


Bild 11: Datei- und Ordneranzahl der Testumgebung auf dem USB-Stick

4.6.4 Erstellung des NTFS Datenträgers

Der von möglichen Altdaten bereinigte USB-Stick wurde mit Bordmitteln eines Windows 10 Systems mit dem NTFS Dateisystem formatiert.

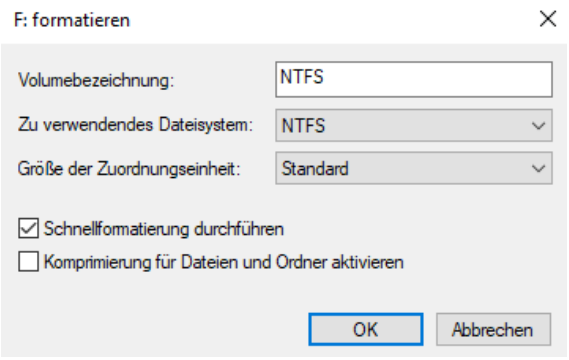


Bild 12: Formatieren des USB-Sticks mit NTFS unter Windows 10

Anschließend erfolgte das Kopieren der Testdaten auf den Datenträger und die Manipulation der einzelnen Testdateien nach dem vorher festgelegten Schema, direkt auf dem Datenträger. Das Verstecken der dafür ausgewählten Dateien erfolgte über den Aufruf der Dateieigenschaften im Windows System.

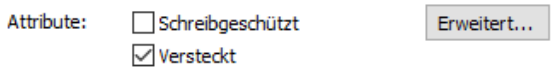


Bild 13: Genutze Systemfunktion zum Verstecken von Testdateien

Nach der Spiegelung der Daten in ein E01 Datenimage wurde der USB-Datenträger erneut unter Standardeinstellungen formatiert und erneut gespiegelt.

4.6.5 Erstellung des APFS Datenträgers

Nachdem der USB-Stick von möglichen Altdaten durch wipen befreit wurde, wurde er mit einem MacBook Pro verbunden und mit dem Dateisystem APFS formatiert. Hierbei wurden die vorgegebenen Standardeinstellungen unverändert übernommen.

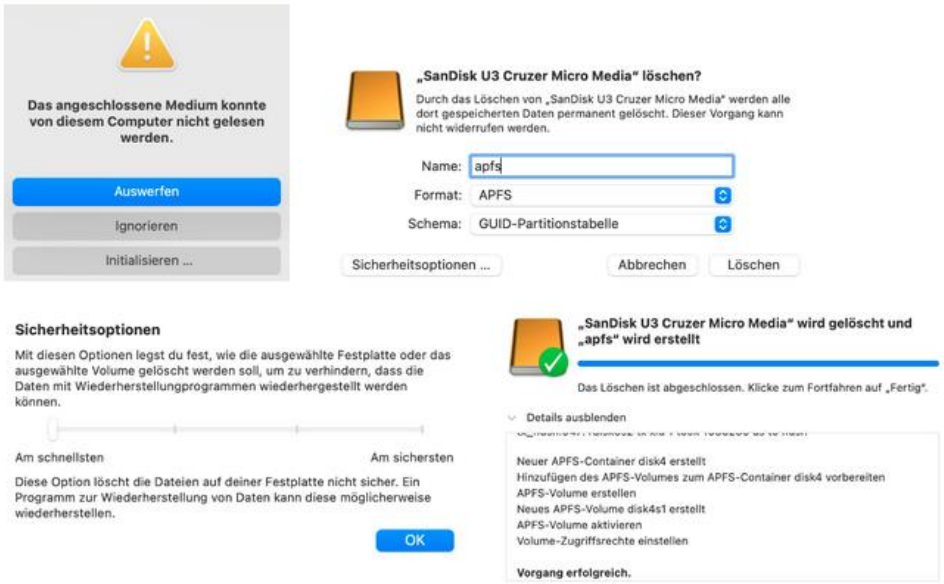


Bild 14: Formatieren des USB-Sticks unter MacOS (Version fehlt noch)

Nach dem Formatieren des Datenträgers, wurden die vorbereiteten Testdaten auf diesen kopiert und im Anschluss daran nach dem vorher festgelegten Schema manipuliert.

Das Verstecken der Dateien wurde über die Nutzung eines Terminals realisiert:

```
chflags hidden /Volumes/apfs/drivers/NVIDIA\ Corporation/Drs/image003.jpg
chflags hidden /Volumes/apfs/de-DE/Licenses/OEM/image005.jpg
chflags hidden /Volumes/apfs/de-DE/Licenses/OEM/LicMan.exe
chflags hidden /Volumes/apfs/DriverStore/FileRepository/Kratzbaum.ast
```

Bild 15: Verstecken von Dateien via Terminal

4.6.6 Erstellung der Datenimages

Sobald die Manipulationen an den Testdateien durchgeführt waren, erfolgte die Erstellung je eines Datenimages im EnCase Image File Format (.E01) für NTFS und APFS. Um dazu eine von den zu evaluierenden Programmen neutrale Software zu nutzen, wurde der Imagevorgang mit dem Programm „AccessData FTK Imager 4.5.0“ durchgeführt.

Um nachträgliche Veränderungen am USB-Datenträger zu verhindern, kam ein Hardwareschreibschutz der Marke „WiebeTECH USB 3.0 WriteBlocker“ zum Einsatz.

Bild 16: Genutzer USB-Schreibschutz

Nach dem Erstellen des jeweils ersten Datenimages, wurde der USB-Datenträger erneut mit Bordmitteln des jeweiligen Betriebssystems formatiert und wieder mit „AccessData FTK Imager“ in eine E01-Datei gespiegelt.

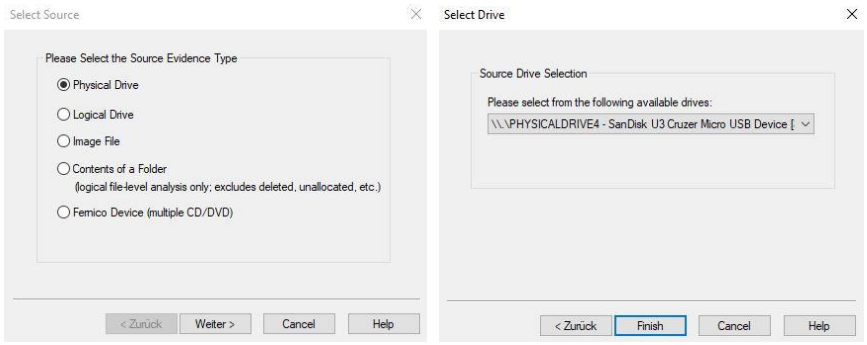


Bild 17: Erstellen eines Datenimages mit FTK Imager 1 & 2/5

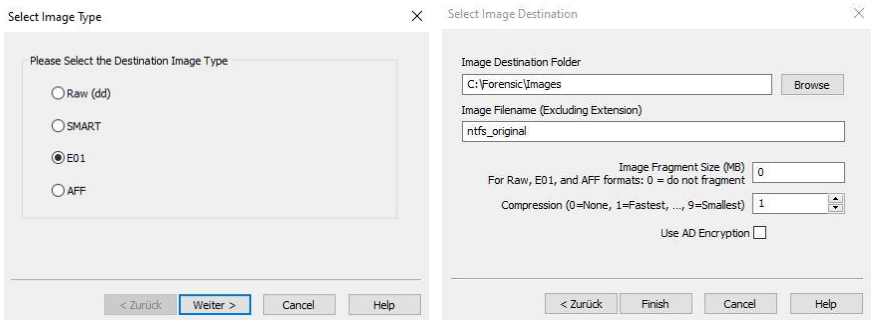


Bild 18: Erstellen eines Datenimages mit FTK Imager 3 & 4/5

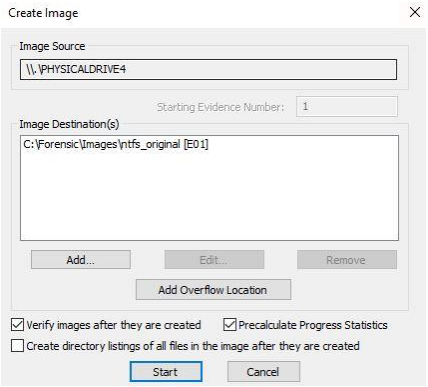


Bild 19: Erstellen eines Datenimages mit FTK Imager 5/5

Die Korrektheit des Datenimages gegenüber des Ursprungsdatenträgers wurde durch einen von “AccessData FTK Imager 4.5” durchgeführten Hashwertabgleich geprüft. Hier wurde bei jedem Durchgang die Fehlerfreiheit bestätigt.

NTFS Original:

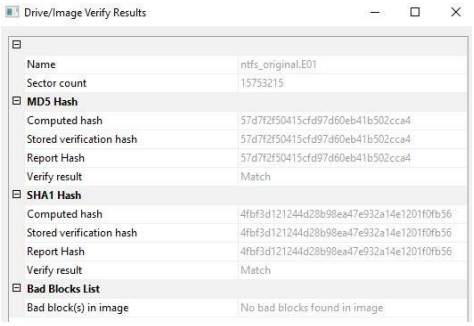


Bild 20: Hashverifikation "NTFS Original"

NTFS Formatiert:

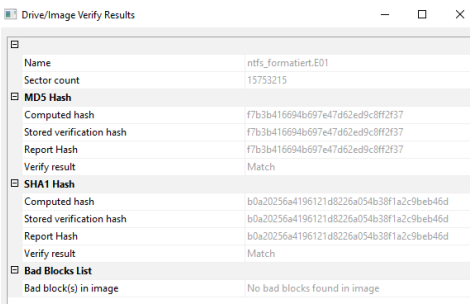


Bild 21: Hashverifikation "NTFS Formatiert"

APFS Original:

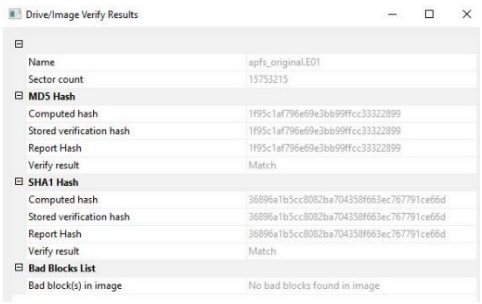


Bild 22: Hashverifikation "APFS Original"

APFS Formatiert:

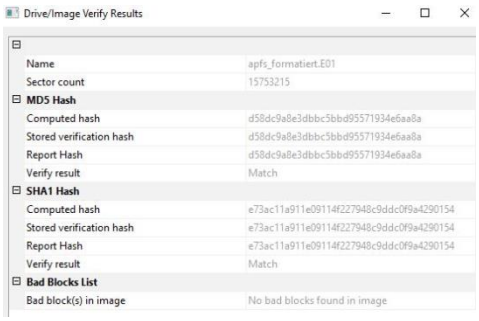


Bild 23: Hashverifikation "APFS Formatiert"

4.6.7 Problemanalyse (welche Schwachstellen können evtl. identifiziert werden)?

Da es sich bei den zu vergleichenden Softwaren um stark unterschiedliche Produkte handelt, die über einen kaum vergleichbaren Funktionsumfang und Konfigurationsmöglichkeiten verfügen, ist ein Vergleich nur bei sehr eng gesteckten Einzelfunktionen möglich und sinnvoll.

Bei der Erstellung der Testimages musste ein Kompromiss zwischen wünschenswerter Konfiguration und Praktikabilität eingegangen werden. Wünschenswert wäre das vorangehende Wipen des USB-Sticks mit Zufallsdaten gewesen, um durch eine Art “Grundrauschen” die zu prüfenden Tools vor eine weitere Herausforderung in Sachen “Carving” zu stellen. Stattdessen musste auf ein Überschreiben mit ausschließlich Null-Bits zurückgegriffen werden, da sich dadurch die Datenimages gut komprimiert und per Internet übertragen ließen.

5 Planung der Durchführung der Auswertungen

5.1 Planung des Workflows

Tabelle 4: Planung der Durchführung und des Workflows

Laufende Nummer	Schritt
1	Korrektheit der Images auf aufgrund der Hashes überprüfen
2	Software vorbereiten und Images hinzufügen
3	Auswahl der Bearbeitungsschritte wie Signaturüberprüfung, File Recovery, File Carving (Konfiguration) und starten der automatisierten Auswertung
4	Manuelle Auswertung/Überprüfung Korrektheit der Ergebnisse
5	Erstellung und Überprüfung des Reports

5.2 Ablauf des Workflows am Beispiel Autopsy

Der von Autopsy empfohlene Workflow deckt sich mit dem in Kapitel 5.1 Planung des Workflows geplanten Vorgehen, was sich positiv auf die Bedienbarkeit auswirkt. Nachfolgend eine Zuordnung der Schritte zum Autopsy-Vorgehen:

Zu 2) Erstellen eines Case: Ein Case ist ein Container für ein oder mehrere Datenquellen. Ohne einen Container kann keine Auswertung erfolgen.

Zu 2) Hinzufügen einer oder mehrerer Datenquellen: Datenquellen können Images aber auch lokale Dateien sein.

Zu 3) Analyse mit Ingest Modules: Die Ingest Module analysieren automatisiert im Hintergrund die hinzugefügte Datenquelle und können bedarfsorientiert aktiviert oder deaktiviert werden. Die Auswertungsergebnisse werden in Echtzeit auf der Oberfläche angezeigt und können auch via Alerts überwacht werden. Es stehen Module für die unterschiedlichsten Aufgaben zur Verfügung. Eine Beschreibung relevanter und interessanter Module kann in der Anlage im Kapitel 0

Beschreibung Ingest Module am Beispiel Autopsy nachgeschlagen werden. Eigene Module können programmiert oder von Drittanbietern bezogen werden.

Zu 4) Manuelle Analyse: Erfolgt auf den Ergebnissen der automatisierten Auswertung mittels der ausgewählten Ingest Module. Hierfür werden die Ergebnisse auf der Autopsy-Oberfläche nach Dateiinhalten oder sonstigen gerichtlich verwertbaren Beweisen gesucht. Treffer können für nachfolgende Auswertungen oder die Reporterstellung mit einem Tag markiert werden.

Zu 5) Erstellung eines Reports: Zum Schluss der Auswertung erstellt der User anhand der markierten Ergebnisse einen Report [4].

5.3 Festlegen der Vergleichskriterien

Folgende Vergleichsrubriken sollen im Vergleich berücksichtigt werden. Die Gewichtung der Rubriken ergibt in Summe 100%, die auf die einzelnen Punkte aufgeteilt wird. Die Rubriken erhalten jeweils konkrete Vergleichskriterien mit eigenen Gewichtungen. Im Kriterienkatalog können für die einzelnen Vergleichskriterien Punkte zwischen 0 (nicht vorhanden) und 5 (sehr gut) vergeben werden, um die Objektivität, Validität und Reliabilität zu gewährleisten. Die vergebenen Bewertungen werden mit der jeweiligen Gewichtung des Vergleichskriteriums multipliziert. Aus den gewichteten Ergebnissen der Vergleichskriterien wird dann die Summe gebildet. Zum Schluss werden die Summen dann wiederrum mit den Gewichtungen der Vergleichsrubriken multipliziert und addiert. Das Ergebnis ist schließlich die finale Beurteilung (Gesamtpunkte). Auf die Auflistung der einzelnen Kriterien wird an dieser Stelle verzichtet und auf den eingebetteten Kriterienkatalog in Kapitel 7.1 Analyse der Auswertungsergebnisse (Kriterienkatalog) verwiesen.

Tabelle 5: Festlegen der Vergleichsrubriken

Vergleichsrubrik	Gewichtungsfaktor in Prozent
Korrektheit	55
Funktionsumfang	25
Kosten	5
Performance	5
Bedienbarkeit	10

6 Durchführung der Auswertungen

6.1 Autopsy

Die im Test der gewählten Funktionen zu Grunde liegende Version von Autopsy ist 4.19.3.

6.1.1 Konfiguration

Nachfolgend eine Übersicht der durchgeführten Schritte zum Starten der Auswertung am Beispiel Autopsy.

Autopsy organisiert seine Daten in Cases. Dabei kann ein Case ein oder mehrere Datenquellen besitzen

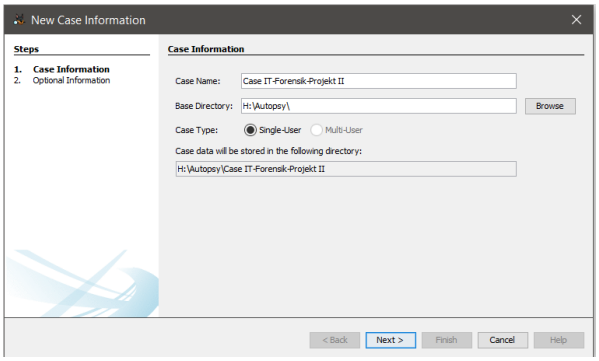


Bild 24: Anlegen eines neuen Falls unter Autopsy

Angabe diverse Metadaten

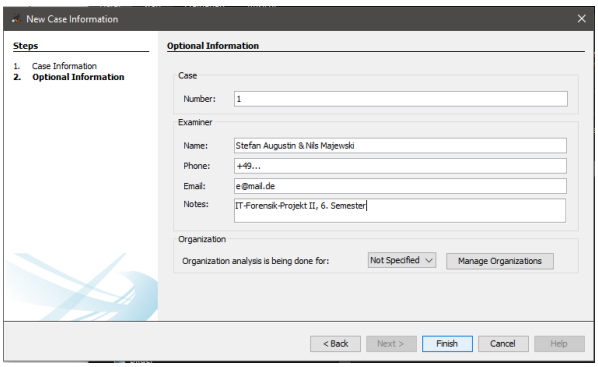


Bild 25: Angabe von Metadaten zu einem neuen Fall unter Autopsy

Die folgenden Schritte wurden je auszuwertender Datenquellen ausgeführt. Dabei wurden sie alle einem Case hinzugefügt.

Anlegen eines sprechenden Hostnamen. Der Hostname ist gleichzusetzen mit dem Namen des zu untersuchenden Geräts

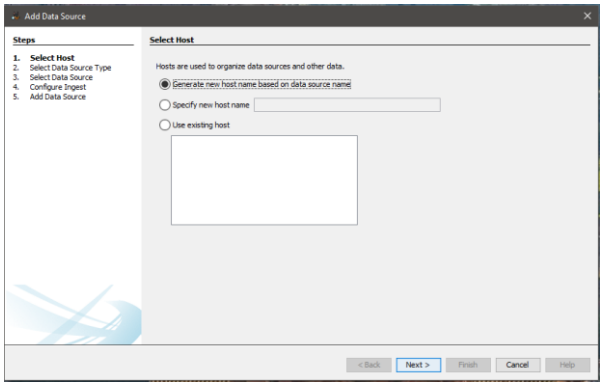


Bild 26: Festlegen des Hostnamens unter Autopsy

Auswahl Disk Image als Datenquelle (aktuell unterstütze Dateiformate: E01, raw (dd))

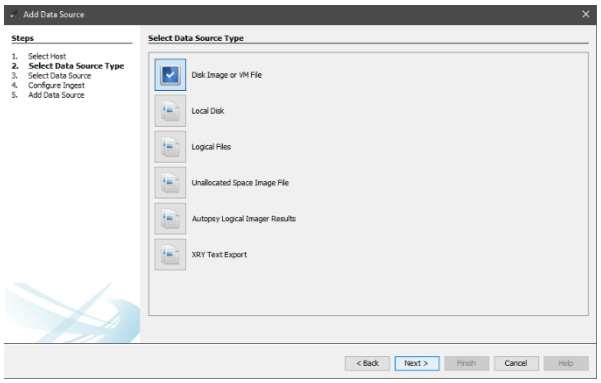


Bild 27: Auswahl der Datenquelle unter Autopsy

Auswahl des Images

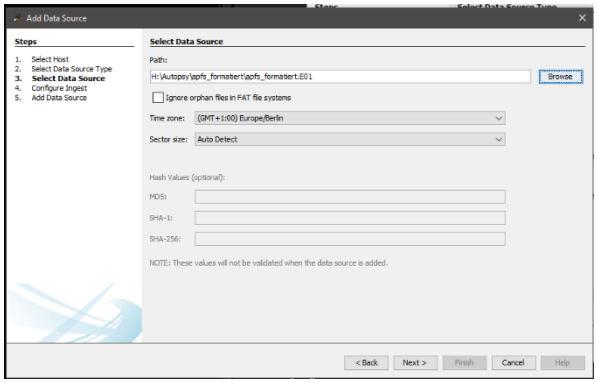


Bild 28: Hinzufügen eines .E01 Images unter Autopsy

Auswahl der Ingest (aufnehmen) Modules

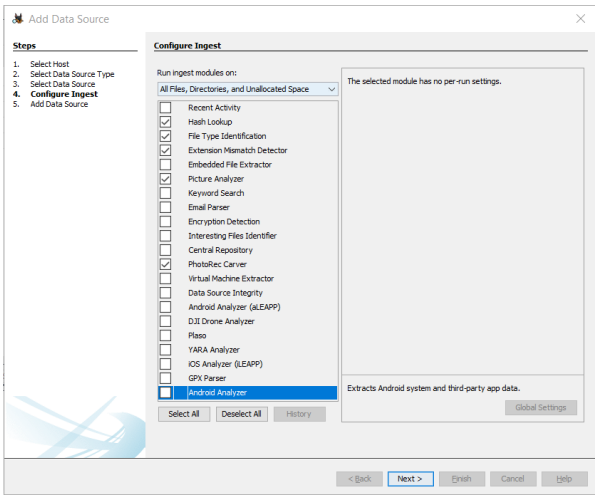


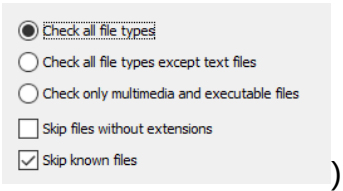
Bild 29: Auswahl der Aufbereitungsmodule unter Auotpsy

In der Default-Konfiguration waren alle Module aktiviert, dies hat in einem Testlauf zu einer Laufzeit von über 12 Stunden geführt.

Folgende für die Aufgabenstellung relevante Module werden verwendet. Es wurde größtenteils die Default-Konfiguration beibehalten. Für mehr Details siehe Kapitel 0

Beschreibung Ingest Module am Beispiel Autopsy in der Anlage:

- Hash Lookup (Default-Konfiguration)
- File Type Identification Module (Default-Konfiguration)
- Extension Mismatch Detector Module (Konfiguration wurde angepasst:



)

Bild 30: Feinkonfiguration des Extension Mismatch Detector Moduls unter Autopsy)

- Picture Analyzer Module (Default-Konfiguration)
- PhotoRec Carver Module (Default-Konfiguration)

Nachdem die Module ausgewählt und die vorherige Maske mit „Next“ bestätigt wurde, startet automatisch die automatisierte Analyse.

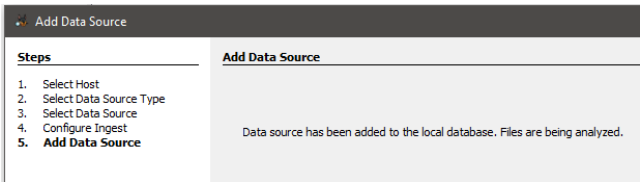


Bild 31: Autopsy Systemmeldung: Analyse in Arbeit

Die Oberfläche von Autopsy ist intuitiv bedienbar und im „Windows Look and Feel“ gehalten. Es gibt einen eingebauten Result Explorer u. a. für Bild und Video-Dateien und es werden viele sowie konfigurierbare Attribute für eine Datei angezeigt. Als Beispiel ist hier eine Eigenschaft zu nennen, die Aufschluss darüber gibt, ob die Datei aus dem Allocated oder Unallocated Space stammt.

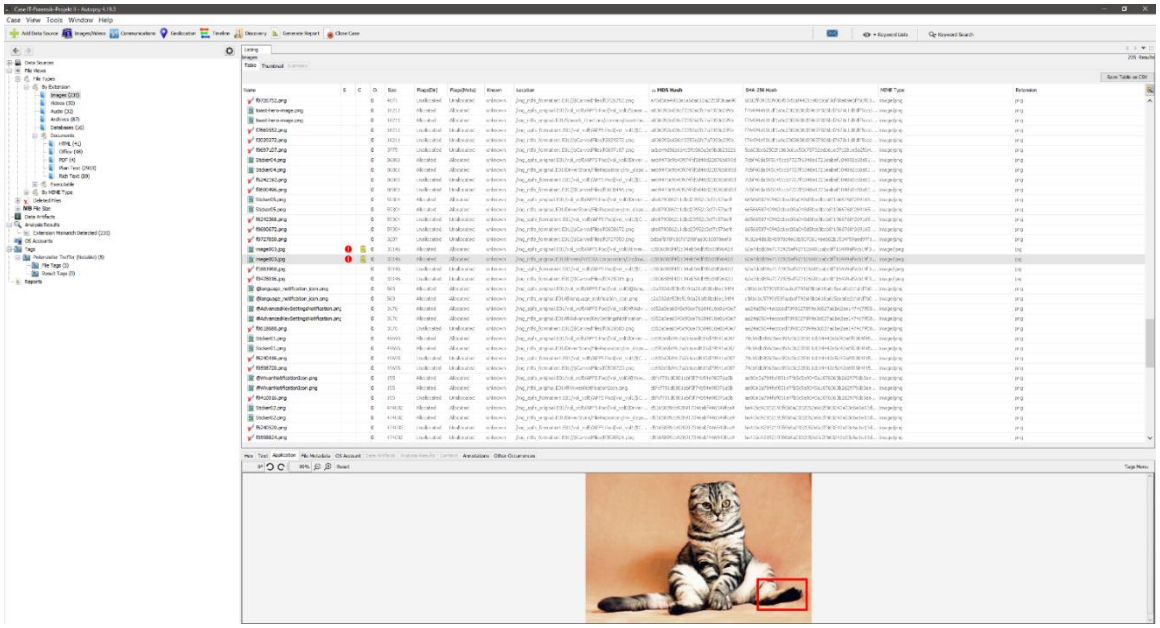


Bild 32: Autopsy Benutzeroberfläche nach der Aufbereitung durch die gewählten Module

Neben dem Result Explorer gibt es noch weitere Reiter in denen bspw. Metadaten (inkl. File Flags wie bspw. Hidden für versteckte Dateien) oder Kommentare der Tags hinterlegt sind.

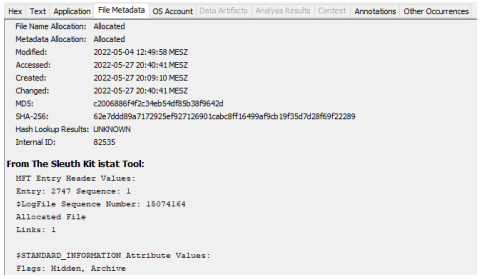


Bild 33: Ansicht von Dateimetadaten unter Autopsy

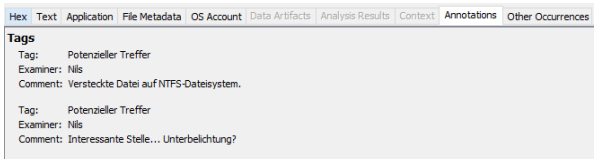


Bild 34: Ansicht gesetzter Kommentare zu einer Datei unter Autopsy

Auf der linken Seite ist ein Menü zu finden, in dem die gefundenen Dateien gruppiert nach diversen Kriterien vorzufinden sind:

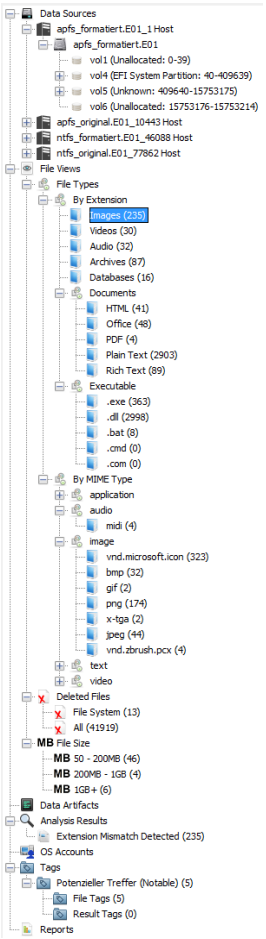


Bild 35: Menüstruktur zu eingruppierten Dateien unter Autopsy

Dateien aus dem Allocated und Unallocated Space können einfach über ein Kontextmenü extrahiert (siehe 1) oder mit einem externen Programm je nach Dateityp geöffnet werden (siehe 2):

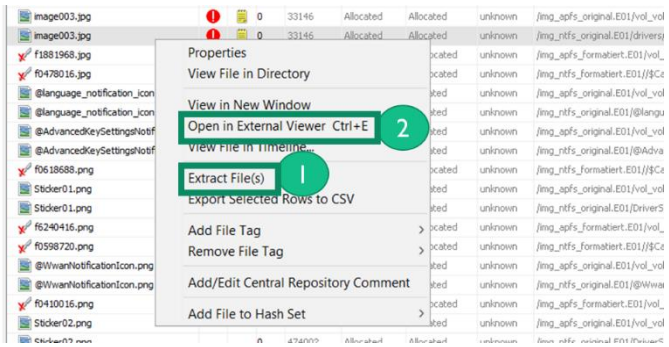


Bild 36: Menü zum Export von Dateien und zum Betrachten in einem externen Sichtungsprogramm unter Autopsy

6.1.2 Ergebnisse

Wurde eine Datei mitsamt ihren Metadaten und den durchgeführten Manipulationen erkannt, wird dies als “OK” vermerkt. Bei Images von formatierten Datenträgern reicht das bloße Wiederherstellen um als Erfolg gewertet zu werden, da beim hier angewandten File Carving ohnehin nicht mit Metadaten gerechnet werden kann. Dieser Fall wird auch mit “OK” dokumentiert.

Auffälligkeiten oder Fehler werden als Kommentare erfasst.

Tabelle 6: Auswertungsergebnisse Autopsy

Datei	Pfad	Hinweis	APFS Formatiert – Gefunden?	APFS-Original – Gefunden?	NTFS Formatiert – Gefunden?	NTFS Original – Gefunden?
image001.jpg	/drivers/	unverändert	OK	OK	OK	OK
image002.jpg	/drivers/	Gelöscht	OK	OK	OK	OK
image003.jpg	/drivers/NVIDIA Corporation/Drs/	versteckt	OK	OK	OK	OK
image004.jpg	/de-DE/Licenses/OEM/	umbenannt in "License Agreement.pdf"	OK	OK + „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung
image005.jpg	/de-DE/Licenses/	versteckt + gelöscht	OK	OK	OK	OK

	OEM/					
image006.jpg	/de-DE/Licenses/OEM/	umbenannt in "LicMan.exe" + versteckt	OK	OK + „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung
image007.jpg	/DriverStore/FileRepository/	umbenannt in "Kratzbaum.ast" + versteckt + gelöscht	OK	OK + „Extension Mismatch Detected“ Markierung + schwierig zu finden (nur unter „By MIME-Type“ zu finden	OK	OK + Keine „Extension Mismatch Detected“ Markierung
image008.jpg	/LogFiles/CloudFiles	umbenannt in "Default Log" + gelöscht	OK	OK + Keine „Extension Mismatch Detected“ Markierung	OK	OK + Keine „Extension Mismatch Detected“ Markierung
video001.mp4	/networklist/icons/	unverändert	OK	OK	OK	OK
video002.mp4	/SecureBoot Updates/	gelöscht	OK	OK	Nicht gefunden unter Reiter „Videos“ → „By Extension“, „By MIME Type“ oder „Deleted Files“	OK
video003.mp4	/winevt/Logs/	versteckt	OK	OK	OK	OK
video004.mp4	/Keywords/	umbenannt in "CV.pdf"	OK	OK + „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung angezeigt
video005.mp4	/Recovery/	versteckt + gelöscht	OK	OK	OK	OK
video006.mp4	/Tasks_Migrated/Mozilla	umbenannt in "Katzengras.plt" + versteckt	OK	OK + Keine „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung
video007.mp4	/wbem/Framework/root	umbenannt in "Directory" + versteckt +	OK	OK + .mp4 als Dateierweiterung	OK	OK

		gelöscht		ergänzt		
video008.mp4	/AdvancedInstallers	umbenannt in "walnut.vbs.mp4" + gelöscht	Nicht gefunden unter Reiter „Videos“ → „By Extension“, „By MIME Type“ oder „Deleted Files“	OK + Keine „Extension Mismatch Detected“ Markierung	OK	OK + Keine „Extension Mismatch Detected“ Markierung
word1.docx	/BestPractices/v1.0	unverändert	OK	OK	OK	OK
word2.docx	/Bthprops	gelöscht	OK	OK	OK	OK
word3.docx	/AppV	versteckt	OK	OK	OK	OK
word4.docx	/catroot2	umbenannt in "text.jpg"	OK	OK + „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung
word5.docx	/CodeIntegrity/Tokens/Active	versteckt + gelöscht	OK	OK	OK	OK
word6.docx	/config/Journal	umbenannt in "Internet.exe" + versteckt	OK	OK + „Extension Mismatch Detected“ Markierung	OK	OK + „Extension Mismatch Detected“ Markierung
word7.docx	/downlevel	umbenannt in "Katzenfutter.lst" + versteckt + gelöscht	OK	OK + Keine „Extension Mismatch Detected“ Markierung	OK	OK + Dateityp als + Keine „Extension Mismatch Detected“ Markierung
word8.docx	/ProximityToast	umbenannt in "Grumpy.cat" + gelöscht	OK	OK + Keine „Extension Mismatch Detected“ Markierung	OK	OK + Keine „Extension Mismatch Detected“ Markierung

Die gelöschten Dateien im .Trashes im APFS-Dateisystem sind auch über das Menü im APFS Pool erreichbar:

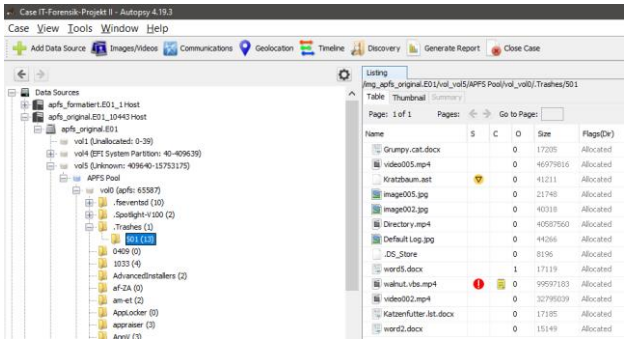


Bild 37: Gelöschte Dateien im Menü "APFS Pool" unter Autopsy

6.1.3 Analyse

Im Folgenden sollen, falls möglich, die im Test getroffenen Feststellungen analysiert werden.

6.1.3.1 Durch File Carving nicht aufgefundene Bilddatei unter NTFS formatiert

Die Videodatei “video002.mp4” (gelöscht) konnte beim formatierten NTFS Datenimage nicht durch File Carving rekonstruiert werden. Das manuelle Suchen nach der Datei ergab, dass sie sich mit intaktem Header und Footer auf dem Datenimage befindet und so aufgefunden hätte werden müssen. Sie liegt intakt zwischen den Sektoren 6055256 und 6119304 im Image.

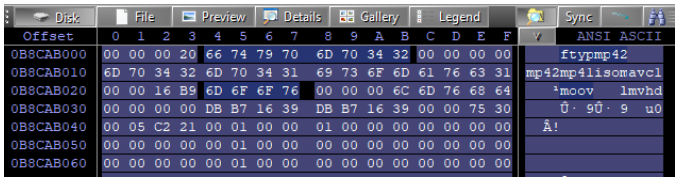


Bild 38: Nicht mit Autopsy gefundene, gelöschte Datei "video002.mp4" unter NTFS formatiert (Dateibeginn)

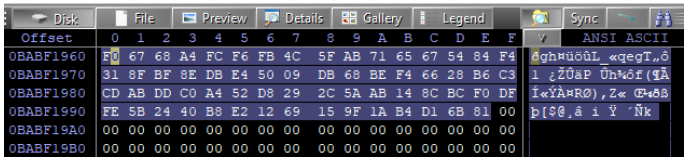


Bild 39: Nicht mit Autopsy gefundene, gelöschte Datei "video002.mp4" unter NTFS formatiert (Dateiende)

Bei der Anzeige nach Dateierweiterung werden nur die APFS-Dateien angezeigt:

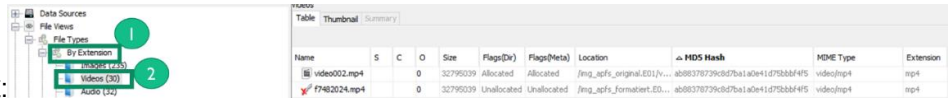


Bild 40: Autopsy Oberfläche Anzeige nach Dateierweiterung, fehlende Videodatei auf NTFS formatiert

Genauso wie bei der Anzeige nach Dateisignaturen:

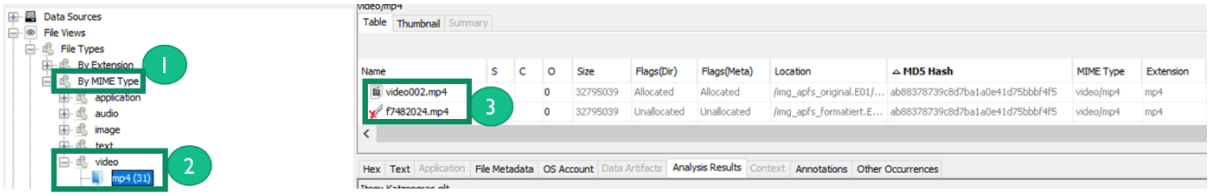


Bild 41: Autopsy Oberfläche Anzeige nach Dateisignatur, fehlende Videodatei auf NTFS formatiert

Bei beiden Ansichten wurde auch die Datei auf dem formatierten NTFS erwartet. Unter den gelöschten Dateien (File System, All) ist lediglich die Videodatei zu finden, die über die MFT auf dem Original NTFS-Image rekonstruiert wurde.

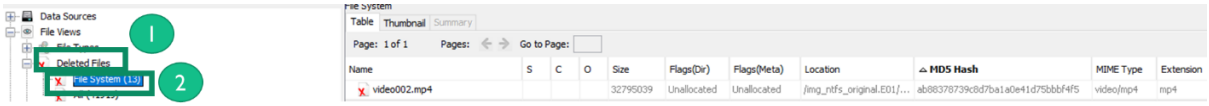


Bild 42: Autopsy Oberfläche Anzeige gelöschte Dateien File System, fehlende Videodatei auf NTFS formatiert

6.1.3.2 Durch File Carving nicht aufgefundene Bilddatei unter APFS formatiert

Die Videodatei “video008.mp4” (gelöscht und umbenannt in „walnut.vbs.mp4“) konnte beim formatierten APFS Datenimage nicht durch File Carving rekonstruiert werden. Das manuelle Suchen nach der Datei ergab, dass sie sich mit intaktem Header und Footer auf dem Datenimage befindet und so aufgefunden hätte werden müssen. Sie liegt intakt zwischen den Sektoren 1468528 und 1663069 im Image.

Dateibeginn:

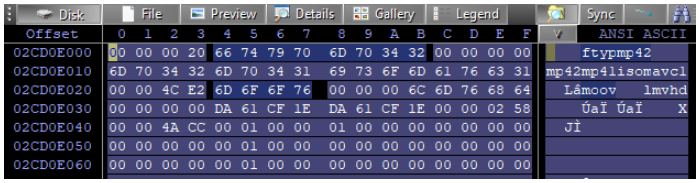


Bild 43: Nicht mit Autopsy gefundene, gelöschte Datei "video008.mp4" unter APFS formatiert (Dateibeginn)

Dateiende:

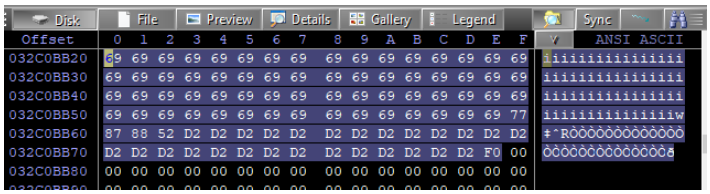


Bild 44: Nicht mit Autopsy gefundene, gelöschte Datei "video008.mp4" unter APFS formatiert (Dateiende)

Auf die Screenshots der Oberfläche wie im vorherigen Unterkapitel wurde an dieser Stelle verzichtet.

6.1.3.3 Alleinstellungsmerkmal 1 gegenüber Axiom und X-Ways

Nur Autopsy kann einwandfrei mit den gelöschten bzw. im Papierkorb befindlichen Dateien auf dem APFS-Dateisystem umgehen und erkennt die Flags (Tags wurden vorher angelegt).

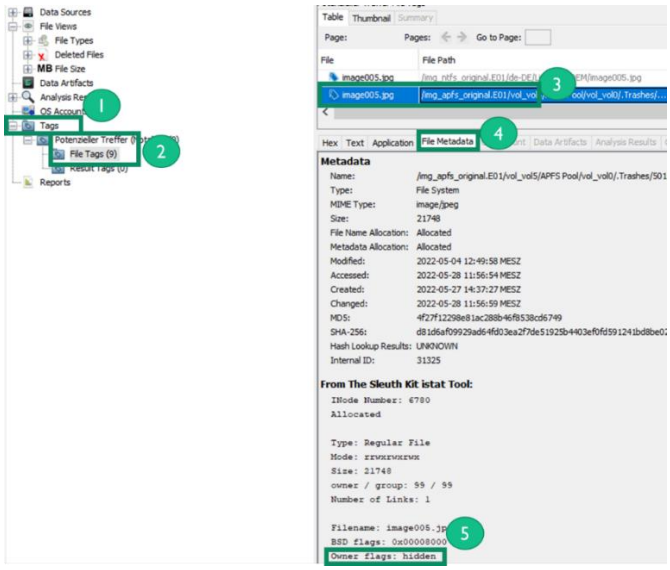


Bild 45: Alleinstellungsmerkmal 1 Autopsy – Anzeige von Flags bei gelöschten bzw. befindlichen Dateien (APFS)

Bei gelöschten Dateien auf dem NTFS-Dateisystem können nur Autopsy und X-Ways die (versteckt) Flags über die MFT korrekt extrahieren.

6.1.3.4 Alleinstellungsmerkmal 2 gegenüber Axiom und X-Ways

Weiteres Alleinstellungsmerkmal ist, dass interessante Stellen in Bildern markiert werden können (inkl. Kommentar), wie am rechten Fuß zu erkennen ist.

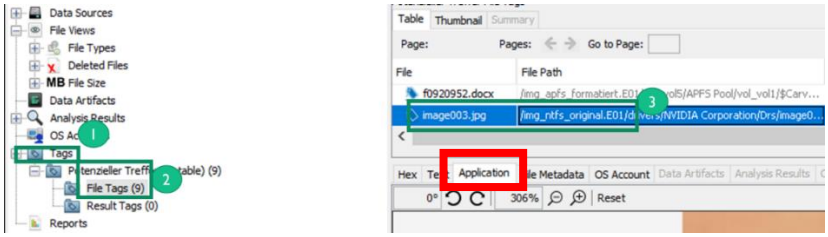


Bild 46: Alleinstellungsmerkmal 2 Autopsy – Markieren von interessanten Bildinhalten 1/2

Bild 47: Alleinstellungsmerkmal 2 Autopsy – Markieren von interessanten Bildinhalten 2/2

6.1.4 Erstellung Report

Wichtiges Werkzeug für die Erstellung von Reports sind Tags, dabei gibt es zwei Ausprägungen:

- Tag File: Wird zum Markieren von relevanten Dateien verwendet
- Tag Result: Wird zum Markieren von relevanten Ergebnissen verwendet

Mit Tags werden die Inhalte der Reports gesteuert. Sie können einfach über die Oberfläche hinzugefügt werden:

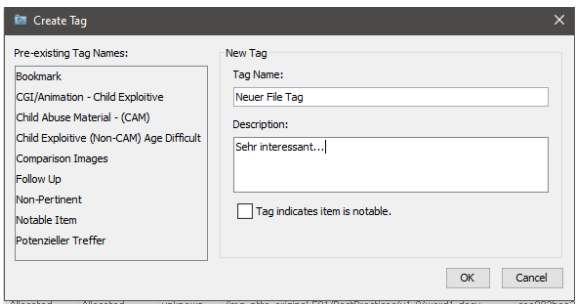


Bild 48: Erstellen eines Tags unter Autopsy

Außerdem gibt es bereits einige vorkonfigurierte Tags:

- Bookmark - Default tag for marking files of interest
- CAT-1 through CAT-5 - For law enforcement use
- Follow Up - Default tag for marking files to follow up on
- Notable item - Default tag for indicating that an item should be marked as notable in the central repository

Bild 49: Liste von vorkonfigurierten Tags unter Autopsy

Tags mit Kommentaren können dann ohne großen Aufwand über das Kontextmenü hinzugefügt werden (siehe 1, 2). Markierte Dateien erhalten dann eine Markierung. Falls ein Kommentar beim Tag hinterlegt ist, wird dies separat angezeigt (siehe 3).

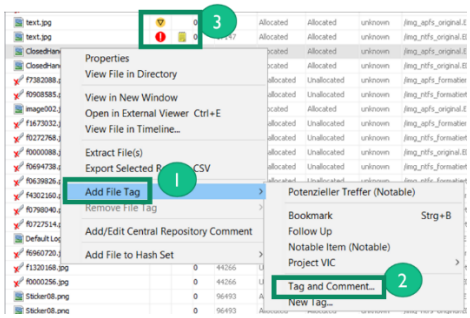


Bild 50: Setzen eines Tags unter Autopsy

Bei Bilddateien können auch Tags mit Markierungen und Kommentar im Bild hinterlegt werden, um interessante Stellen zu markieren.

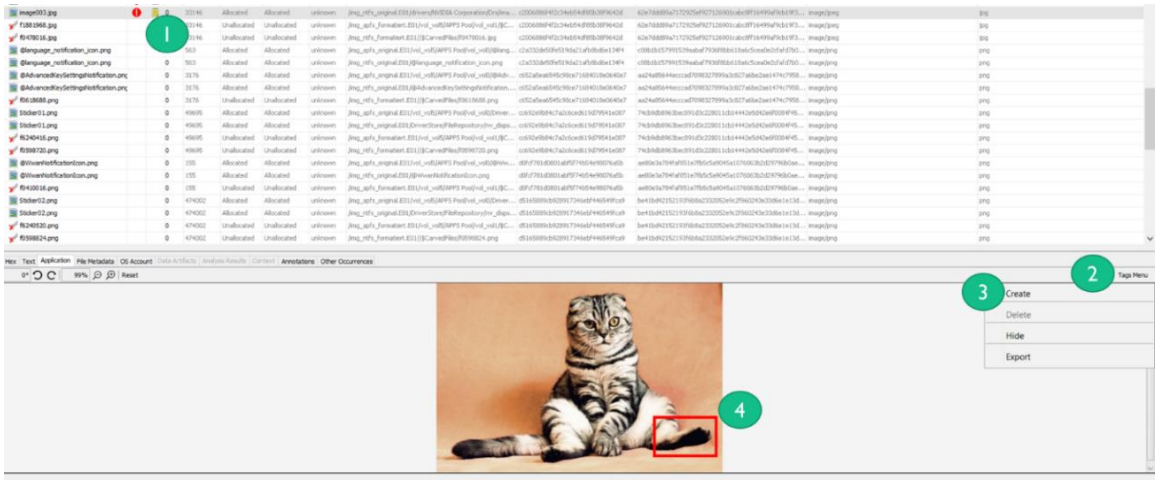


Bild 51: Hervorheben von Bereichen einer Bilddatei durch Markierungen unter Autopsy

Nachdem interessante Dateien getaggt und Bildinhalte markiert wurden, kann ein Report über das Report Modul erstellt werden. Das Report Modul ist in den Kopfreitern unter „Generate Report“ erreichbar. Leider gibt es nicht die Möglich-

keit einen PDF-Report zu erstellen. Daher wurde die HTML-Report-Funktionalität geprüft.

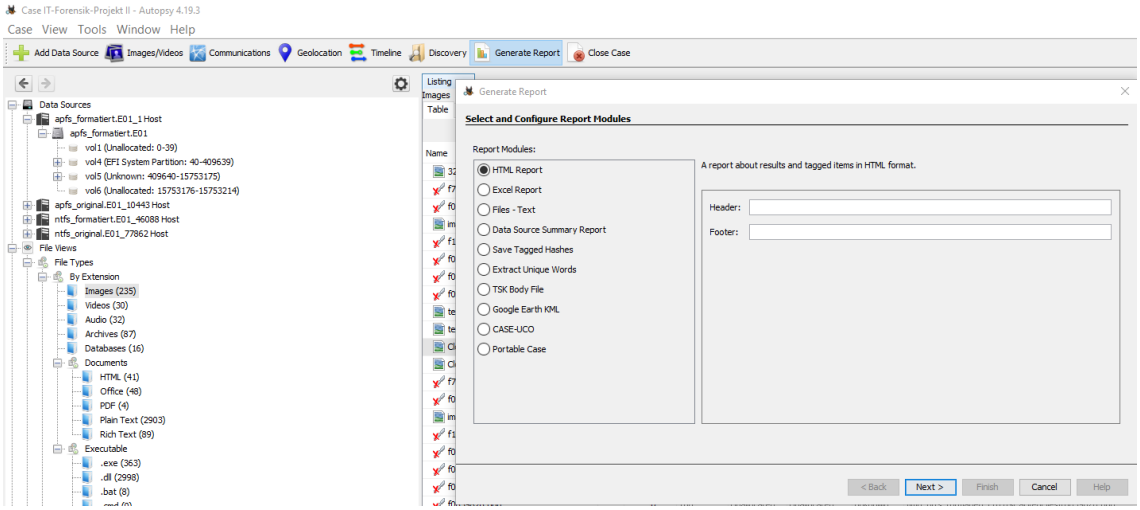


Bild 52: Auswahl der Art eines Reports unter Autopsy

In der folgenden Maske können die für den Report relevanten Asservate ausgewählt werden:

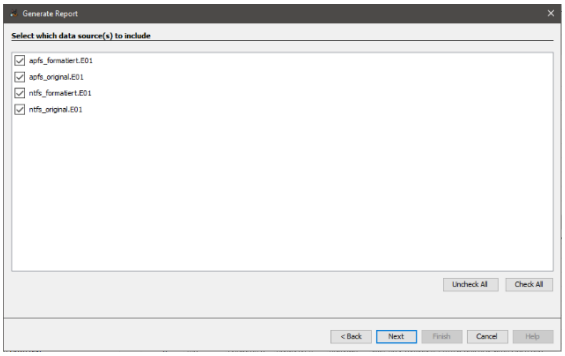


Bild 53: Auswahl der in den Report aufzunehmenden Asservate unter Autopsy

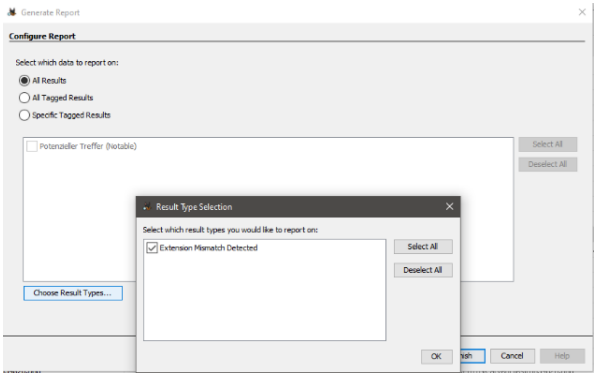


Bild 54: Auswahl der in den Report aufzunehmenden Untersuchungsergebnisse unter Autopsy

Mit dem darauffolgenden Klick auf „Finish“ wird die Report-Erstellung gestartet:

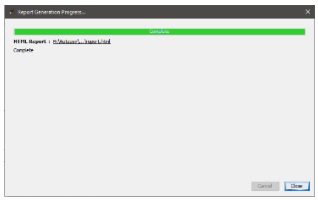


Bild 55: Fortschrittsstatus beim Erstellen eines Reports unter Autopsy

Erreichbar sind sie dann über das eigene Dateisystem oder über die Autopsy-Oberfläche unter dem Reiter „Reports“.

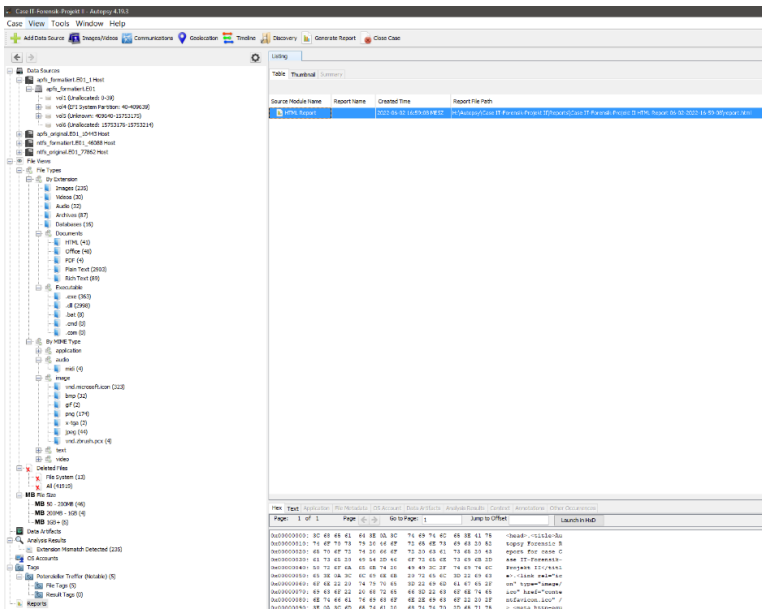


Bild 56: Zugriffsort auf einen erstellten Report innerhalb von Autopsy

Im HTML-Report sind auch die getaggten Bilder im Original und mit Markierungen enthalten, sie können direkt geöffnet werden:

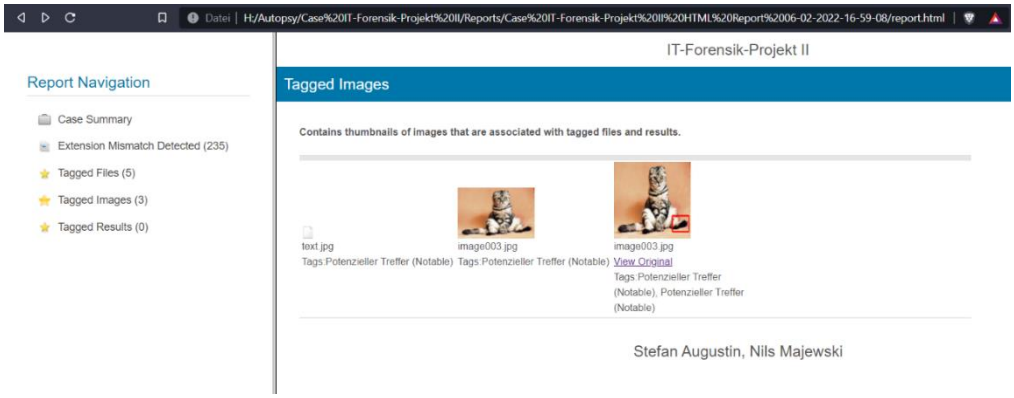


Bild 57: In einem Autopsy Report dargestellte Bilddateien

Andere Dateitypen wie bspw. Word müssen beim Aufruf erst heruntergeladen werden.

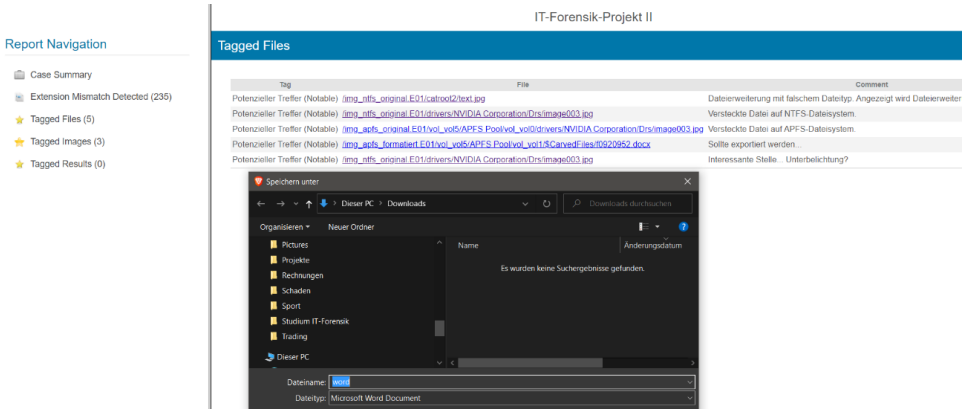


Bild 58: Herunterladen von Dateien aus einem Autopsy HTML Report zur späteren Ansicht durch ein entsprechendes Sichtungsprogramm

Beim Test konnte keine Fehlfunktion festgestellt werden, alle ausgewählten Dateien wurden korrekt angezeigt. Die fehlende PDF-Export-Funktionalität ist ein Kritikpunkt.

6.2 Axiom

Die im Test der gewählten Funktionen zu Grunde liegende Version von Magnet Forensics AXIOM ist 6.1.0.31400.

6.2.1 Konfiguration

Im Folgenden werden die getätigten Programmkonfigurationen für AXIOM dokumentiert, die zur Aufbereitung der Datenimages gewählt wurden. Jedes der vier Datenimages wurde mit exakt der gleichen Konfiguration bearbeitet, um vergleichbare Ergebnisse zu erzielen. Jedes Datenimage wurde einzeln aufbereitet.

Erstellen der Cases:

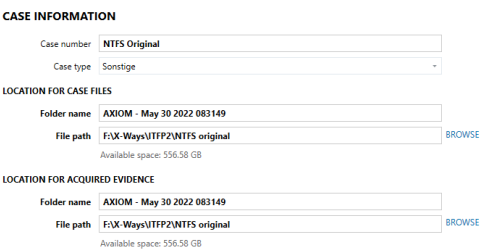


Bild 59: Festlegen des Speicherorts einer AXIOM Auswertung und deren Benennung

Hinzufügen der Datenimages:

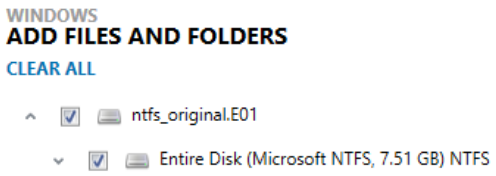


Bild 60: Hinzufügen eines .E01 Images unter AXIOM

Konfigurieren der Suchtiefe (Keine Bereiche aus der Suche ausgeschlossen):

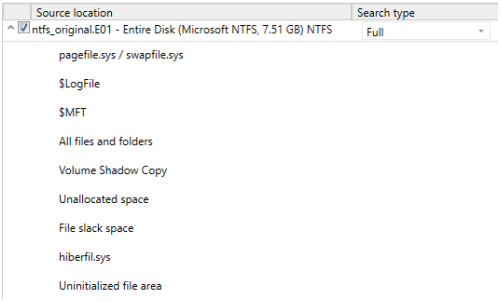


Bild 61: Auswahl der Suchtiefe unter AXIOM

Konfiguration der Artefaktsuche (Vollsuche, keine Artefakte ausgeschlossen):

Es wurden alle verfügbaren Aufbereitungsoptionen ausgewählt, da z. B. nicht klar ersichtlich war, welche der Optionen versteckte Dateien identifiziert.

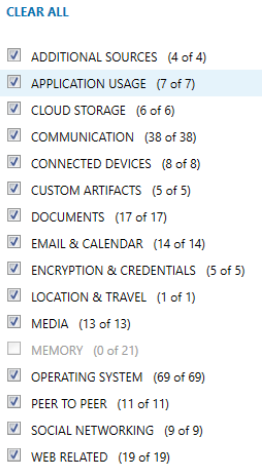


Bild 62: Auswahl der aufzubereitenden Artefakte unter AXIOM

Nach durchschnittlich 4 Minuten war die Aufbereitung der Images abgeschlossen. Die Ergebnisse wurden anschließend manuell ausgewertet.

Dies geschah einerseits über den Überblick aufbereiteter Artefakte:

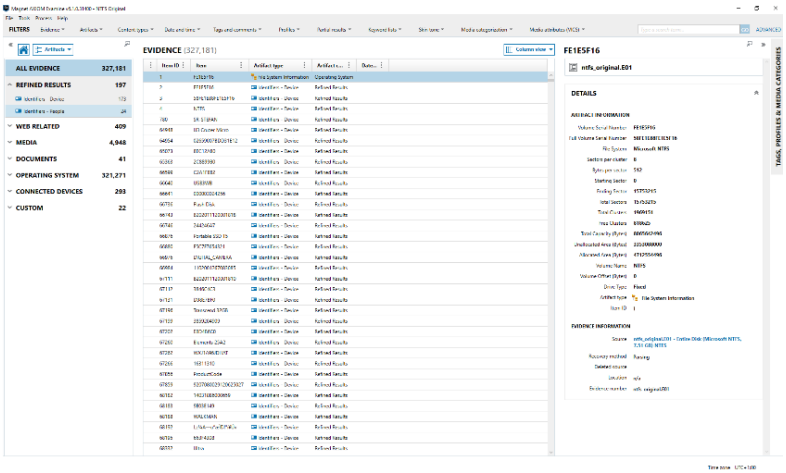


Bild 63: Artefaktansicht nach erfolgter Aufbereitung durch AXIOM

Als auch über die FileSystem Ansicht:

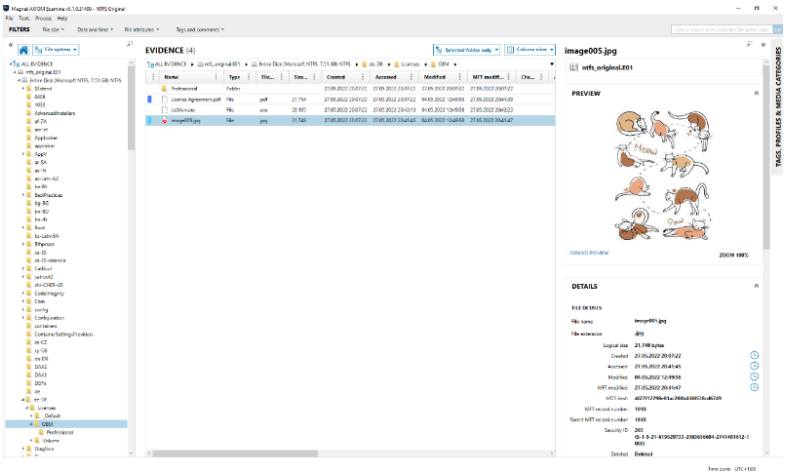


Bild 64: File System Ansicht unter AXIOM

6.2.2 Ergebnisse

In nachfolgender Tabelle werden die Auswertungsergebnisse dargestellt.

Wurde eine Datei mitsamt ihren Metadaten und den durchgeführten Manipulationen erkannt, wird dies als “OK” vermerkt. Bei Images von formatierten Datenträgern reicht das bloße Wiederherstellen um als Erfolg gewertet zu werden, da beim hier angewandten File Carving ohnehin nicht mit Metadaten gerechnet werden kann. Dieser Fall wird auch mit “OK” dokumentiert.

Auffälligkeiten oder Fehler werden als Kommentare erfasst.

Tabelle 7: Auswertungsergebnisse Axiom

Datei	Manipulation	APFS formatiert	APFS original	NTFS Formatiert	NTFS original
image001.jpg	Unverändert	OK	OK	OK	OK
image002.jpg	Gelöscht	OK	OK	OK	OK
image003.jpg	versteckt	OK	“versteckt” nicht erkannt	OK	OK
image004.jpg	umbenannt in "License Agreement.pdf"	OK	OK	OK	OK
image005.jpg	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
image006.jpg	umbenannt in "LicMan.exe" + versteckt	OK	“versteckt” nicht erkannt	OK	OK
image007.jpg	umbenannt in "Kratzbaum.ast" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
image008.jpg	umbenannt in "Default Log" + gelöscht	OK	OK	OK	OK
video001.mpg	Unverändert	OK	OK	OK	OK
video002.mpg	Gelöscht	OK	OK	OK	OK
video003.mpg	versteckt	OK	“versteckt” nicht erkannt	OK	OK
video004.mpg	umbenannt in "CV.pdf"	OK	OK	OK	OK
video005.mpg	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
video006.mpg	umbenannt in "Katzengras.plt" + versteckt	OK	“versteckt” nicht erkannt	OK	OK
video007.mpg	umbenannt in "Directory" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
video008.mpg	umbenannt in "walnut.vbs.mp4"	OK	OK	OK	OK

	+ gelöscht				
word1.docx	Unverändert	OK	OK	OK	OK
word2.docx	Gelöscht	OK	OK	OK	OK
word3.docx	versteckt	OK	“versteckt” nicht erkannt	OK	OK
word4.docx		OK	OK	OK	OK
word5.docx	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
word6.docx	umbenannt in "Inter-net.exe" + versteckt	OK	“versteckt” nicht erkannt	OK	OK
word7.docx	umbenannt in "Katzenfut-ter.lst" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK	“versteckt” nicht erkannt
word8.docx	umbenannt in "Grumpy.cat" + gelöscht	OK	OK	OK	OK

6.2.3 Analyse

Im Folgenden sollen, falls möglich, die im Test getroffenen Feststellungen analysiert werden.

6.2.3.1 Nicht erkanntes Verstecken von Dateien unter APFS

Für das Nichterkennen des "versteckt” Zustandes von Dateien unter APFS konnte keine Erklärung gefunden werden. Weitere Untersuchungen sind angeraten.

6.2.3.2 Nicht erkanntes Verstecken von Dateien in gelöschtem Zustand unter NTFS

Bei der manuellen Auswertung der AXIOM Aufbereitung konnte festgestellt werden, dass AXIOM den in der $MFT festgehaltenen Status “versteckt" bei der Anzeige der Dateiattribute nicht mehr berücksichtigt, sobald die Datei gelöscht wurde. Dies konnte bei jeder Datei, die die “versteck"+ "gelöscht” Kombination aufwies festgestellt werden.

Bild 65: Metadaten einer Bilddatei unter AXIOM

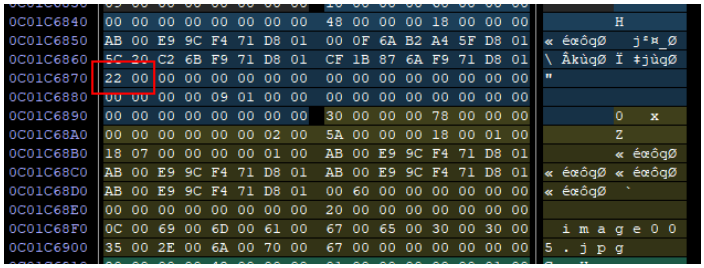


Bild 66: Dateiattribute "Archiv" und "Gelöscht" als Hexadezimalwerte 0x22 in einer $MFT Systemdatei

Da AXIOM Metadaten zu diesen gelöschten Dateien ausweist, musste es Zugriff auf die entsprechenden Einträge in der $MFT haben. Hierbei wurde jedoch offensichtlich der “versteckt” Status der Dateien ignoriert und nur “Archiv” übernommen.

6.2.3.3 Alleinstellungsmerkmal gegenüber Autopsy und X-Ways Forensics

Nach dem Aufbereiten der Daten können diese in einen "AXIOM Portable Case" exportiert werden, der durch seinen geringen Umfang auch auf durchschnittlichen Workstations durch einen Analysten gesichtet werden kann. AXIOM bietet mit seiner sehr intuitiven Oberfläche des "Artifact View" dabei auch nicht computeraffinen Anwendern die Möglichkeit aufbereitete Daten effizient zu sichten und relevante Inhalte zu markieren und exportieren.

6.2.4 Verhalten beim Hashwertabgleich

Obwohl die Betrachtung der Hashfunktionen der untersuchten Programme nicht Teil der zu prüfenden Programmfunktionen ist, soll das hier festgestellte Verhalten von AXIOM dennoch Erwähnung finden, da der Umgang mit Hashwerten für die forensische Arbeit von fundamentaler Bedeutung ist.

AXIOM bietet die Möglichkeit, Dateien bei der Aufbereitung anhand ihres Hashwertes zu identifizieren und diese nach Benutzerwünschen zu markieren. Diese Option wurde verwendet, um bei der manuellen Auswertung des Aufbereitungsergebnisses die Testdateien schnellst möglich aufzufinden.

Hierbei wurde festgestellt, dass AXIOM die Markierung bei vorhandenen und im Betriebssystem gelöschten Dateien wie gewünscht vornimmt:

Bild 67: Anzeige von erkannten Hashtreffern unter AXIOM

Sobald Dateien jedoch über File-Carving wiederhergestellt werden, erfolgt kein Hashwertabgleich, obwohl die Hashwerte der gecarvten Dateien berechnet werden und diese Hashwerte mit den Original-Hashwerten übereinstimmen.

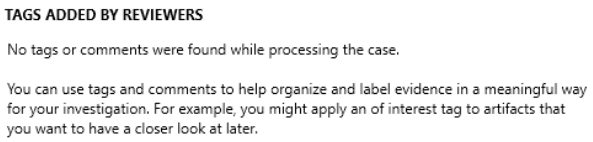


Bild 68: Anzeige bei keinen erkannten Hashtreffern unter AXIOM

AXIOM bietet zwar die Möglichkeit, nach Abschluss der Aufbereitung erneut einen Hashwertabgleich durchzuführen, womit die gecarvten Dateien gefunden und markiert werden können. Diese Funktion ist jedoch auf Bild- und Filmdateien beschränkt.

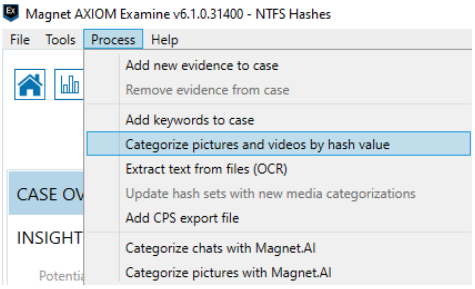


Bild 69: Menüauswahl zur nachträglichen Hashwertkategorisierung von Film- und Bilddateien unter AXIOM

Somit ist es nicht möglich, alle relevanten Dateien, die nur noch über File-Carving rekonstruiert werden konnten, automatisiert markieren zu lassen.

6.2.5 Erstellung eines Reports

AXIOM verfügt über eine Vielzahl von Funktionen, mit denen eine Dokumentation von Inhalten der Auswertung möglich ist:

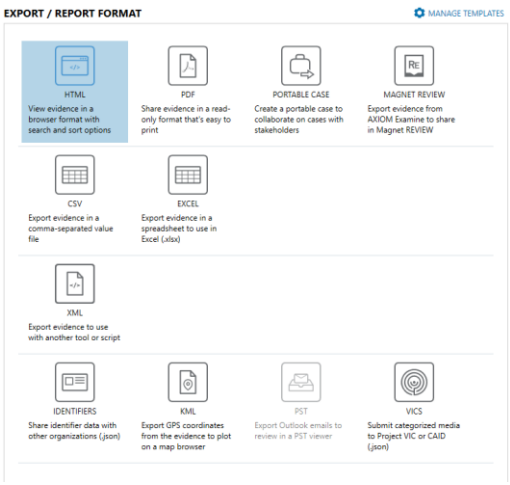


Bild 70: Auswahl des Report Formats unter AXIOM

Für die Evaluierung des Programms wurde die Dokumentation in Form einer PDF-Datei gewählt, da diese (neben des Portable Case) die nachträglich am wenigsten manipulierbare Option und den Standard beim Reporting darstellt. Bei dieser Art des Reportings werden die eingeschlossenen Dateien zusätzlich zum PDF-Bericht als Anlagen exportiert, um sie nativ betrachten zu können.

Folgende Optionen wurden gewählt und für jedes der vier Datenimages beibehalten:

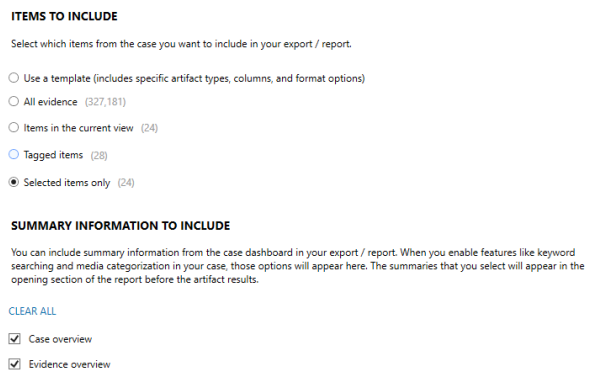


Bild 71: Auswahl der in den Report aufzunehmenden Artefakte unter AXIOM

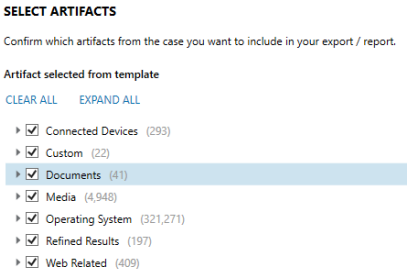


Bild 72: Auswahl der in den Report aufzunehmenden Artefakttypen unter AXIOM

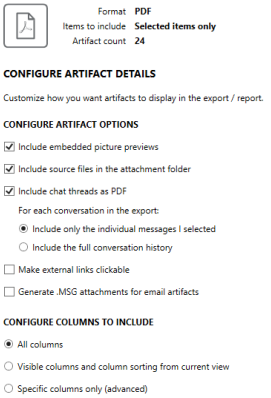


Bild 73: Konfiguration des Erscheinungsbilds eines PDF Reports unter AXIOM

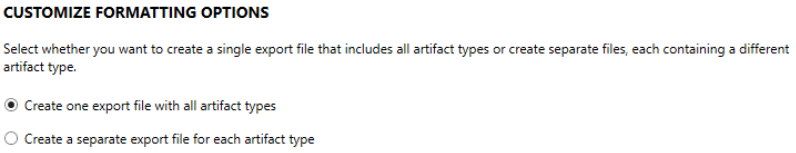
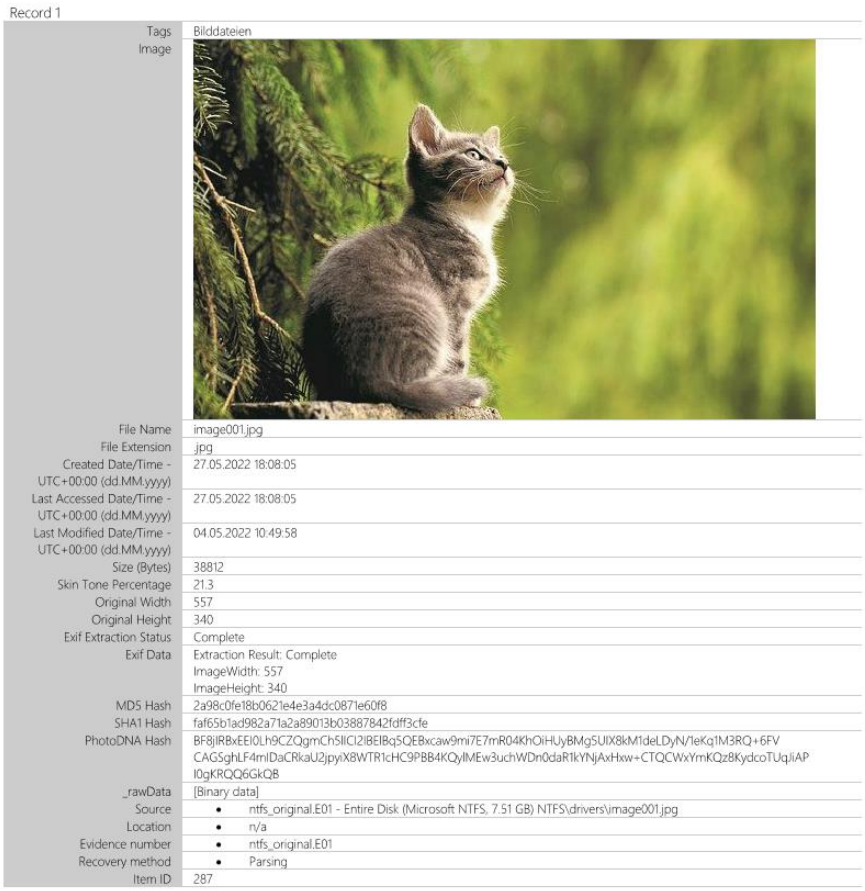


Bild 74: Konfiguration der Aufteilung eines PDF Reports unter AXIOM

Das Reporting der Dateien in das gewählte PDF Format zeigte die gleichen Schwächen auf, wie sie bereits bei der Auswertung im Programm selbst vorzufinden waren:

Während alle gewünschten Dateien in den Report aufgenommen wurden, waren deren Attribute (z. B. „versteckt“) nicht enthalten. Während man dies im Programm noch durch das Wechseln des Ansichtsmodus selbst eruieren kann, stellt der PDF Report keine derartige Funktion zur Verfügung. So ist das Erkennen von vom Benutzer versteckter Dateien im Report nicht möglich.



Record 1

Tags	Bilddateien
Image	
File Name	image001.jpg
File Extension	.jpg
Created Date/Time - UTC+00:00 (dd.MM.yyyy)	27.05.2022 18:08:05
Last Accessed Date/Time - UTC+00:00 (dd.MM.yyyy)	27.05.2022 18:08:05
Last Modified Date/Time - UTC+00:00 (dd.MM.yyyy)	04.05.2022 10:49:58
Size (Bytes)	38812
Skin Tone Percentage	21.3
Original Width	557
Original Height	340
Exif Extraction Status	Complete
Exif Data	Extraction Result: Complete ImageWidth: 557 ImageHeight: 340
MD5 Hash	2a98c0fe18b0621e4e3a4dc0871e60f8
SHA1 Hash	faf65b1ad982a71a2a89013b03887842fdff3cfe
PhotoDNA Hash	BF8jIRBxEEI0Lh9CZQgmChSIICI2IBEIBq5QEBxcaw9mi7E7mR04KhOiHUyBMg5UIX8kM1deLDyN/1eKq1M3RQ+6FV CAG5ghLF4mIDaCRkaU2jpyiX8WTR1cHC9PBB4KQyIMEw3uchWDn0daR1kYNjAxHxw+CTQCWxYmKQz8KydcoTUqJiAP I0gKRQQ6GkQB
_rawData	[Binary data]
Source	• ntfs_original.E01 - Entire Disk (Microsoft NTFS, 7.51 GB) NTFS\drivers\image001.jpg
Location	• n/a
Evidence number	• ntfs_original.E01
Recovery method	• Parsing
Item ID	287

Bild 75: Ansicht eines im Report aufgenommenen Artefakts unter AXIOM

Als besonders hinderlich wurde der Umstand gewertet, dass die als Anhang exportierten Dateien, trotz erkannter falscher Dateiendungen, mit diesen falschen Endungen exportiert wurden. So wurden etwa bei den Filmdateien die falschen Endungen .vbs und .pdf übernommen, was die Sichtung der Dateien erschwerte. Unbekannte oder verifiziert falsche Dateiendungen (z. B. .lst, .exe) wurden, ebenso durch .bin ersetzt:

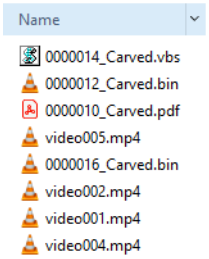


Bild 76: Mit dem Report exportierte Dateien unter AXIOM

6.3 X-Ways

Die im Test der gewählten Funktionen zu Grunde liegende Version von Magnet Forensics AXIOM ist 20.5 SR-1 x64.

Im Folgenden werden die getätigten Programmkonfigurationen für X-Ways Forensics dokumentiert, die zur Aufbereitung der Datenimages gewählt wurden. Jedes der vier Datenimages wurde mit exakt der gleichen Konfiguration bearbeitet um vergleichbare Ergebnisse zu erzielen. Jedes Datenimage wurde einzeln aufbereitet.

6.3.1 Konfiguration

Zur Erstellung der Cases wurden die Vorgaben übernommen. Lediglich die Verifizierung („Sicherung sofort überprüfen / Hash berechnen“) des einzulesenden Datenimages wurde aktiviert.

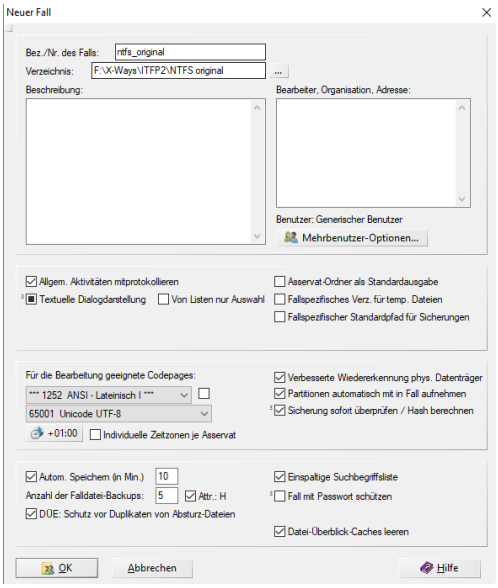


Bild 77: Erstellen eines neuen Falls unter X-Ways Forensics

Bei der Konfiguration zur Datenaufbereitung wurden ausschließlich Optionen aktiviert, die für die durchgeführten Dateimanipulationen relevant sind. Zusätzlich wurde die Option „Bildanalyse und –verarbeitung“ wurde aktiviert.

Um die Suche nach den Testdateien zu erleichtern wurden deren Hashwerte in die Hashwertdatenbank aufgenommen.

Diese Konfiguration wurde für alle vier zu untersuchenden Datenimages beibehalten.

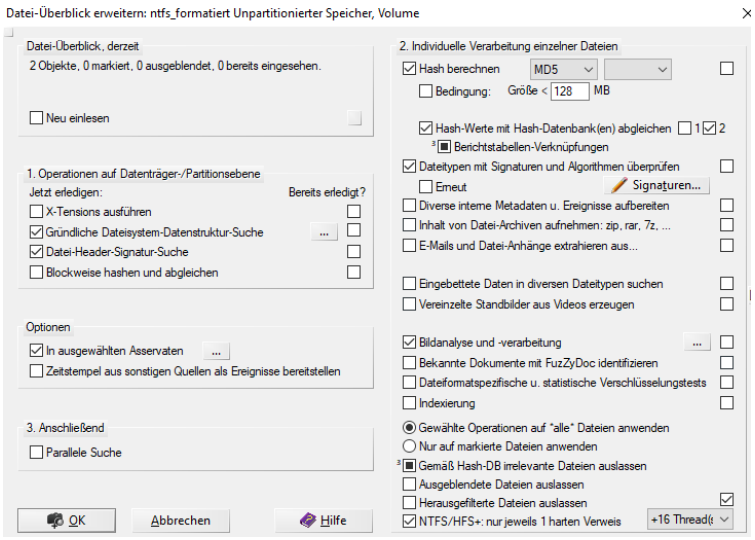


Bild 78: Auswahl der Aufbereitungsmodule unter X-Ways Forensics

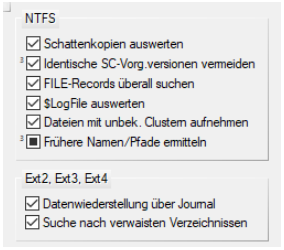


Bild 79: Feinkonfiguration zum Modul "Gründliche Dateisystem-Datenstruktur-Suche"

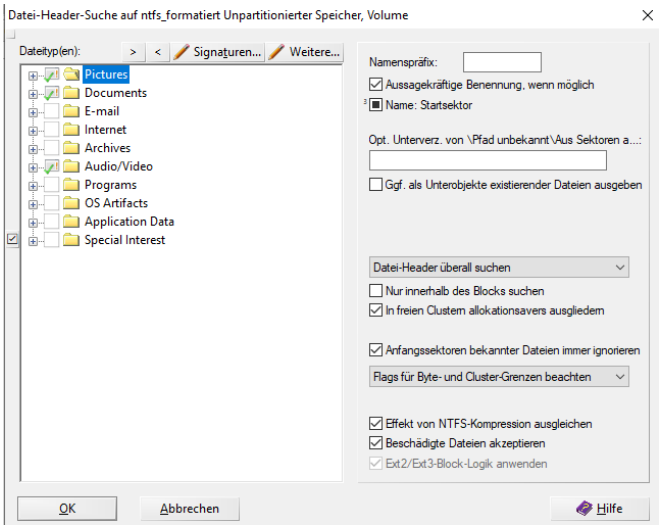


Bild 80: Auswahl der zu carvenden Dateitypen und Konfiguration des Carving-Vorgangs unter X-Ways Forensics

Nach durchschnittlich 15 Sekunden war die Aufbereitung der Images abgeschlossen. Die Ergebnisse wurden anschließend manuell ausgewertet.

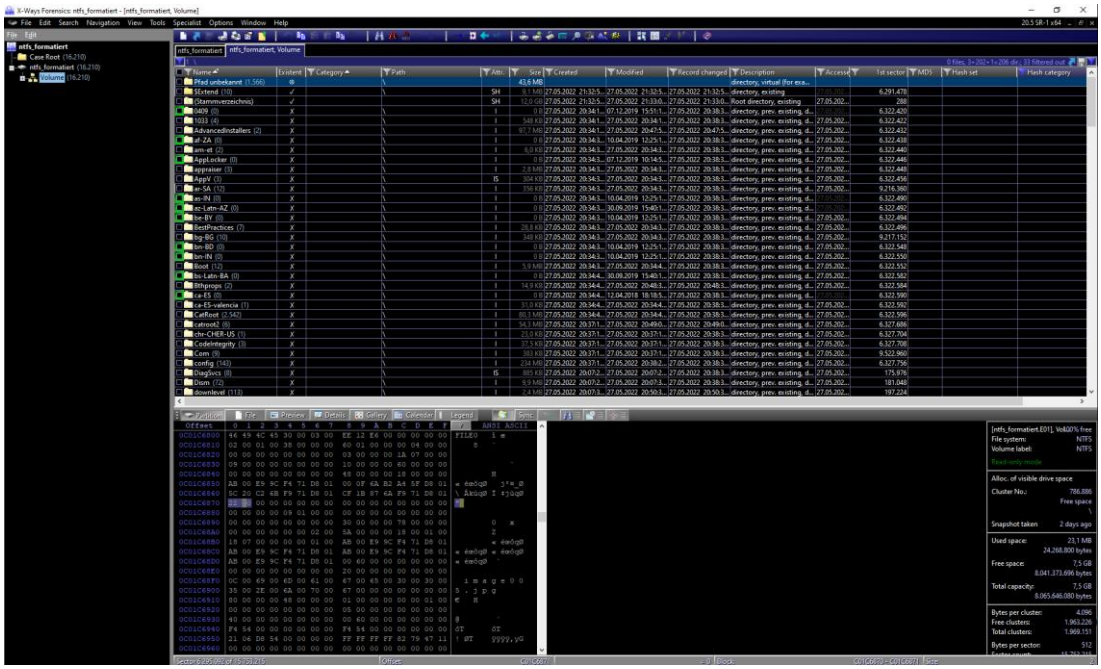


Bild 81: Programmoberfläche von X-Ways Forensics nach erfolgter Datenaufbereitung

6.3.2 Ergebnisse

In nachfolgender Tabelle werden die Auswertungsergebnisse dargestellt.

Wurde eine Datei mitsamt ihren Metadaten und den durchgeführten Manipulationen erkannt, wird dies als “OK” vermerkt. Bei Images von formatierten Datenträgern reicht das bloße Wiederherstellen um als Erfolg gewertet zu werden, da beim hier angewandten File Carving ohnehin nicht mit Metadaten gerechnet werden kann. Dieser Fall wird auch mit “OK” dokumentiert.

Auffälligkeiten oder Fehler werden als Kommentare erfasst.

Tabelle 8: Auswertungsergebnisse X-Ways

Datei	Manipulation	APFS formatiert	APFS original	NTFS Formatiert	NTFS original
image001.jpg	Unverändert	OK	OK	OK + alle Metadaten gecarvt	OK
image002.jpg	Gelöscht	OK	OK	OK + alle Metadaten gecarvt	OK
image003.jp	versteckt	OK	“versteckt”	OK + alle	OK

g			nicht erkannt	Metadaten gecarvt	
image004.jpg	umbenannt in "License Agree-ment.pdf"	OK	OK	OK + alle Metadaten gecarvt	OK
image005.jpg	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
image006.jpg	umbenannt in "Lic-Man.exe" + versteckt	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
image007.jpg	umbenannt in "Kratz-baum.ast" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
image008.jpg	umbenannt in "Default Log" + ge-löscht	Nicht gefun-den	OK	OK + alle Metadaten gecarvt	OK
vi-deo001.mpg	Unverändert	OK	OK	OK	OK
vi-deo002.mpg	Gelöscht	OK	OK	OK + alle Metadaten gecarvt	OK
vi-deo003.mpg	versteckt	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
vi-deo004.mpg	umbenannt in "CV.pdf"	OK	OK	OK + alle Metadaten gecarvt	OK
vi-deo005.mpg	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
vi-deo006.mpg	umbenannt in "Katzen-gras.plt" + versteckt	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
vi-deo007.mpg	umbenannt in "Directory" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
vi-deo008.mpg	umbenannt in "wal-nut.vbs.mp4" + gelöscht	OK	OK	OK + alle Metadaten gecarvt	OK
word1.docx	Unverändert	OK	OK	OK	OK
word2.docx	Gelöscht	OK	OK	OK + alle	OK

				Metadaten gecarvt	
word3.docx	versteckt	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
word4.docx		OK	OK	OK + alle Metadaten gecarvt	OK
word5.docx	versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
word6.docx	umbenannt in "Inter-net.exe" + versteckt	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
word7.docx	umbenannt in "Katzenfut-ter.lst" + versteckt + gelöscht	OK	“versteckt” nicht erkannt	OK + alle Metadaten gecarvt	OK
word8.docx	umbenannt in "Grumpy.cat" + gelöscht	OK	OK	OK + alle Metadaten gecarvt	OK

6.3.3 Analyse

Im Folgenden sollen, falls möglich, die im Test getroffenen Feststellungen analysiert werden.

6.3.3.1 Durch File Carving nicht aufgefundene Bilddatei unter APFS formatiert

Die Bilddatei “image008.jpg” (als Testdatei umbenannt in "Default Log”) konnte beim formatierten APFS Datenimage nicht durch File Carving rekonstruiert werden. Das manuelle Suchen nach der Datei ergab, dass sie sich mit intaktem Header und Footer auf dem Datenimage befindet und so aufgefunden hätte werden müssen.

6.3.3.2 Nicht erkanntes verstecken von Dateien unter APFS

Für das Nichterkennen des "versteckt” Zustandes von Dateien unter APFS konnte keine Erklärung gefunden werden.

6.3.3.3 Metadaten beim File-Carving unter NTFS

X-Ways Forensics war es beim File-Carving unter NTFS gelungen, sämtliche Metadaten aller Testdateien zu erhalten, obwohl diese aufgrund der Natur des File Carvings nicht verfügbar sein sollten.

Verantwortlich hierfür ist eine Konfigurationsmöglichkeit der Funktion zur Untersuchung des vorhandenen Dateisystems:

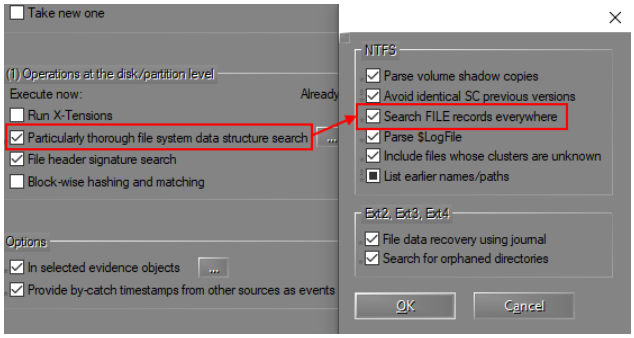


Bild 82: Menüstruktur zur "Gründlichen Dateisystem-Datenstruktur-Suche"

Hierdurch sucht X-Ways Forensics nicht nur in der aktuellen $MFT, sondern auf dem gesamten Datenspeicher nach FILE Einträgen und kann somit auch Daten von vergangenen $MFT nutzbar machen.

6.3.3.4 Alleinstellungsmerkmal gegenüber Autopsy und AXIOM

Gegenüber der beiden anderen Programmen zeichnet sich X-Ways Forensics vor allem durch seine geringe Größe (Kernsoftware in der Version 20.5-SR1 unter 50MB) und die Möglichkeit, portabel eingesetzt zu werden aus.

Durch seine Grundlagen als Hexeditor bietet X-Ways Forensics zusätzlich die Möglichkeit, Daten bis auf Byteebene zu untersuchen, was bei der digitalforensischen Arbeit oft ein Vorteil ist, jedoch nur durch erfahrene Forensiker voll ausgereizt werden kann. Dies gilt auch für die vielfältigen Filter- und Datenaufbereitungsoptionen, die X-Ways Forensics mitbringt.

6.3.4 Erstellen eines Reports

Zum Reporting der Testdateien wurde die vorgegebene Konfiguration verwendet. Von allen verfügbaren Informationen, die im Report aufgenommen werden können, wurden so nur wenige übernommen.

Da die Standardeinstellung das Erstellen eines HTML-Reports vorsah, wurde dies in die Erstellung eines PDF-Reports abgeändert.

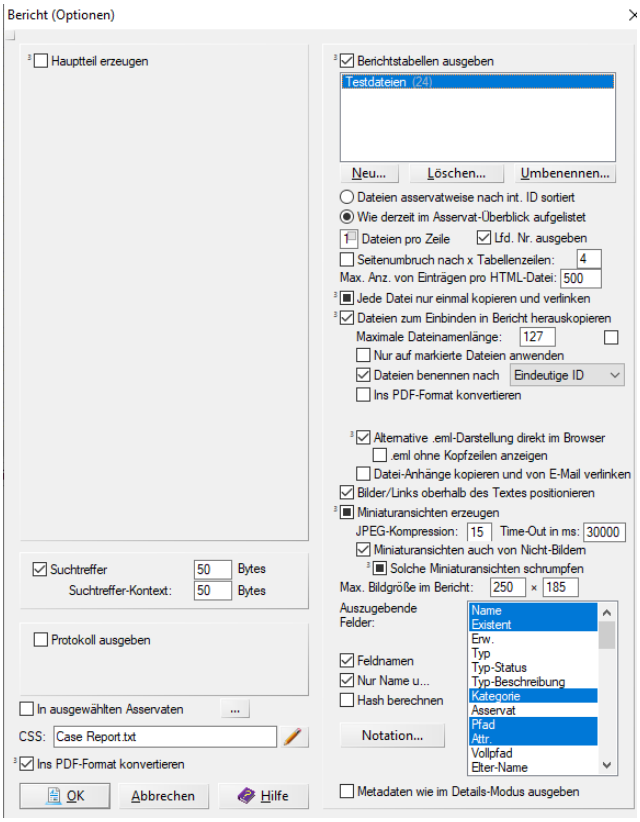


Bild 83: Menü zur Konfiguration eines Reports unter X-Ways Forensics

Die Reportingeinstellungen wurden für jedes der vier Datenimages beibehalten.

In die beim Reporting erstellte PDF Datei wurden sämtliche erkannten Manipulationen an den Testdateien fehlerfrei übernommen.

Testdateien (24 Einträge)

1

Name: Default Log
Existent: Nein
Kategorie: Pictures
Pfad: \LogFiles\CloudFiles
Attr.: A
MD5: 65371E22C6371FD70CD9C8373C39563F

Bild 84: Ansicht eines im Report aufgeführten Artefakts unter X-Ways Forensics

7 Gegenüberstellung, Bewertung und Vergleich

7.1 Analyse der Auswertungsergebnisse (Kriterienkatalog)

In folgendem eingebetteten Excel-Dokument ist der aus den Vergleichsrubriken in Kapitel 5.3 Festlegen der Vergleichskriterien erstellte Kriterienkatalog zu finden. Die Vergleichskriterien wurden in der Phase „Planung der Durchführung“ ausgewählt. In dieser Phase („Gegenüberstellung, Bewertung und Vergleich“) ist lediglich die Bewertung gem. Bewertungsmatrix erfolgt.

1. Platz: X-Ways

2. Platz: Autopsy

3. Platz: Axiom

7.2 Performance

Folgende Tabelle gibt eine Übersicht über die benötigte Zeit bei der automatischen Auswertung mittels Labor-Rechner. Bei allen Tools wurde eine möglichst ähnliche Konfiguration gewählt. Für weitere Details zur Konfiguration siehe jeweilige Unterkapitel in Kapitel 6 Durchführung der Auswertungen. Die Systemspezifikation ist in Kapitel 4.5 Beschreibung Systemspezifikationen zu finden.

Tabelle 9: Laufzeiten der Auswertungen (Performancevergleich)

Asservat	Axiom (in min.)	X-Ways (in min.)	Autopsy (in min.)
apfs_formatiert.E01	3:26	0:15	4:18
apfs_original.E01	5:16	0:20	2:51
ntfs_formatiert.E01	3:22	0:26	1:48
ntfs_original.E01	4:02	0:15	1:18

8 Fazit

Die große Überraschung ist Autopsy, das die beiden großen Player schon fast ausstechen konnte. Bis auf die nicht auffindbaren Videodateien konnte es im Gegensatz zu X-Ways und AXOM alle Manipulationen nachweisen. Alleinstellungsmerkmal von Autopsy ist bspw. auch die Möglichkeit Bereiche in Bilddateien zu markieren. Im Gegensatz zu NTFS, besitzt APFS viele forensisch interessante Artefakte wie bspw. Checkpoints/Versionierung oder die Copy-on-Write-Funktion. Eine weitere Überraschung war der hohe Aufwand, den die Evaluierung mit sich brachte. Trotz der wenigen Testziele und der überschaubaren Anzahl an Testdateien, war vor allem die Erstellung der Testimages und die manuelle Auswertung der Ergebnisse ein richtiger Zeitfresser. Nicht umsonst heißt es, dass Testen eine endlose Aufgabe ist. Da auf Basis der Ergebnisse dieser Programme Gerichtsurteile gefällt werden, sollten sie maximal zuverlässig funktionieren. Daher erachten wir die Durchführung einer solchen Evaluation für sinnvoll. Die schon während unseres kleinen Projekts gefundenen Fehler sprechen für sich.

8.1 Kritik und Verbesserungsvorschläge

Neben der Behebung der gefundenen Fehler in Kapitel 6 Durchführung der Auswertungen nachfolgend einige Verbesserungsvorschläge je Tool.

Autopsy überzeugte bei der Auswertung durch das Feststellen von Spuren, die keines der beiden anderen Programme entdecken konnte. Die manuelle Auswertung wurde jedoch dadurch verlangsamt, dass nicht alle gefundenen Dateierweiterungen unter dem Anzeigepunkt “By Extension” eingruppiert wurden. Hier wäre neben den vorhandenen Gruppen, wie z. B. “Images” noch eine Gruppierung “Other” hilfreich, in der alle Dateien mit bisher nicht berücksichtigter Dateiendung fallen. Auch das flüssige Abspielen von Videos stellte ein Problem dar, das eine Lösung finden sollte. Außerdem wäre eine Gruppierungsfunktion nach Hashes für die manuelle Auswertung hilfreich, wenn eine Datei mehrfach bspw. via MFT und File Carving gefunden wird. Eine PDF-Export-Funktion für die erstellten Reports ist wünschenswert. Außerdem sollten auch weitere Metadaten der Dateien wie bspw. die Datei-Flags im Report hinzufügbar sein.

Bei AXIOM wurde die Evaluierung der Programmfunktionen dadurch erschwert, dass wichtige Informationen, wie etwa die Dateiattribute (“Versteckt”, “Archiv”, usw.) erst durch den Wechsel der Ansicht in den “System View” einzusehen waren. Dies ist zwar mit nur einem Klick erreichbar, beim Überprüfen mehrerer Dateien jedoch sehr hinderlich, da hierfür ständig zwischen den Ansichten gewechselt werden muss. Wichtige Informationen, wie die Dateiattribute sollten auch bereits im “Artifact View” dargestellt werden, wie es andere Metadaten, z.B. Zeitstempel, bereits werden. Gleiches gilt für das Aufnehmen dieser Informationen in die generierten Reports, wo sie bislang fehlen.

Bei X-Ways Forensics war die Berichtserstellung mit einigem Trial-and-Error verbunden, bis es zu einem korrekt formatierten Report kam. Das lag daran, dass die eingebetteten Vorschaubilder der Dateien in nicht einheitlicher Größe dargestellt wurden, was teilweise dazu führte, dass Vorschaubilder und Dateibeschreibungen (z.B. Speicherpfad, Hashwert, usw.) durch einen Seitenumbruch getrennt wurden. Es war zwar möglich, dies durch Konfiguration des Reports (Dateien pro Zeile und Dateien pro Reihe) auszugleichen, eine automatisierte Größenanpassung würde die Reportgenerierung jedoch erleichtern.

8.2 Lessons Learned

Testen ist eine schier endlose Aufgabe, daher ist es wichtig sich im Vorfeld einen genauen Testplan zu erstellen und konkrete Testziele zu definieren. Testdaten müssen sorgfältig ausgewählt und auf die Testziele abgestimmt werden, um einen möglichst effizienten Test zu gewährleisten. Außerdem ist hinreichend Zeit einzuplanen. Bei den Testdaten muss darauf geachtet werden, dass mit möglichst wenig Datensätzen, möglichst viele Verhaltensweisen abgedeckt werden. Mehrere Dateien, welche dieselben Funktionalitäten prüfen, vergrößern nur den Dokumentationsaufwand. Eine Orientierung an internationalen Standards wie die ISO/IEC/IEEE 29119: Software Testing. oder die Normreihe ISO/IEC 25000: System and Software Quality Requirements and Evaluation sollte im Vorfeld eines Tests geprüft werden.

8.3 Ausblick

Aufgrund des Projektergebnisses und der gefundenen Fehler sind weitere Tests zu begrüßen, in denen zusätzliche Funktionen überprüft werden und andere Dateisysteme und Dateitypen mit einbezogen werden. Aber auch eine tiefergehende Betrachtung von NTFS und APFS lohnt sich, da diese nur rudimentär getestet wurden. Zu nennen ist hier bspw. die Versionierungsfunktionalität von APFS. Auch das Testen von Dateien mit eingebetteten Dokumenten, Archiven, der Umgang mit Verschlüsselung sind spannende Themen. Außerdem werden die gefundenen Bugs zur Fehlerbehebung an die Hersteller gemeldet.

9 Anlagen

9.1 Dokumentation der Testdateien

Tabelle 10: Testdateien

Dateiname	Speicherpfad	Manipulation
image001.jpg	/drivers/	unverändert
image002.jpg	/drivers/	gelöscht
image003.jpg	/drivers/NVIDIA Corporation/Drs/	versteckt
image004.jpg	/de-DE/Licenses/OEM/	umbenannt in “License Agreement.pdf”
image005.jpg	/de-DE/Licenses/OEM/	versteckt + gelöscht
image006.jpg	/de-DE/Licenses/OEM/	umbenannt in “LicMan.exe” + versteckt
image007.jpg	/DriverStore/FileRepository/	umbenannt in “Kratzbaum.ast” + versteckt + gelöscht
image008.jpg	/LogFiles/CloudFiles/	umbenannt in “Default Log” + gelöscht
video001.mp4	/networklist/icons/	unverändert
video002.mp4	/SecureBootUpdates/	gelöscht
video003.mp4	/winevt/Logs/	versteckt
video004.mp4	/Keywords/	umbenannt in “CV.pdf”
video005.mp4	/Recovery/	versteckt + gelöscht
video006.mp4	/Tasks_Migrated/Mozilla/	umbenannt in “Katzengras.plt” + versteckt
video007.mp4	/wbem/Framework/root/	umbenannt in “Directory” + versteckt + gelöscht
video008.mp4	/AdvancedInstallers/	umbenannt in “walnut.vbs.mp4” + gelöscht
word1.docx	/BestPractices/v1.0/	unverändert
word2.docx	/Bthprops/	gelöscht
word3.docx	/AppV/	versteckt

word4.docx	/catroot2/	umbenannt in “text.jpg”
word5.docx	/CodeIntegrity/Tokens/Active/	versteckt + gelöscht
word6.docx	/config/Journal/	umbenannt in “Internet.exe” + versteckt
word7.docx	/downlevel/	umbenannt in “Katzenfutter.lst” + versteckt + gelöscht
word8.docx	/ProximityToast/	umbenannt in “Grumpy.cat” + gelöscht

9.2 Beschreibung Ingest Module am Beispiel Autopsy [4]

Eine ausführliche Beschreibung ist auch in der Autopsy-Dokumentation zu finden: http://sleuthkit.org/autopsy/docs/user-docs/4.19.3/ingest_page.html:

9.2.1 Hash Lookup

Mit diesem Modul werden MD5 und SHA256 Hashes der gefundenen Dateien berechnet. Sie können auch gegen eine interne Datenbank mit gespeicherten Hashes gehalten werden, um hiermit die gefundenen Dateien auf die in der Datenbank enthaltenen Hashes zu filtern. Durch diesen Hashwertabgleich können etwa unwichtige Dateien identifiziert (whitelisting) oder relevante Dateien sofort erkannt werden. Dieses Modul wird in der Default-Konfiguration verwendet.

9.2.2 File Type Identification Module

Dieses Modul ist verantwortlich für das Aufspüren von Dateien anhand interner Signaturen. Optional dem Dateinamen hinzugefügte Dateierweiterungen (bspw. .pdf) werden nicht betrachtet.

Die Funktionalität stammt dabei von Apache Tika, ein Drittanbieterbibliothek zur Identifizierung und Extraktion von Metadaten und Text unzähliger Dateiformate: https://tika.apache.org/

Dieses Modul bzw. seine Ergebnisse werden von vielen weiteren Modulen wie bspw. dem Extension Mismatch Detector Module verwendet und sollte stets aktiviert sein.

Aufgrund der Nutzung von Apache Tika müssen keinerlei Konfigurationen vorgenommen werden, bei Bedarf können allerdings eigene Dateiformate inkl. ihrer Signatur hinzugefügt werden:

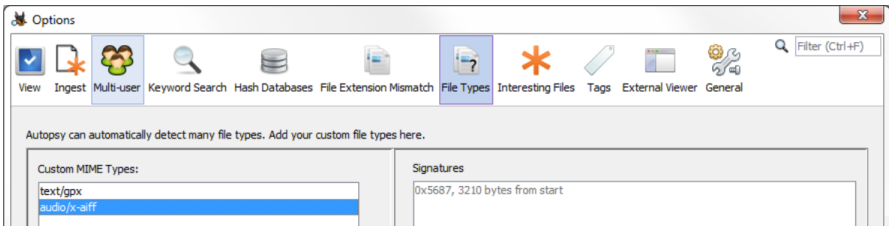


Bild 85: File Type Ansicht unter den Optionen von Autopsy

Angezeigt werden die Ergebnisse dann in Autopsy unter: Views → File Types → By MIME Type

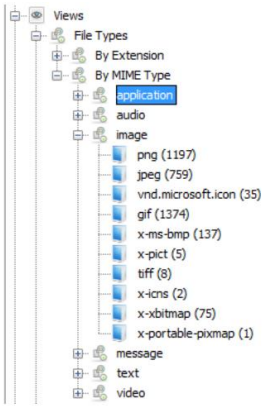


Bild 86: Nach MIME Typ sortierte Dateien in Autopsy

Für die Auswertung wurde die Default-Konfiguration verwendet.

9.2.3 Extension Mismatch Detector Module

Dieses Modul nutzt die Ergebnisse des File Type Identification Module und markiert Dateien, bei denen die Dateierweiterung nicht zum tatsächlichen Dateityp (gem. Signatur) passt. Dies ist ein Hinweis darauf, dass jemand die Datei verstecken wollte. Die Default-Einstellungen:

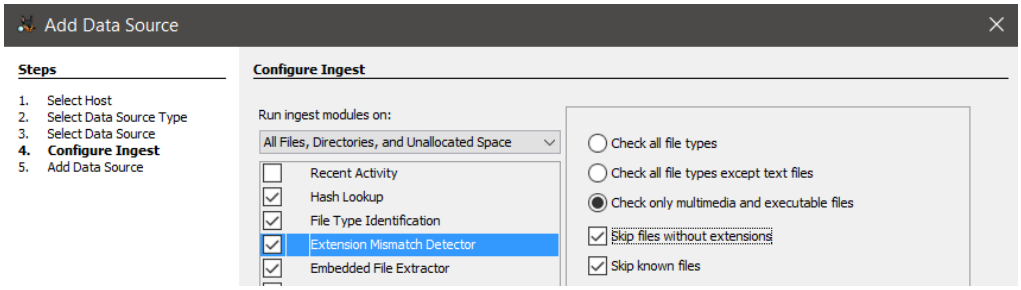


Bild 87: Konfiguration der Extension Mismatch Erkennung unter Autopsy

wurden mit folgender Konfiguration überschrieben:

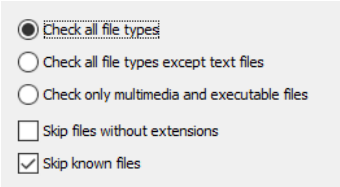


Bild 88: Feinkonfiguration der Extension Mismatch Erkennung unter Autopsy

An den Dateitypen und erlaubten Erweiterungen wurde nichts geändert.

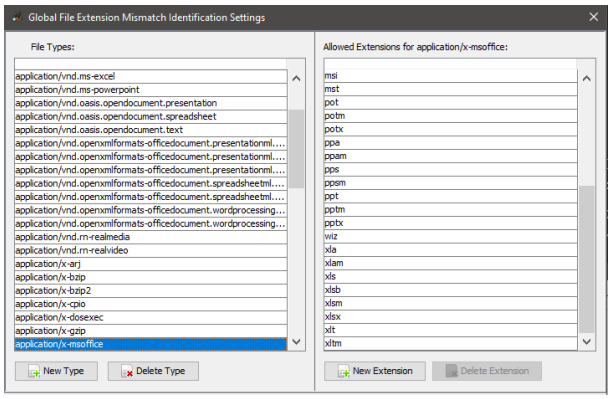


Bild 89: Globale Einstellungen zur Extension Mismatch Erkennung unter Autopsy

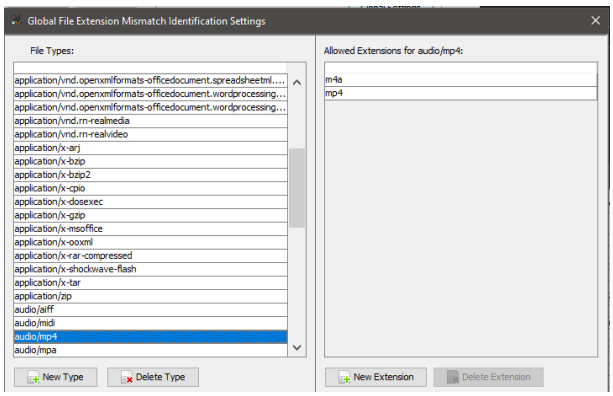


Bild 90: Globale Einstellungen zur Extension Mismatch Erkennung unter Autopsy

Die Ergebnisse werden auf der Oberfläche unter dem Punkt „Extension Mismatch Detected“ angezeigt.

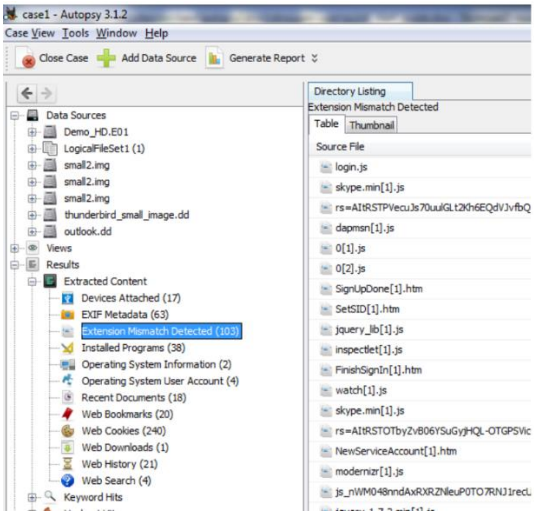


Bild 91: Erkannte Extension Mismatches im Auswertungsergebnis unter Autopsy

9.2.4 Embedded File Extractor Module

Zwar nicht Bestandteil dieser Projektarbeit, dennoch erwähnenswert ist das Modul zum Auswerten von eingebetteten Dateien in unterschiedlichsten Archiv-Formaten wie bspw. .zip, .doc, .docx, .xlsx. Das Modul besitzt keinerlei Konfigurationsmöglichkeiten.

In der Dokumentation ist ein nicht näher konkretisierter Hinweis zu finden, dass es Einschränkungen bei der Extraktion von Dateien innerhalb MS-Office-Dateitypen wie .docx, .xlsx gibt.

Archive werden auf der Oberfläche angezeigt unter: View → File Types → Archives

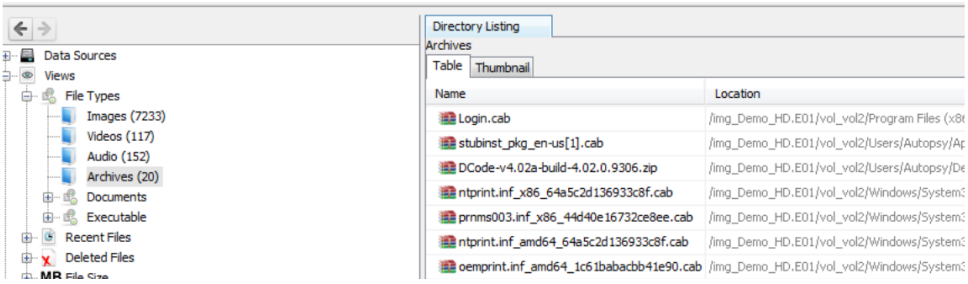


Bild 92: Erkannte Dateiarchive im Auswertungsergebnis unter Autopsy

Die extrahierten Dateien sind auf der Oberfläche unter der jeweiligen Datenquelle zu finden:

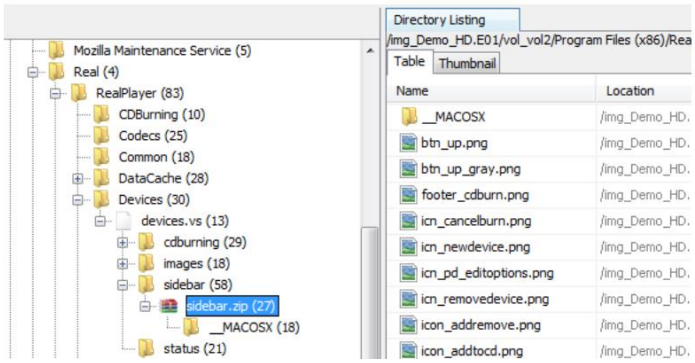


Bild 93: Extrahierte Archivinhalte im Auswertungsergebnis unter Autopsy

9.2.5 Picture Analyzer Module

Mit diesem Modul können Exchangeable Image File Format Informationen wie bspw. Geopositions-Daten oder Kameratyp von Bildern extrahiert werden. Auch dieses Modul besitzt keine Konfigurationsmöglichkeiten.

Die gefundenen Informationen sind im Ergebnis-Menüpunkt unter „EXIF Metadata“ zu finden:

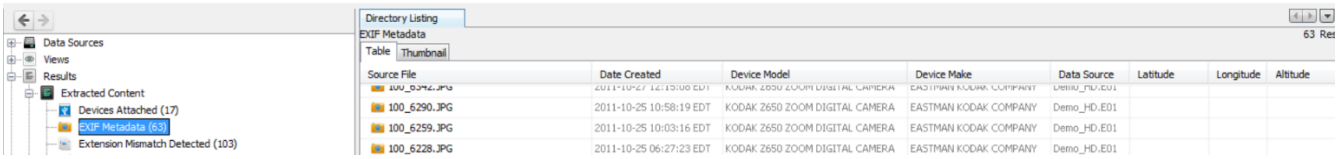


Bild 94: Darstellung von EXIF Metadaten im Auswertungsergebnis unter Autopsy

9.2.6 Interesting Files Identifier

Dieses Modul ist zwar nicht für diese Projektarbeit relevant, allerdings dennoch erwähnenswert. Mit diesem Modul können gem. hinterlegter Regeln Dateien und Verzeichnisse automatisch mit einem Flag markieren. Bspw. können alle Dateien markiert werden, welche einen bestimmten Namen oder Dateityp haben. Auch Wallets für Kryptowährungen können so markiert werden, diese Regeln sind per Default aktiviert:

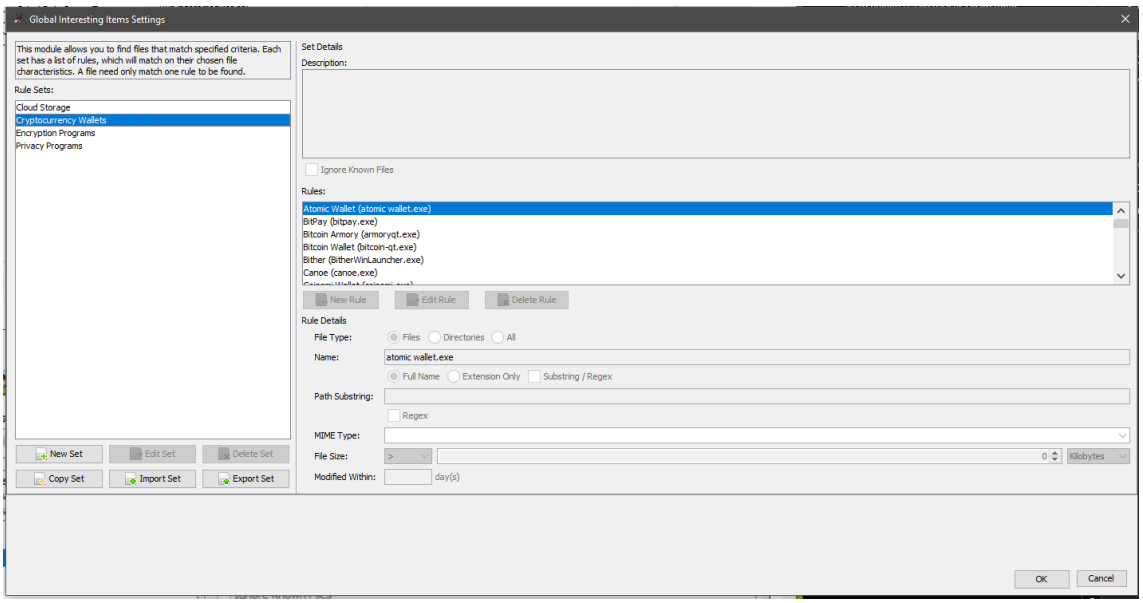


Bild 95: Konfigurationsmöglichkeiten für Dateien von Interesse unter Autopsy

Im folgenden Beispiel würden alle Dateien markiert werden, die den Substring „bomb“ im Namen tragen sowie dem Dateityp .png besitzen.

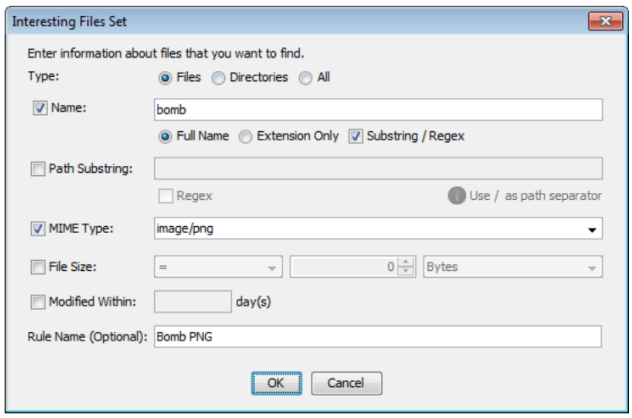


Bild 96: Feinkonfiguration für Dateien von Interesse unter Autopsy

Die Ergebnisse werden auf der Autopsy-Oberfläche unter Results → Interesting Items → jeweiliges Flag angezeigt:

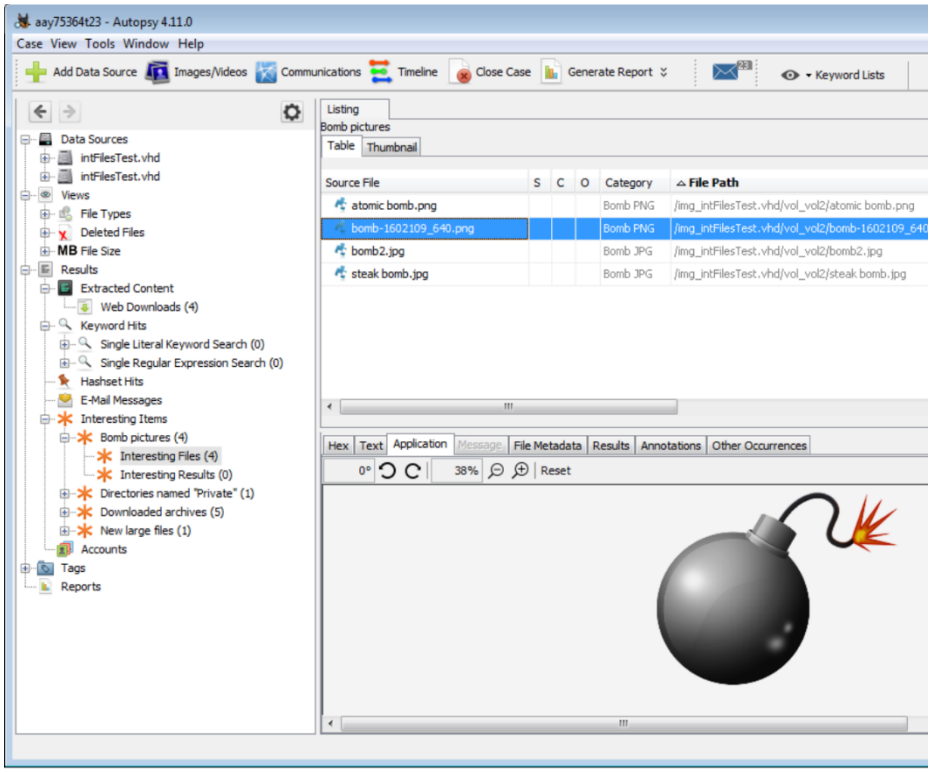


Bild 97: Anzeige von Dateien von Interesse im Auswertungsergebnis von Autopsy

9.2.7 PhotoRec Carver Module

Dieses Modul verwendet das Drittanbieterprogramm PhotoRec und wertet den Unallocated-Space via File Carving aus. Hierdurch können gelöschte Dateien gem. ihrer Dateisignaturen wiederhergestellt werden. Auch dieses Modul besitz nahezu keine Konfigurationsmöglichkeiten. Für die Auswertung wird die Default-Konfiguration verwendet:

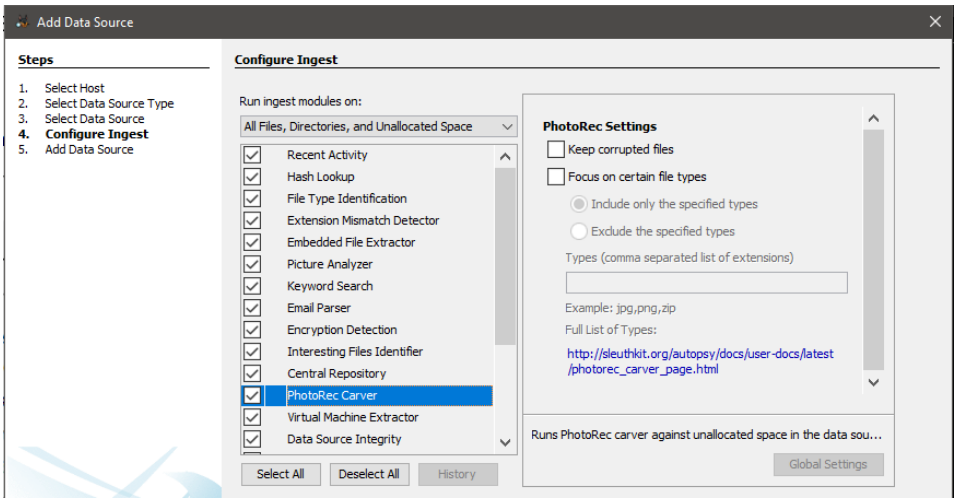


Bild 98: Auswahl des PhotoRec File Carvers bei der Aufbereitungskonfiguration von Autopsy

Folgende Dateitypen werden in der verwendeten Version unterstützt:

Valid File Types

The following is the list of valid file types for the version of PhotoRec currently used by Autopsy:

1cd	caf	dwg	gp2	max	pdb	rw2	vfb
3dm	cam	dxf	gp5	mb	pdf	rx2	vib
7z	catdrawing	e01	gpg	mcd	pds	sav	vmdk
a	cdt	eCryptfs	gpx	mdb	pf	save	vmg
ab	che	edb	gsm	mdf	pfx	ses	wallet
abr	chm	elf	gz	mfa	plist	sgcta	wdp
acb	class	emf	hdf	mfg	plr	shn	wee
accdb	comicdoc	ess	hdr	mft	plt	sib	wim
ace	cow	evt	hds	mid	png	sit	win
ado	cp_	evtx	hfsp	mig	pnm	skd	wks
afdesign	cpi	exe	hm	mk5	prc	skp	wld
ahn	crw	exs	hr9	mkv	prd	snag	wmf
aif	csh	ext	http	mlv	prt	snz	wnk
all	ctg	fat	ibd	mobi	ps	sp3	woff
als	cwk	fbf	icc	mov	psb	sparseimage	wpb
amd	d2s	fbk	icns	mov/mdat	psd	spe	wpd
amr	dad	fcp	ico	mp3	psf	spf	wtv
apa	dar	fcs	idx	mpg	psp	sqlite	wv
ape	dat	fdb	ifo	mpl	pst	sqm	x3f
apple	DB	fds	imb	mrw	ptb	steuer2014	x3i
ari	db	fh10	indd	msa	ptf	stl	x4a
arj	dbf	fh5	info	mus	pyc	studio	xar
asf	dbn	fit	iso	mxf	pzf	swf	xcf
asl	dcm	fits	it	MYI	pzh	tar	xfi
asm	ddf	flac	itu	myo	qbb	tax	xfs
atd	dex	flp	jks	nd2	qdf	tg	xm
au	diskimage	flv	jpg	nds	qkt	tib	xml
axp	djv	fm	jsonlz4	nes	qxd	tif	xpt
axx	dmp	fob	kdb	njx	r3d	TiVo	xsv
bac	doc	fos	kdbx	nk2	ra	torrent	xv
bdm	dpx	fp5	key	nsf	raf	tph	xz
bim	drw	fp7	ldf	oci	rar	tpl	z2d
bin	ds2	freeway	lit	ogg	raw	ts	zcode
binvox	DS_Store	frm	lnk	one	rdc	ttf	zip
bkf	dsc	fs	logic	orf	reg	tx?	zpr
blend	dss	fwd	lso	paf	res	txt	
bmp	dst	gam	luks	pap	rfp	tz	
bpg	dta	gct	lxo	par2	riff	v2i	
bvr	dump	gho	lzh	pcap	rlv	vault	
bz2	dv	gi	lzo	pcb	rm	vdi	
c4d	dvi	gif	m2ts	pct	rns	vdj	
cab	dvr	gm*	mat	pcx	rpm	veg	

Bild 99: Von PhotoRec standardmäßig unterstützte Dateitypen unter Autopsy

Es ist auch möglich eigene Dateisignaturen hinzuzufügen. Hierfür muss eine „photorec.sig“ unter folgendem Pfad abgelegt werden: /home/john/ (Linux), C:\Users\john\ (Windows)

Pro Zeile kann eine Dateisignatur angegeben werden. Im folgenden Beispiel wird der Dateityp .bar mit dem Offset 0 und der Signatur 0x4141414141414141 hinzugefügt:

```
bar 0 0x4141414141414141
```

Bild 100: Hinzufügen einer neuen Dateisignatur in Autopsy

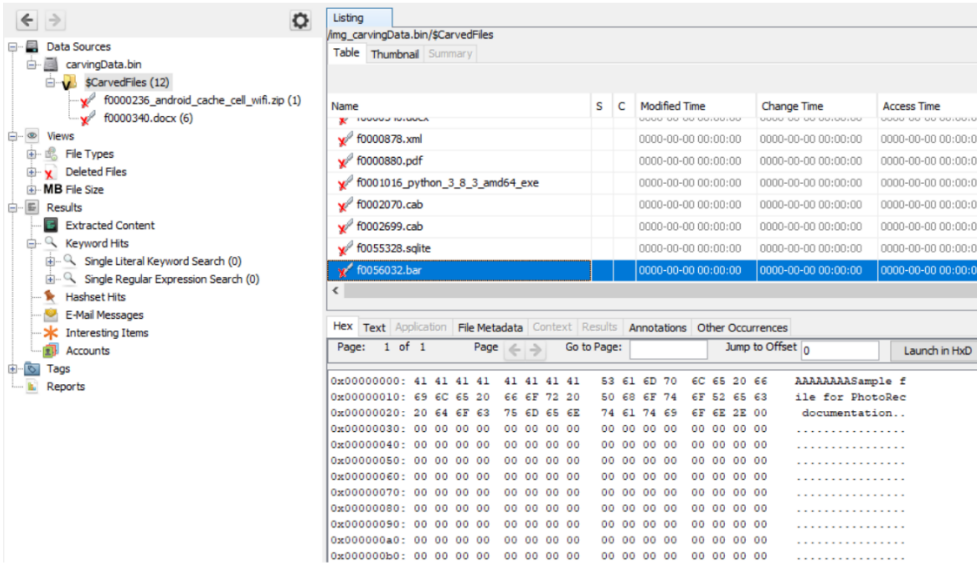


Bild 101: Anzeige einer erkannten Datei mit selbst hinzugefügten Datei Header unter Autopsy

Auf der Autopsy-Oberfläche werden die gefundenen Dateien unter der jeweiligen Datenquelle im Menüpunkt „$CarvedFiles“ angezeigt:

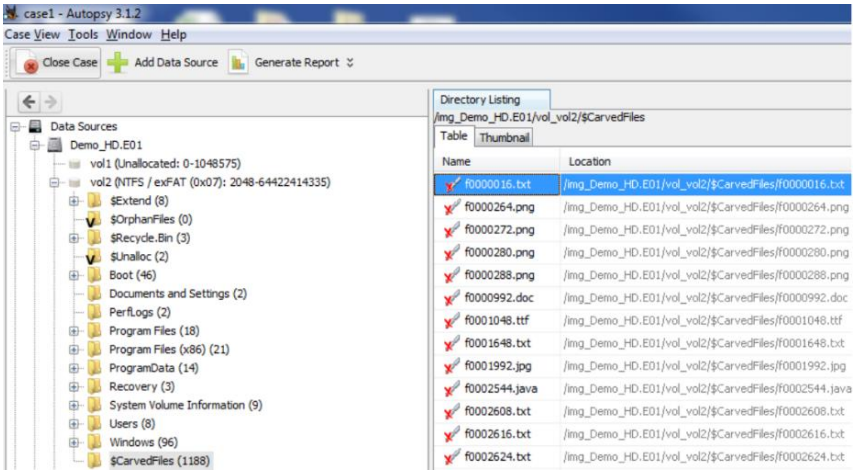


Bild 102: Anzeige erfolgreich gecarvter Dateien im Auswertungsergebnis von Autopsy

9.2.8 iOS Analyzer (iLEAPP), Android Analyzer Module (aLEAPP)

Diese Module sind zwar auch nicht für diese Projektarbeit relevant, allerdings dennoch erwähnenswert. Mit dem iOS Analyzer (iLEAPP) Modul können iOS sowie iPadOS Logs und Events analysiert werden. Dieses Modul besitzt keine Konfigurationsmöglichkeiten und nutzt iLEAPP: https://github.com/abrignoni/iLEAPP.

Mit dem Android Analyzer Module können SQLite und anderen Dateien von Android-Systemen analysiert werden.

9.3 Grundlagen der Dateiwiederherstellung

Im Folgenden soll das Prinzip der Wiederherstellung gelöschter Daten am Beispiel eines NTFS Dateisystems aufgezeigt werden. Hierfür soll die gelöschte Bilddatei „image004.jpg“ durch manuelle Arbeit in einem Hexeditor aus einem Datenimage rekonstruiert werden.

9.3.1 Suche mit Hilfe der $MFT

Im ersten Sektor der Partition befindet sich der Master Boot Record (MBR, NTFS Systemdatei $Boot). Aus diesem kann der Startsektor der $MFT Datei entnommen werden:

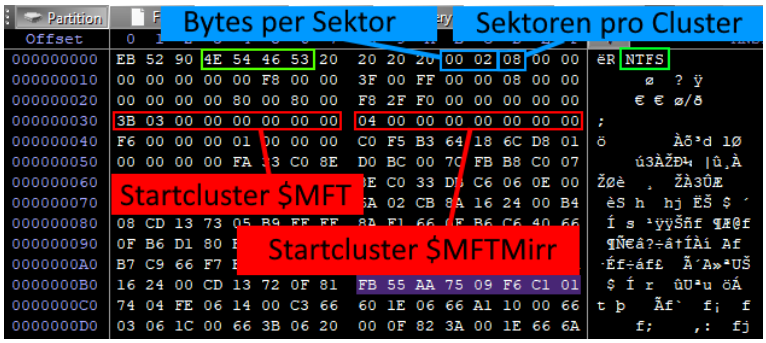


Bild 103: Aufbau des Bootsektors eines NTFS Dateisystems

Startcluster der $MFT: 0x033B

Zusammen mit den weiteren Informationen

Bytes per Sektor: 0x0200
Sekoren pro Cluster: 0x08

kann der Byteoffset der $MFT Berechnet werden:

Startcluster * Sektoren pro Cluster * Bytes per Sektor = Byteoffset

0x033B * 0x08 * 0x0200 = 0x33B000

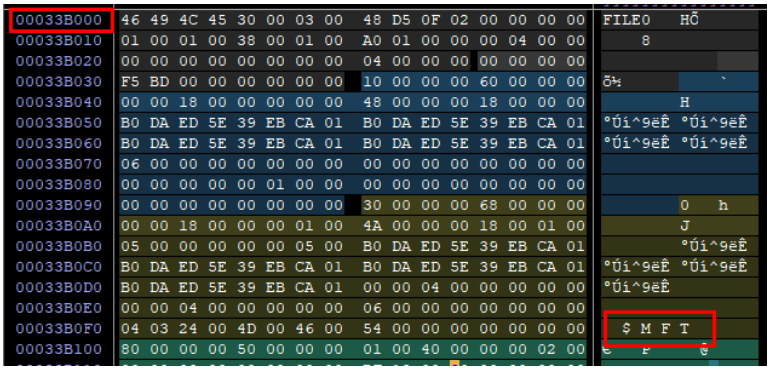


Bild 104: Erster Sektor der $MFT Systemdatei

Von diesem Offset startend wird anschließend entweder direkt nach dem Dateinamen „image004“ oder dessen Entsprechung in ASCII Hexwerten gesucht werden, um den entsprechenden Eintrag in der $MFT aufzufinden.

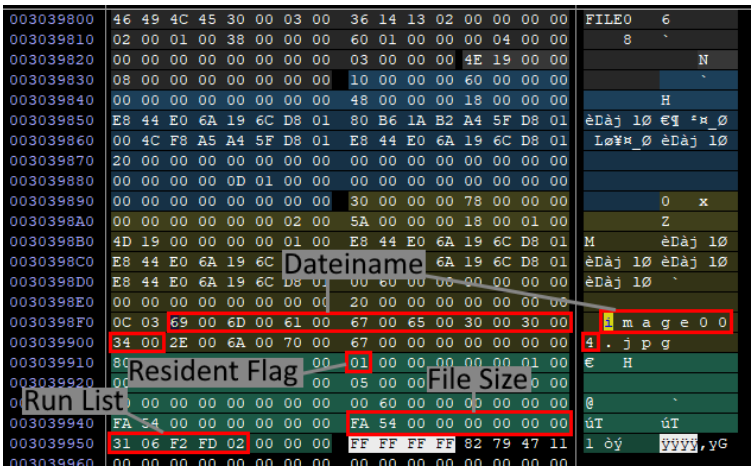


Bild 105: Offsetadresse, Clusteranzahl und Dateigröße einer nicht residenten Bilddatei in der $MFT

Aus den Werten des $MFT Eintrags geht hervor, dass die Daten der Datei nicht resident in der $MFT vorliegen (Resident Flag 0x01).

Es existiert lediglich eine Run List, da die Datei nicht fragmentiert auf dem Datenträger abgelegt wurde. Als Startcluster wird 0x02FDF2 angegeben. Die Datei belegt 0x06 Cluster.

Berechnung des Startwerts (Byteoffset zum Partitionsbeginn):

Startcluster * Sektoren pro Cluster * Bytes per Sektor = Byteoffset

0x02FDF2 * 0x08 * 0x0200 = 0x2FDF2000

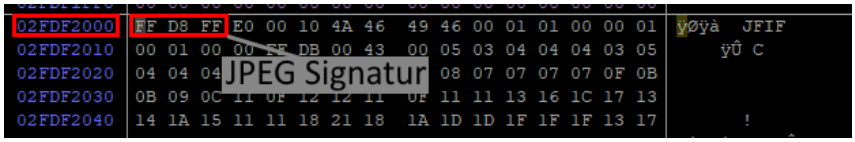


Bild 106: Startsektor der gesuchten Bilddatei mit JPEG Signatur

Da die ersten drei Bytes auf eine JPEG Signatur **0xFFD8FF** hindeuten, stellt dieser Sektor vermutlich den Beginn der gesuchten JPEG Datei dar.

Die Adresse des Dateiendes ergibt sich aus dem errechneten Startbytewert und der logischen Dateigröße (aus $MFT):

0x2FDF2000 + 0x54FA – 0x01 = 0x2FDF74F9

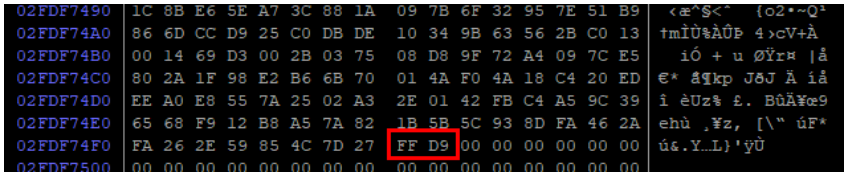


Bild 107: Dateiende mit JPEG Footer der gesuchten Bilddatei

Die letzten beiden Bytes **0xFFD9** bilden einen JPEG Footer. Die Bytes vom Start- bis zum Endwert können jetzt in eine Datei extrahiert und mit einem Bildbetrachtungsprogramm geöffnet werden:

Bild 108: Inhalt der wiederhergestellten Bilddatei

9.3.2 Dateiwiederherstellung ohne $MFT

Ist der Eintrag der wiederherzustellenden Datei in der $MFT bereits für eine dritte Datei verwendet worden oder ist gar die $MFT selbst unbrauchbar oder durch Neuformatierung überschrieben worden, kann diese nicht mehr zum Auffinden der wiederherzustellenden Datei verwendet werden, selbst wenn deren gespeicherte, nicht residente Daten noch auf dem Datenträger liegen.

In diesem Fall kann auf die Technik des „Carvings“ zurückgegriffen werden. Hier werden die nicht allokierten Bereiche (Unallocated Clusters) des Datenträgers nach Dateisignaturen (Fileheader/-footer) durchsucht und diese zur Wiederherstellung der Daten genutzt [18, S. 99 ff].

Beispielhaftes manuelles Carving einer JPEG Datei:

Als erster Schritt wird auf dem Datenträger der Hex-Wert der JPEG Signatur (FF D8 FF) zu suchen. Hierbei ist darauf zu achten, dass sich die Signatur an einem Sektorenanfang befindet.

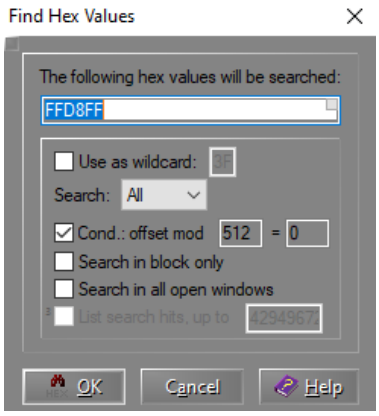


Bild 109: Suchmenü für Hexadezimalwerte in X-Ways Forensics: JPEG Header Suche

Der erste Treffer wird hierbei an Byteoffset 0x02FEFE000 erzielt. Die der Signatur nachfolgenden ASCII Werte „JFIF“ bestätigen in der Regel, dass es sich um den Beginn einer JPEG Datei handelt.

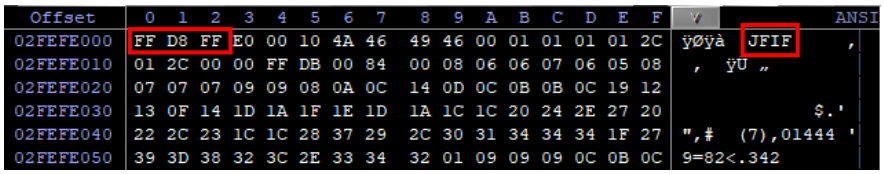


Bild 110: Treffer für die gesuchten Hexadezimalwerte: Dateibeginn einer JPEG Datei

Von diesem Punkt auf dem Datenträger erfolgt die Suche nach einem JPEG Footer 0XFFD9. Hier wird darauf geachtet, dass ganze Sektoren durchsucht werden und nicht nur deren Startwerte:

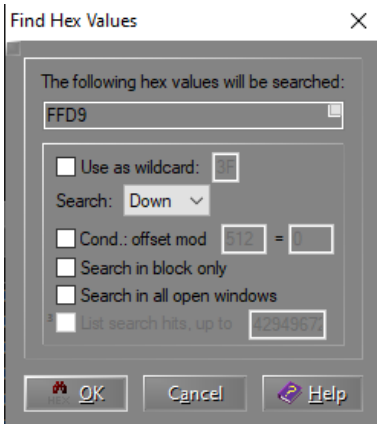


Bild 111: Suchmenü für Hexadezimalwerte in X-Ways Forensics: JPEG Footer Suche

Der erste Treffer für einen Footerkandidaten wird bei Byteoffset 0x2FF03A3F erzielt:

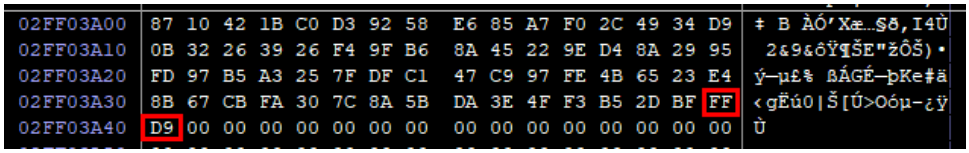


Bild 112: Treffer für die gesuchten Hexadezimalwerte: Dateiende einer JPEG Datei

Die Hexwerte von Header bis Footer können anschließend in eine Datei extrahiert und mit einem Bildbetrachtungsprogramm überprüft werden:

Bild 113: Inhalt der wiederhergestellten Bilddatei

Für das Filecarving ist jedoch anzumerken, dass hierbei nicht gezielt nach einer bestimmten Datei gesucht werden kann (z. B. eine bestimmte Bilddatei), da hierbei die reinen Dateiinhalte gefunden werden, jedoch keine Metadaten, wie Dateiname, Speicherort, Zeitstempel gewonnen werden können.

Als Praxis-Beispiel für den Zugriff auf ein APFS Dateisystem und zur Demonstration soll eine exemplarische Auswertung gem. den vermittelten Modulinhalten

von Herrn Diplom Ingenieur Hans-Peter Merkel auf seiner Lernplattform https://www.4n6.de/ mittels Sleuth Kit dienen. Sleuth Kit wird im Hintergrund von Autopsy verwendet, daher werden sind die folgenden Schritte auch für Autopsy relevant. Bevor die üblichen Befehle ausgeführt werden können, muss zuerst der Pool (228224) zur jeweiligen Partition (409640) in den Metadaten des Containers abgefragt werden. Die Partitionen sind über die Partitionstabelle erreichbar.

```
mmls apfs_sample2.E01
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

Slot Start End Length Description
000: Meta    0000000000 0000000000 0000000001 Safety Table
001: ------- 0000000000 0000000039 0000000040 Unallocated
002: Meta    0000000001 0000000001 0000000001 GPT Header
003: Meta    0000000002 0000000033 0000000032 Partition Table
004: 000     0000000040 0000409639 0000409600 EFI System Part
005: 001     0000409640 0117210199 0116800560
006: ------- 0117210200 0117210239 0000000040 Unallocated
```

Bild 114: Anwendung des Sleuth Kit Befehls mmls

```
pstat -o 409640 apfs_sample2.E01 | less
POOL CONTAINER INFORMATION
----------------------------------------------
Container dd69cf03-7fe5-4ae3-b28b-97b9eb23982a
==============================================
Type: APFS
NX Block Number: 0
NX oid: 1
NX xid: 85
Checkpoint Descriptor Block: 169

Capacity Ceiling (Size): 59801886720 B

Capacity In Use: 658972672 B
Capacity Available: 59142914048 B
Block Size: 4096 B
Number of Blocks: 14600070
Number of Free Blocks: 14439188

+-> Volume 83125ce0-ece6-4e8c-b878-9b614026bd79
| ============================================
| APSB Block Number: 228224
| APSB oid: 1026
| APSB xid: 85
| Name (Role): MacHD (No specific role)
| Capacity Consumed: 536494080 B
| Capacity Reserved: None
| Capacity Quota: None
| Case Sensitive: No
| Encrypted: No
| Formatted by: diskmanagementd (1412.61.1)
| Created: 2020-01-20 12:09:34.439765863 (CET)
| Changed: 2020-01-25 09:17:58.020371149 (CET)
```

Bild 115: Anwendung des Sleuth Kit Befehls pstat

Mit dem Partitions-Offset und dem Pool Volume Block kann nun bspw. das Dateisystem durchsucht werden. Anders als NTFS (512 Byte) besitzt APFS i. d. R. eine Blockgröße von 4096 Byte:

```
fls -o 409640 -B 228224 apfs_sample2.E01
r/r 124: ivanka.jpg
d/d 16: .Spotlight-V100
r/r 119: .DS_Store
r/r 122: putin_rettet_russland.jpg
r/r 138: jre-8u181-macosx-x64.dmg
d/d 1751: .Trashes
r/r 137: gimp-2.10.12-x86_64.dmg
r/r 139: Skype-8.15.0.4.dmg
d/d 19: .fseventsd
r/r 123: erdo.jpg
r/r 143: X2GoClient_latest_macosx.dmg
d/d 144: forensik
```

Bild 116: Anwendung des Sleuth Kit Befehls fls

Ähnlich dem NTFS besitzen auch hier die Dateien eine Inode, mit der sie letztendlich extrahiert werden können:

```
icat -o409640 -B 228224 apfs_sample2.E01 123 > erdo.jpg
file erdo.jpg
```

Bild 117: Anwendung des Sleuth Kit Befehls icat

Abschließend sei erwähnt, dass APFS anders als NTFS auch den Timestamp speichert, an dem die Attribute einer Datei das letzte Mal geändert wurden.

```
istat -o 409640 -B 228224 apfs_sample2.E01 123
Inode Number: 123
Allocated

Type: Regular File
Mode: rrwxrwxrwx
Size: 115587
owner / group: 99 / 99
Number of Links: 1

Filename: erdo.jpg
BSD flags: 0x00000000

Times:
(B) Created:            2020-01-20 12:43:56.000000000 (CET)
(M) Content Modified:   2019-11-09 11:33:46.000000000 (CET)
(C) Attributes Modified: 2020-01-20 13:45:13.853022632 (CET)
(A) Accessed:           2020-01-20 13:45:15.031313876 (CET)
Date Added:         2020-01-20 13:45:13.773675722 (CET)
```

Bild 118: Anwendung des Sleuth Kit Befehls istat