



Fakultät für Ingenieurwissenschaften

Bachelor-Thesis

Forensische Analyse des Microsoft Windows Thumbnail Cache

Abschlussarbeit zur Erlangung des Grades eines

Bachelor of Engineering

der Hochschule Wismar

Eingereicht am: 22. August 2023
Eingereicht von: Stefan Augustin
Studiengang IT-Forensik

Erstgutachterin: Prof. Dr.-Ing. Antje Raab-Düsterhöft
Zweitgutachter: Dipl.-Ing. Hans-Peter Merkel

Aufgabenstellung

Titel: Forensische Analyse des Microsoft Windows Thumbnail Cache

Title: Forensic Analysis of the Microsoft Windows Thumbnail Cache

Das Ziel der Bachelor-Thesis besteht darin, diverse Aspekte des Microsoft Windows Thumbnail Cache der Betriebssysteme Windows 10 und Windows 11 zu vergleichen.

Hierbei wird untersucht, welchen Einfluss die jeweiligen Vorschau-Einstellungen des Windows-Explorers und der Speicherort der Originaldateien auf die Entstehung von Einträgen im Thumbnail-Cache haben.

Zusätzlich sind Wege der Manipulation des Thumbnail-Cache und deren forensischer Nachweis zu beleuchten.

Die mögliche Gewinnung von Metadaten zu den Einträgen des Windows Thumbnail-Cache ist ebenfalls zu behandeln.

Kurzreferat

In der IT-Forensik dient der Microsoft Windows Thumbnail Cache als potentielle Quelle von Beweisen, die über Schuld oder Unschuld im Strafverfahren entscheiden kann. Ein genaues Verständnis über diese Beweisquelle ist bei der forensischen Arbeit also von großer Bedeutung. Diese Thesis verfolgt deshalb das Ziel einer vergleichenden Analyse der Mechanismen des Thumbnail Cache der Betriebssysteme Microsoft Windows 10 und Windows 11. Optionen zur Metadatengewinnung werden ebenso untersucht wie Manipulationsmöglichkeiten und deren forensischer Nachweis.

Die praktische Umsetzung erfolgte mit Hilfe virtueller Maschinen, in denen der Aufbau und das Verhalten des Thumbnail Cache durch Experimente untersucht wurde.

Die gewonnenen Erkenntnisse zeigen, dass die Einträge des Thumbnail Cache anhand bestimmter Regeln in den jeweiligen Datenbank- und Verwaltungsdateien erstellt werden. Es existieren nur wenige Quellen für Metadaten, die noch dazu Einschränkungen unterworfen sind. Versuche haben gezeigt, dass der Thumbnail Cache als ganzes deaktiviert und vorhandene Einträge beliebig manipuliert werden können. Ein forensischer Nachweis ist dabei nicht in jedem Fall möglich.

Abstract

In IT forensics, the Microsoft Windows thumbnail cache serves as a potential source of evidence that can determine guilt or innocence in criminal proceedings. An accurate understanding of this source of evidence is therefore of great importance in forensic work. This thesis therefore pursues the objective of a comparative analysis of the mechanisms of the thumbnail cache of the operating systems Microsoft Windows 10 and Windows 11. Sources for metadata acquisition are examined as well as ways of manipulation and their forensic detection.

The practical implementation was carried out with virtual machines, in which the structure and behavior of the thumbnail cache was investigated by experiments.

The findings show that the thumbnail cache entries are created according to certain rules in the respective databases and management files. There are only a few sources for metadata, which are also subject to restrictions. Tests have shown that the thumbnail cache can be deactivated as a whole and existing entries can be manipulated at will. Forensic proof is only partially possible.

Inhaltsverzeichnis

1. Einleitung	7
1.1. Motivation	7
1.2. Abgrenzung	8
1.3. Sprachliche und typographische Anmerkungen	8
2. Erläuterung der Forschungsfragen	9
2.1. Mechanismen	9
2.2. Gewinnung von Metadaten	10
2.3. Manipulationen und deren Erkennung	10
3. Grundlagen	11
3.1. Thumbnail	11
3.2. IT-Forensik	11
3.2.1. Definition	11
3.2.2. Antiforensik	12
3.3. Windows- / Datei-Explorer	12
3.4. Windows Search	13
3.5. SQLite Datenbanken	14
3.5.1. Definition	14
3.5.2. Identifizierung	14
3.6. Extensible Storage Engine	15
3.6.1. Definition	15
3.6.2. Identifizierung	15
3.6.3. Überprüfung und Reparatur	15
4. Windows Thumbnail Cache	18
4.1. Allgemeines	18
4.2. Aufbau der Thumbcache-Datenbanken	20
4.2.1. Dateiheder	21
4.2.2. Einträge in der Thumbcache-Datenbank	21
4.3. Aufbau der Thumbcache-Indexdatei	23

4.4.	Windows 10 / Windows 11	25
4.4.1.	Beteiligte Dateien	25
4.4.2.	Aufbau der Thumbcache-Datenbanken	26
4.4.3.	Einträge in der Thumbcache-Datenbank	29
4.4.4.	Aufbau der Indexdatei	30
4.5.	Nutzungsverhalten der Thumbcache-Datenbanken	31
5.	Forschungsumgebung und verwendete Software	33
5.1.	Forschungsumgebung	33
5.2.	Virtuelle Maschinen	33
5.3.	Verwendete Software	34
6.	Analyse der Mechanismen	35
6.1.	Vorschaufunktionen des Datei-Explorers	35
6.1.1.	Vorgehen	35
6.1.2.	Betrachtung der einzelnen Vorschauoptionen	38
6.1.3.	Betrachtung der Wechselwirkungen	39
6.1.4.	Bewertung	40
6.2.	Einfluss des Speicherortes der Originaldateien	41
6.2.1.	Lokale Speicherung	42
6.2.2.	Cloudspeicher	42
6.2.3.	Externe Datenträger	44
6.2.4.	Netzwerkspeicher	45
6.2.5.	Bewertung	46
6.3.	Sonstige Erkenntnisse	47
6.3.1.	Inhaltsvorschau von Dateiordnern	47
6.3.2.	Drag-and-drop	47
6.3.3.	Automatisierte Thumbnails	48
6.3.4.	Bewertung	48
7.	Gewinnung von Metadaten	49
7.1.	Stand der Forschung	49
7.2.	Vorgehen	50
7.3.	Ergebnis Windows 10	51
7.4.	Ergebnis Windows 11	53
7.5.	Untersuchung von Windows Search	53
7.6.	Weitere Quellen zur Gewinnung von Metadaten	56
7.7.	Bewertung	56

8. Manipulationsmöglichkeiten	57
8.1. Deaktivierung des Thumbcache	57
8.1.1. Vorgehen	57
8.1.2. Methode 1: Explorer-Optionen	58
8.1.3. Methode 2: Systemeinstellungen	58
8.1.4. Methode 3: Gruppenrichtlinie	59
8.1.5. Kontrolle der Wirksamkeit der Konfigurationen	60
8.1.6. Nachweis der Konfigurationsänderungen	60
8.1.7. Bewertung	62
8.2. Löschen der Thumbcache-Datenbanken	62
8.2.1. Nutzung der Datenträgerbereinigung	62
8.2.2. Nutzung der Systemeinstellungen	65
8.2.3. Manuelles löschen	66
8.2.4. Wiederherstellung der gelöschten Daten	66
8.2.5. Bewertung	68
8.3. Löschen einzelner Einträge des Thumbnail Cache	68
8.3.1. Vorgehen	68
8.3.2. Ergebnis	69
8.3.3. Bewertung	70
8.4. Einfügen von Inhalten in Thumbcache-Datenbanken	70
8.4.1. Erstellen neuer Einträge	70
8.4.2. Manipulation vorhandener Einträge	74
8.4.3. Bewertung	76
9. Empfohlenes Vorgehen bei der forensischen Auswertung	77
10. Zusammenfassung und Ausblick	81
Literatur	84
Abbildungsverzeichnis	90
Tabellenverzeichnis	91
Verzeichnis der Abkürzungen	92
Anhang	93
A. Anhang	93

1. Einleitung

1.1. Motivation

In der digitalen Forensik werden unzählige Artefakte aus installierten Programmen oder Betriebssystemen verwendet, um Sachverhalte nachzuvollziehen und Täter_innen zu überführen. Eines dieser Artefakte stellt der Windows Thumbnail Cache dar, der Ort, an dem Vorschaubilder des Windows Datei-Explorers gespeichert werden.

Die Forensiker_innen haben dabei den Vorteil, dass die Existenz des Thumbnail Cache vielen Computernutzer_innen nicht bekannt ist [32]. So finden sich darin oft noch Thumbnails bereits gelöschter Originaldateien, was etwa in Fällen illegaler Bildinhalte von Nutzen sein kann [20].

Fließen Beweise wie die Daten aus dem Thumbnail Cache in ein Strafverfahren ein, so unterliegen Sie letztendlich der Würdigung im Strafprozess, wobei die Anforderungen an deren Beweiskraft mit höherwertigen Delikten und deren drohenden Strafen ebenfalls steigen. Um zu einem Urteil zu kommen, müssen die vorgelegten Beweise Richter_innen und Schöff_innen von Schuld oder Unschuld der angeklagten Person überzeugen [19]. Hier sind vor allem die Forensiker_innen in der Pflicht, die von ihnen angefertigten Auswertungen von Asservaten im Zeugenstand adäquat zu vertreten. Dies fällt umso leichter, je vertrauter man mit der Quelle der Beweise und deren Mechanismen und Techniken ist.

Ziel dieser Arbeit ist die vergleichende Analyse des Thumbnail Cache der Betriebssysteme Windows 10 und Windows 11. Der Fokus liegt dabei auf der Feststellung der Mechanismen, unter denen die Einträge im Thumbnail Cache erzeugt werden, Quellen für Metadaten zu den eingebetteten Thumbnails zu identifizieren und den Thumbnail Cache auf Manipulationsmöglichkeiten und deren forensischen Nachweis zu prüfen. Die der Arbeit zugrunde liegenden Fragestellungen entstammen dabei Gerichtsverfahren, an denen der Verfasser als Zeuge der Digitalen Forensik einer deutschen Behörde teilnahm. Deren Beantwortung kann aktuellen und zukünftigen Forensiker_innen als Informations- und Referenzquelle bei der täglichen Arbeit dienen.

1.2. Abgrenzung

Zur Umsetzung dieser Thesis werden folgende Abgrenzungen formuliert:

- In der Thesis wird ausschließlich der zentrale Windows Thumbnail Cache behandelt, der mit Windows Vista eingeführt wurde. Dessen Vorgänger, die dezentralen Dateien *thumbs.db* werden nicht behandelt.
- Es werden ausschließlich Thumbnail Caches der Windows Betriebssysteme 10 und 11 untersucht.
- Die Entwicklung von Programmen zur Analyse oder Manipulation des Windows Thumbnail Cache ist nicht Teil der Thesis.
- Die Analyse der Mechanismen des Windows Thumbnail Cache erfolgt bei maximiertem Fenster des Datei-Explorers.
- Die Untersuchungen zum Löschen der Thumbcache-Datenbanken beschränken sich auf Bordmittel der Betriebssysteme. Software von Drittanbietern wird nicht mit einbezogen.
- Es werden ausschließlich Bilddateien als Ursprung der Thumbcache-Einträge verwendet.

1.3. Sprachliche und typographische Anmerkungen

Sprachliche Anmerkungen

Der Windows Thumbnail Cache verfügt über eine Vielzahl von Komponenten, die sich weder wörtlich noch sinngemäß adäquat in die deutsche Sprache übersetzt lassen. Aus diesem Grund wird im Rahmen der Thesis auf die Übersetzung von Fachbegriffen verzichtet, sobald die Verständlichkeit und sprachliche Ästhetik darunter zu leiden haben. Aus demselben Grund wird, abhängig vom Kontext, der förmliche Begriff „Thumbnail Cache“ durch „Thumbcache“ ersetzt.

Typografische Anmerkungen

In dieser Arbeit dient die *Kursivschrift* zur Hervorhebung von Begriffen und wörtlichen Zitaten.

2. Erläuterung der Forschungsfragen

2.1. Mechanismen

Dieser Teil der Forschungsfragen beschäftigt sich mit den grundlegenden Mechaniken, die den Funktionen des Windows Thumbnail Cache zugrunde liegen. Es soll geklärt werden, welche Nutzeraktionen die einzelnen Thumbcache-Datenbanken beeinflussen.

Welchen Einfluss haben die einzelnen Vorschauoptionen des Datei Explorer?

Wie in Punkt 4.1 erläutert, existieren für ein Benutzerkonto nicht nur eine einzelne, sondern mehrere Thumbcache-Datenbanken, die jeweils Thumbnails unterschiedlicher Dimensionen enthalten. Die Forschungsfrage soll klären, durch welche Konfiguration der Vorschaufunktion des Datei Explorers, welche Datenbanken mit Vorschaubildern befüllt werden. Die Resultate geben in der IT-forensischen Praxis Antworten darauf, welche Vorschauansicht aktiv war, um bestimmte Dateien im Explorer zu betrachten.

Welchen Einfluss hat die Speicherung außerhalb des Benutzerverzeichnisses?

Das Heimverzeichnis eines Benutzerkontos ist nur ein Ort, an dem Daten abgelegt werden können. Die Forschungsfrage soll deshalb klären, ob und wie Einträge im Windows Thumbcache gesetzt werden, wenn die Speicherung der Originaldateien außerhalb dieser Verzeichnisstruktur, jedoch auf einem internen Speichermedium erfolgt.

Welchen Einfluss hat die externe Speicherung?

Es ist nicht zwingend nötig, Daten nur auf internen Datenspeichern abzulegen. Hierfür können auch externe Datenträger genutzt werden. Die Forschungsfrage soll klären, ob die Nutzung solcher externer Medien das Erstellen von Einträgen in den Thumbcache-Datenbanken beeinflusst.

Welchen Einfluss hat die Speicherung in der Cloud?

Im Jahr 2021 nutzten über ein Drittel der Deutschen Cloudspeicher zum Ablegen ihrer Daten [2]. Die Forschungsfrage soll den Einfluss von Cloudspeichern unterschiedlicher Anbieter auf die untersuchte Thumbcache-Funktion beleuchten.

2.2. Gewinnung von Metadaten

Können Metadaten zu den Thumbcache-Einträgen gewonnen werden?

Wie in Abschnitt 4 gezeigt, sind in den Thumbcache-Datenbanken keine Metadaten vorhanden, die mit den Originaldateien zusammenhängen. Deshalb wird untersucht, ob und wie die Gewinnung dieser Metadaten möglich ist.

2.3. Manipulationen und deren Erkennung

Die folgenden Forschungsfragen betreffen die Manipulierbarkeit des Windows Thumbnail Cache. Es wird untersucht, ob und wie Täter_innen Spuren verwischen oder auch setzen können und wie diese Manipulationen erkannt werden können.

Kann die Thumbcache-Funktion deaktiviert werden?

Diese Forschungsfrage soll klären, ob es Möglichkeiten gibt, die Windows Thumbcache-Funktion in Gänze zu deaktivieren.

Ist das Löschen der Thumbcache-Datenbanken möglich?

Da die Speicherung der Vorschaubilder in Dateien erfolgt (siehe 4.1), geht diese Forschungsfrage den Möglichkeiten auf den Grund, diese Dateien als Ganzes vom Datenträger zu löschen. Das Augenmerk liegt hierbei auf Mitteln, die die Betriebssysteme bereits beinhalten.

Können Einträge in einer Thumbcache-Datenbank manipuliert werden?

Manipulation von Beweisquellen sind Teil der Antiforensik (siehe 3.2.2). Die Einträge im Windows Thumbnail Cache sind somit ebenfalls ein potentielles Ziel von Manipulationen jeder Art.

Im Detail wird untersucht, ob das Platzieren von falschen Beweisen im Thumbnail Cache möglich ist. Hierfür wird geprüft, ob neue Einträge für nicht existierende Originaldateien hinzugefügt werden können und ob ein Thumbnail eines bestehenden Eintrages durch ein anderes ausgetauscht werden kann. Auch das Löschen eines vorhandenen Eintrags aus den Datenbanken ist Gegenstand dieser Forschungsfrage.

3. Grundlagen

Im nachfolgenden Kapitel werden Techniken und Begriffe erläutert, die mit dem Windows Thumbnail Cache und dessen Funktionsweise in Verbindung stehen.

Als potentiellem Beweismittel im Strafprozess unterliegt die Aufarbeitung des Windows Thumbnail Cache den Anforderungen der IT-Forensik und möglichen anti-forensischen Eingriffen, weshalb auf die beiden Begriffe ebenfalls eingegangen wird.

3.1. Thumbnail

In der Informatik versteht man unter dem Begriff Thumbnail eine miniaturisierte Version eines größeren Objekts, wie z.B. ein Vorschaubild eines damit verlinkten Originalbildes auf einer Internetseite. In diesem Beispiel wird dadurch die Ladezeit der Internetseite optimiert, da nicht die deutlich größere Originaldatei heruntergeladen werden muss. [31]

3.2. IT-Forensik

Im Folgenden wird der Begriff der IT-Forensik definiert, im System der Wissenschaften eingeordnet und die als ihr Gegenpol fungierende Antiforensik vorgestellt.

3.2.1. Definition

Seinen Ursprung hat der Begriff ‚Forensik‘ im lateinischen Wort *forēnsis*, das in etwa mit ‚im Forum (Marktplatz)‘ oder ‚auf dem Forum‘ übersetzt werden kann. Im historischen Kontext war der Marktplatz, bzw. das Zentrum einer Ansiedlung oftmals Mittelpunkt der Gerichtsbarkeit, der Ort, an dem Recht gesprochen wurde. So stellt das Adjektiv ‚forensisch‘ bis heute einen Bezug zum Rechtssystem her. Jede Wissenschaft, die im straf- oder zivilrechtlichen Rahmen Handlungen identifiziert, ausschließt, analysiert oder rekonstruiert, kann der Forensik zugeordnet werden [19].

Die IT-Forensik stellt einen Teilbereich der Forensik dar [13] und wird vom Bundesamt für Sicherheit in der Informationstechnik (BSI) folgendermaßen definiert [4]:

„IT-Forensik ist die streng methodisch vorgenommene Datenanalyse auf Datenträgern und in Computernetzen zur Aufklärung von Vorfällen unter Einbeziehung der Möglichkeiten der strategischen Vorbereitung insbesondere aus der Sicht des Anlagenbetreibers eines IT-Systems.“

3.2.2. Antiforensik

In ihrer Eigenschaft als Teilgebiet der Forensik kann die Arbeit der IT-Forensik von antiforensischen Maßnahmen betroffen sein, die in folgende Kategorien eingeteilt werden können [43]:

- Data Hiding (Verstecken von Daten)
- Artifact Wiping (Vernichten von Daten)
- Trail Obfuscation (Verschleiern der Herkunft von Daten)
- Attacks Against The Computer Forensic Process / Tools (Angriffe auf Vorgehensweisen / Software der IT-Forensik)

Neben der Kategorie des Vernichtens von Daten wird in der Thesis zusätzlich die Möglichkeit des Setzens falscher Spuren behandelt, die keiner der oben genannten Kategorien zuzuweisen ist.

3.3. Windows- / Datei-Explorer

Beim Windows-Explorer handelt es sich um eine im Windows Betriebssystem integrierte Software, die es ermöglicht, den Inhalt von Datenträgern über ein Benutzerinterface zu verwalten [11]. Zwischenzeitlich wurde das Programm in Datei-Explorer umbenannt, so dass diese Bezeichnung bei Windows 10 und Windows 11 Verwendung findet. [29].

Der Datei-Explorer in Windows 10 und Windows 11 hat die Fähigkeit, Ordnerinhalte in verschiedenen Ansichtsmodi zu präsentieren. So ist es möglich, Miniaturansichten des Inhalts diverser Dateiformate anzeigen zu lassen. Die verfügbaren Vorschauoptionen sind dabei in beiden Betriebssystemen identisch.

Des Weiteren besteht die Möglichkeit, den Inhalt einzelner Dateien in einem Bereich am rechten Rand des Datei-Explorer Fensters anzeigen zu lassen, dem sog. Vorschau-fenster.

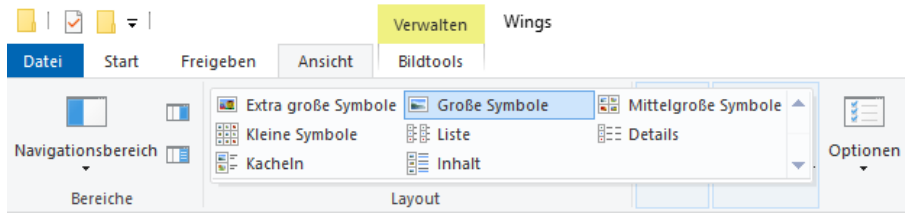


Abbildung 3.1.: Windows 10: Verfügbare Vorschaumodi im Datei-Explorer

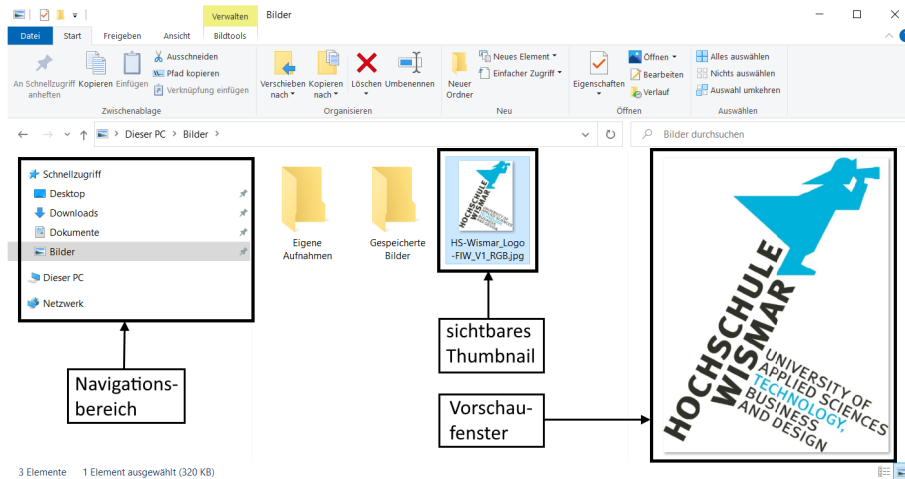


Abbildung 3.2.: Windows 10: Bestandteile des Datei-Explorers

3.4. Windows Search

Bei Microsoft Windows Search handelt es sich um eine Windowsfunktion, die die Suche nach Dateien, installierter Software, E-Mails und dem Internetverlauf erlaubt. Dazu werden beim Erstellen oder Ändern von Dateien die zugehörigen Metadaten in einer zentralen Datenbank erfasst. Diese enthält beispielsweise Dateinamen, Speicherpfade, Zeitstempel, die Adressen einer E-Mail oder die Dimensionen gespeicherter Bilddateien. Die Datenbank beherbergt dabei Informationen zu Dateien von sämtlichen vorhandenen Benutzerprofilen. In dem Index sind Informationen zu Dateien verfügbar, die aus möglicherweise nicht mehr zur Verfügung stehenden Quellen, wie Wechseldatenträgern oder verschlüsselten Speicherbereichen stammen [6].

Über das Eingabefeld neben dem Start-Button oder der Tastenkombination  + S kann dieser Index abgefragt werden [28].

3.5. SQLite Datenbanken

Im Folgenden wird das Datenbankformat SQLite vorgestellt.

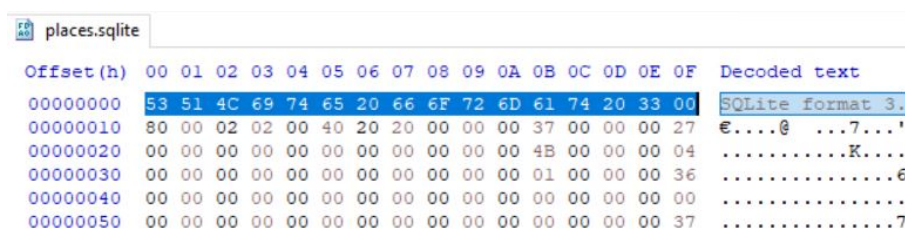
3.5.1. Definition

Bei SQLite handelt es sich um eine Programmbibliothek eines relationalen Datenbanksystems, die gemeinfrei für den privaten und kommerziellen Gebrauch verfügbar ist [1]. Es handelt sich hierbei um das am weitesten verbreitete Datenbanksystem der Welt, das alleine 2016 schon auf ungefähr 2,3 Milliarden Mobiltelefonen Anwendung fand. Anders als die meisten anderen Datenbanksysteme arbeitet es nicht mit einer Client-Server Infrastruktur, sondern läuft als alleinstehendes Programm oder Bibliothek auf dem Client selbst [44].

Der aktuelle Status der Datenbank befindet sich in einer einzelnen Datei. Während einer Transaktion werden zusätzliche Informationen in eine weitere Datei geschrieben, bei der es sich konfigurationsabhängig um ein sog. *Rollback Journal* oder ein *Write-Ahead Log* handelt [51].

3.5.2. Identifizierung

SQLite Datenbanken können durch ihre Dateisignatur (Magic Bytes) identifiziert werden. Diese 16 Byte lange Zeichenfolge startet an Offset 0 der Datenbank Hauptdatei und besteht aus der Hexadezimalen Codierung der Zeichenfolge *SQLite format 3* [44]



Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	53	51	4C	69	74	65	20	66	6F	72	6D	61	74	20	33	00	SQLite format 3.
00000010	80	00	02	02	00	40	20	20	00	00	00	37	00	00	00	27	€....@ ...7...'
00000020	00	00	00	00	00	00	00	00	00	00	00	4B	00	00	00	04K....
00000030	00	00	00	00	00	00	00	00	00	00	00	01	00	00	00	366
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	377

Abbildung 3.3.: Magic Bytes einer SQLite Datenbank

3.6. Extensible Storage Engine

Unter Windows 10 werden die Daten des Windows Search Dienstes in einer ESE Datenbank abgelegt [6]. Das Format wird im Folgenden vorgestellt.

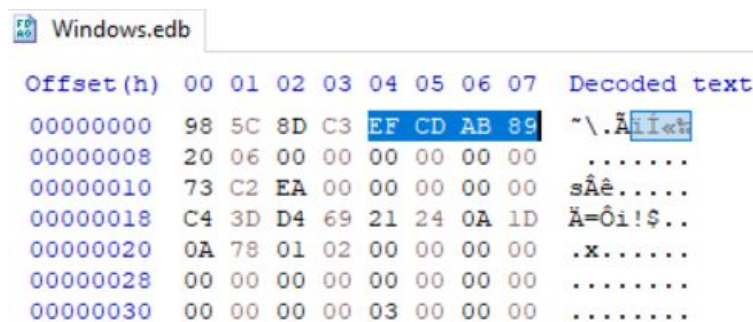
3.6.1. Definition

Bei der Extensible Storage Engine, auch bekannt als JET Blue, handelt es sich um ein von Microsoft entwickeltes Datenbankmodell. Es basiert auf der Technik *Index Sequential Access Method* (ISAM) [49]. ISAM war die erste Speicherstruktur, die effizient auf darin abgelegte Daten zugreifen konnte. Dies wird durch einen gesonderten Index ermöglicht, in dem Daten aufsteigend sortiert werden [50].

In der Vergangenheit erfuhr das ESE Format diverse Änderungen und kommt z.B. als Speicherstruktur im Microsoft Exchange Server, Active Directory oder Windows Search zum Einsatz [24].

3.6.2. Identifizierung

ESE Datenbanken besitzen eine Dateisignatur, anhand derer sie identifiziert werden können. Diese befindet sich an Offset 04 in der Datenbankdatei und bestehen aus den vier aufeinander folgenden Hexadezimalwerten \xEF \xCD \xAB \x89 [24].



Offset(h)	00	01	02	03	04	05	06	07	Decoded text
00000000	98	5C	8D	C3	EF	CD	AB	89	~\..ÄI<et
00000008	20	06	00	00	00	00	00	00
00000010	73	C2	EA	00	00	00	00	00	sÄê.....
00000018	C4	3D	D4	69	21	24	0A	1D	Ä=Öi!\$. .
00000020	0A	78	01	02	00	00	00	00	.x.....
00000028	00	00	00	00	00	00	00	00
00000030	00	00	00	00	03	00	00	00

Abbildung 3.4.: Magic Bytes einer ESE Datenbank

3.6.3. Überprüfung und Reparatur

Je nachdem, ob eine ESE Datenbank korrekt geschlossen wurde, kann sie die zwei Zustände *Clean Shutdown* oder *Dirty Shutdown* annehmen. Der jeweilige Zustand kann mit dem Windows Kommandozeilentool *esentutl.exe* überprüft werden [24].

3.6. EXTENSIBLE STORAGE ENGINE

```
E:\WTC>esentutl /mh Windows.edb

Extensible Storage Engine Utilities for Microsoft(R) Windows(R)
Version 10.0
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
    Database: Windows.edb

DATABASE HEADER:
Checksum Information:
Expected Checksum: 0xc38d5c98
    Actual Checksum: 0xc38d5c98

Fields:
    File Type: Database
    Checksum: 0xc38d5c98
    Format ulMagic: 0x89abcdef
    Engine ulMagic: 0x89abcdef
    Format ulVersion: 0x620,110,240 (attached by 9180)
    Engine ulVersion: 0x620,110,240 (efvCurrent = 9180)
    Created ulVersion: 0x620,20
    DB Signature: Create time:10/29/2020 10:36:33.128 Rand:1775517124 Computer:
    cbDbPage: 32768
    dbtime: 15385203 (0xeac273)
    State: Clean Shutdown ←
    Log Required: 0-0 (0x0-0x0)
```

Abbildung 3.5.: ESE Datenbank im Zustand *Clean Shutdown*

```
E:\WTC>esentutl /mh Windows.edb

Extensible Storage Engine Utilities for Microsoft(R) Windows(R)
Version 10.0
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
    Database: Windows.edb

DATABASE HEADER:
Checksum Information:
Expected Checksum: 0xcd48c26a
    Actual Checksum: 0xcd48c26a

Fields:
    File Type: Database
    Checksum: 0xcd48c26a
    Format ulMagic: 0x89abcdef
    Engine ulMagic: 0x89abcdef
    Format ulVersion: 0x620,20,0 (attached by 0)
    Engine ulVersion: 0x620,110,240 (efvCurrent = 9180)
    Created ulVersion: 0x620,20
    DB Signature: Create time:03/04/2020 16:50:07.068 Rand:2868354596 Computer:
    cbDbPage: 32768
    dbtime: 195335 (0x2fb07)
    State: Dirty Shutdown ←
    Log Required: 156-156 (0x9c-0x9c)
```

Abbildung 3.6.: ESE Datenbank im Zustand *Dirty Shutdown*

Obwohl das Verhindern von Veränderungen von Daten in der IT-Forensik als Notwendigkeit gilt [4], kann es unvermeidbar sein, eine ESE Datenbank zu reparieren, wenn ein Programm diese im Dirty Shutdown Zustand nicht öffnen kann [24].

Zur Reparatur der Datenbank kann wieder `esentutl.exe` verwendet werden und bietet hierfür die zwei Optionen *Recovery* (Schalter `/r`) und *Repair* (Schalter `/p`). Die *Recovery* Methode ist vorzuziehen, da hierbei noch nicht durchgeführte Transaktionen in die Datenbank übernommen werden. Hierfür werden jedoch neben der Datenbankdatei auch die zugehörigen Logdateien benötigt. Die *Repair* Methode dagegen versetzt die Datenbank im aktuellen Zustand wieder in einen fehlerfreien Status, ohne auf Logdateien angewiesen zu sein. Hierbei gehen jedoch noch nicht übernommene Transaktionen verloren [30].

```
E:\WTC>esentutl /p Windows.edb

Extensible Storage Engine Utilities for Microsoft(R) Windows(R)
Version 10.0
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating REPAIR mode...
    Database: Windows.edb
    Temp. Database: TEMPREPAIR24252.EDB

Checking database integrity.

The database is not up-to-date. This operation may find that
this database is corrupt because data from the log files has
yet to be placed in the database.

To ensure the database is up-to-date please use the 'Recovery' operation.


        Scanning Status (% complete)

    0   10   20   30   40   50   60   70   80   90  100
|----|----|----|----|----|----|----|----|----|----|
.....

Scanning the database.


        Scanning Status (% complete)

    0   10   20   30   40   50   60   70   80   90  100
|----|----|----|----|----|----|----|----|----|----|
.....

Repairing damaged tables.


        Scanning Status (% complete)

    0   10   20   30   40   50   60   70   80   90  100
|----|----|----|----|----|----|----|----|----|----|
.....

Repair completed. Database corruption has been repaired!

Note:
  It is recommended that you immediately perform a full backup
  of this database. If you restore a backup made before the
  repair, the database will be rolled back to the state
  it was in at the time of that backup.

Operation completed successfully with 595 (JET_wrnDatabaseRepaired, Database corruption has been
repaired) after 2.297 seconds.
```

Abbildung 3.7.: Reparatur einer ESE Datenbank

4. Windows Thumbnail Cache

In diesem Abschnitt wird der aktuelle Stand der Forschung zum Windows Thumbnail Cache dargelegt. Da die vorhandene Literatur fast ausnahmslos die Windows Versionen Vista bis Windows 8 behandelt, werden eigene Erkenntnisse zu den Betriebssystemen Windows 10 und 11 ergänzt. Zur Vorbereitung der Thesis wurden diese durch den Vergleich der bisherigen Forschungsergebnisse mit den aktuellen Gegebenheiten in den Betriebssystemen erarbeitet.

4.1. Allgemeines

Beginnend mit Windows 95b führte Microsoft die Möglichkeit ein, Ordnerinhalte im Windows-Explorer in Miniaturform anzuzeigen, anstatt diese nur als Liste oder Icons darzustellen, wie es in vorangegangenen Windows Versionen üblich war [5]. Dies wurde durch die Einführung von Systemdatenbanken namens *thumbs.db* erreicht, die automatisiert in Dateiordnern mit Bilddateien erstellt werden. In diesen Datenbanken werden nicht nur Thumbnails der jeweiligen Dateien, sondern auch deren Metadaten (z.B. Dateiname oder Pfad der Datei) abgelegt [14]. Mit Windows Vista begann der Weg weg von den dezentralen, hin zu zentralen Thumbnail Datenbanken im jeweiligen Benutzerverzeichnis [48].

Unter dem Speicherpfad

`%systemdrive%\Users\<Benutzername>\AppData\Local\Microsoft\Windows\Explorer\`

sind diverse Datenbanken vorzufinden, die nach den Dimensionen der darin abgelegten Vorschaubilder benannt sind. So steht der Dateiname *thumbcache_96.db* für eine Datenbank, in der Thumbnails mit einer maximalen Größe von 96x96 Pixel gespeichert werden [41, 48].

Die Datei *thumbcache_idx.db* enthält Pointer auf die einzelnen Einträge der verschiedenen Thumbcache-Datenbanken [32].

Thumbcache-Datenbanken

Wie obenstehend beschrieben, enthalten die Datenbankdateien Thumbnails in den im Dateinamen festgelegten maximalen Dimensionen. Die Speicherung erfolgt dabei in den Bildformaten JPEG, BMP oder PNG [32].

Im Laufe der Zeit fand kontinuierlich eine Erweiterung der unterstützten Dimensionen der gespeicherten Thumbnails statt, wie Tabelle 4.1 zeigt.

Windows Version	Datenbankdateien	Speicherort
95b, 98, ME, 2000, XP, 2003	Thumbs.db	Im jeweiligen Dateordner
Vista	Thumbcache_32, 96, 256, 1024	..\AppData\Local\Microsoft\Windows\Explorer\
Windows 7	Thumbcache_32, 96, 256, 1024	..\AppData\Local\Microsoft\Windows\Explorer\
Windows 8	Thumbcache_16, 32, 48, 96, 256, 1024, WIDE	..\AppData\Local\Microsoft\Windows\Explorer\

Tabelle 4.1.: Thumbcache-Datenbanken nach Betriebssystem [42]

Datei thumbcache_idx.db

Die Datei thumbcache_idx.db dient als Index, der den Zugriff auf die Einträge der einzelnen Thumbcache-Datenbanken und den in ihnen enthaltenen Thumbnails ermöglicht. Dies wird durch einen Pointer zum Offset des Thumbnails in der entsprechenden Thumbcache-Datenbank realisiert. [32].

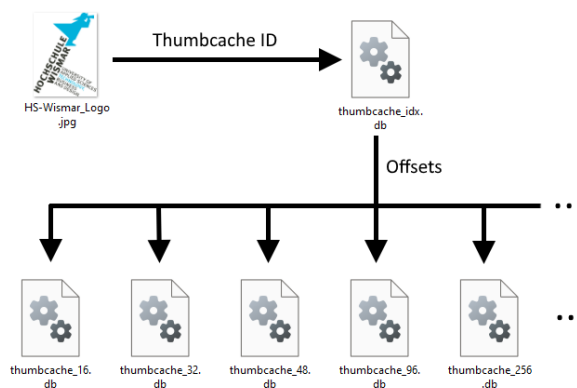


Abbildung 4.1.: Funktionsweise des Windows Thumbnail Cache

Datei thumbcache_sr.db

Obwohl die Indexdatei thumbcache_idx.db über die nötigen Datenfelder verfügt, um auf diese Datei zu verweisen [25], ist die Funktion der Datei im Unklaren. Sie besteht lediglich aus dem Standard Fileheader der Windows Thumbcache-Datenbanken [32].

4.2. Aufbau der Thumbcache-Datenbanken

Die Thumbcache-Datenbanken bestehen aus einem Dateiheader von 24 Bytes Länge, gefolgt von den einzelnen Einträgen in der Datenbank, die ihrerseits selbst über einen eigenen Header verfügen. Als Byte-Reihenfolge wird das Little-Endian Format genutzt. Text wird in UTF-16 Unicode dargestellt [25].

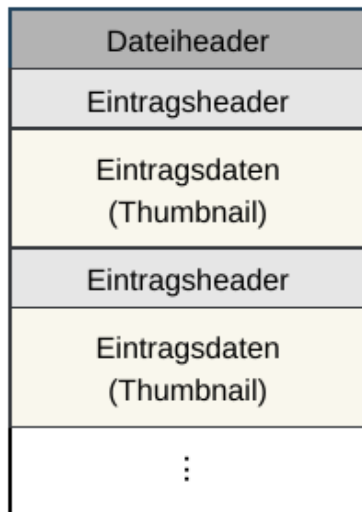


Abbildung 4.2.: Aufbau der Thumbcache-Datenbank

4.2.1. Dateiheader

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Dateisignatur (CMMM)
4	4	Format Version
8	4	Cache Typ
12	4	Offset zum ersten Cache Eintrag (oder Headergröße)
16	4	Offset zum ersten vorhandenen Cache Eintrag
20	4	Anzahl der vorhandenen Cache Einträge (nicht immer korrekt, möglicherweise auch die Anzahl der zugewiesenen Einträge)

Tabelle 4.2.: Thumbcache-Datenbank Header bei Windows Vista und Windows 7 [25]

Format Version

Das im Dateiheader angegebene Versionsformat gibt an, welches Format und Länge die Header einzelnen Cacheeinträge aufweisen. Es wird zwischen dem Format *Windows Vista* (0x20) und *Windows 7* (0x21) unterschieden [25].

Cache Typ

Jede Thumbcache-Datenbank identifiziert sich anhand eines Eintrags im Dateiheader, der den Cache Typ symbolisiert [25]. Für die Betriebssysteme Windows Vista und Windows 7 sind folgende Cache Typen und ihre Zuordnung dokumentiert:

Wert	Beschreibung
0x00	thumbcache_32.db
0x01	thumbcache_96.db
0x02	thumbcache_256.db
0x03	thumbcache_1024.db
0x04	thumbcache_sr.db

Tabelle 4.3.: Cache Typen für Windows Vista und Windows 7 [25]

4.2.2. Einträge in der Thumbcache-Datenbank

Das im Dateiheader angegebene Cache Format bestimmt die Länge und den Aufbau der Eintragsheader des Thumbnail Cache. Wie obenstehend erwähnt, unterscheidet man zwischen den Formaten Windows Vista und Windows 7 [25].

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Signatur (CMMM)
4	4	Gesamtgröße des Cache Eintrags
8	8	Thumbcache-ID
16	8	Dateierweiterung (UTF-16, muss nicht befüllt sein)
24	4	Größe des Eintragsnamens (identifiziert)
28	4	Größe des Paddings
32	4	Größe der Daten
36	4	Unbekannt (leeres Feld)
40	8	Prüfsumme der Daten (CRC-64)
48	8	Prüfsumme des Eintragsheaders (CRC-64)
56	Größe des Eintragsnamens	Eintragsname (UTF-16 String)
...	Paddinggröße	Padding aus 0x00 Bytes (nur vorhanden wenn Paddinggröße > 0)
...	Datengröße	Daten (Thumbnail Bilddaten)

Tabelle 4.4.: Thumbcache-Eintrag im Windows Vista Format [25]

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Signatur (CMMM)
4	4	Gesamtgröße des Cache Eintrags
8	8	Thumbcache-ID
16	4	Größe des Eintragsnamens (identifiziert)
20	4	Größe des Paddings
24	4	Größe der Daten
28	4	Unbekannt (leeres Feld)
32	8	Prüfsumme der Daten (CRC-64)
40	8	Prüfsumme des Eintragsheaders (CRC-64)
48	Größe des Eintragsnamens	Eintragsname (UTF-16 String)
...	Paddinggröße	Padding aus 0x00 Bytes (nur vorhanden wenn Paddinggröße > 0)
...	Datengröße	Daten (Thumbnail Bilddaten)

Tabelle 4.5.: Thumbcache-Eintrag im Windows 7 Format [25]

Thumbcache-ID

Die Thumbcache-ID an Offset 8 besteht aus 16 Hexadezimalwerten, die durch acht Bytes im Header eines Thumbcache-Eintrags dargestellt werden [35]. Wie die Experimente von Morris und Chivers [33] zeigen konnten, basiert die Berechnung der Thumbcache-ID auf dem Dateityp, dem Inhalt des MFT Eintrags der Originaldatei sowie dem Volume, auf dem diese gespeichert wurde. Durch ihre Einmaligkeit erfüllt sie die Definition eines Primärschlüssels in der Datenbank [45].

Prüfsummen

Die Berechnung der beiden vorhandenen Prüfsummen erfolgt mit einem proprietären CRC-64 Algorithmus mit unbekanntem Polynom. Es ist jedoch möglich, mit Hilfe der in der Systemdatei *thumbcache.dll* eingebetteten Wertetabelle, korrekte Prüfsummen zu berechnen [22].

4.3. Aufbau der Thumbcache-Indexdatei

Die Indexdatei *thumbcache_idx.db* enthält für jede Originaldatei einen Eintrag mit Pointern zu den ihr zugehörigen Thumbnails der einzelnen Thumbcache-Datenbanken. Die Zuordnung erfolgt hier über die Thumbcache-ID. [35]

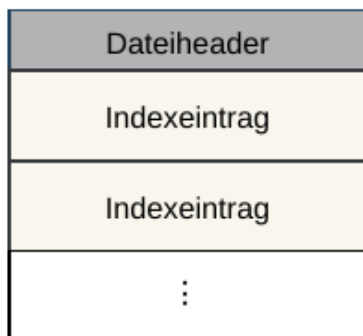


Abbildung 4.3.: Aufbau der Thumbcache-Indexdatei

Den Anfang der Indexdatei bildet ein 24 Bytes langer Dateiheader, in dem das Format der Indexeinträge festgelegt wird. Das Format wird wie bei den Thumbcache-Datenbanken durch die Werte 0x20 (Windows Vista) und 0x21 (Windows XP) bestimmt. Die Indexeinträge selbst haben eine Länge von 40 Bytes (Windows Vista Format) oder 32 Bytes (Windows 7 Format) [25].

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Signatur (IMMM)
4	4	Format Version
8	4	Unbekannt
12	4	Anzahl der benutzten Einträge
16	4	Gesamtzahl der Einträge (Summe der benutzten und nicht benutzten Einträge)
20	4	Unbekannt

Tabelle 4.6.: Dateiheader der Indexdatei thumbcache_idx.db [25]

Offset (Byte)	Größe (Byte)	Beschreibung
0	8	Thumbcache-ID
8	8	Datum und Uhrzeit der letzten Modifikation (FILETIME Format)
16	4	Flags
20	4	Offset in thumbcache_32.db
24	4	Offset in thumbcache_96.db
28	4	Offset in thumbcache_256.db
32	4	Offset in thumbcache_1024.db
36	4	Offset in thumbcache_sr.db

Tabelle 4.7.: Indexeintrag im Windows Vista Format [25]

Offset (Byte)	Größe (Byte)	Beschreibung
0	8	Thumbcache-ID
8	4	Flags
12	4	Offset in thumbcache_32.db
16	4	Offset in thumbcache_96.db
20	4	Offset in thumbcache_256.db
24	4	Offset in thumbcache_1024.db
28	4	Offset in thumbcache_sr.db

Tabelle 4.8.: Indexeintrag im Windows 7 Format [25]

4.4. Windows 10 / Windows 11

In der Literatur wird der Aufbau des Windows Thumbcache nur bis Windows 8 tiefergehend betrachtet. So beziehen sich die von Metz [25] durchgeführten Analysen des inneren Aufbaus der beteiligten Dateien auf Windows Vista und Windows 7. In diesem Abschnitt wird deshalb der Thumbnail Cache der Betriebssysteme Windows 10 und Windows 11 untersucht.

4.4.1. Beteiligte Dateien

Die Prüfung beider Betriebssysteme ergab, dass der ursprüngliche Speicherort der Thumbcache-Dateien beibehalten wurde, jedoch befinden sich dort zusätzliche Thumbcache-Datenbanken, die das Spektrum der unterstützten Bilddimensionen erweitern.

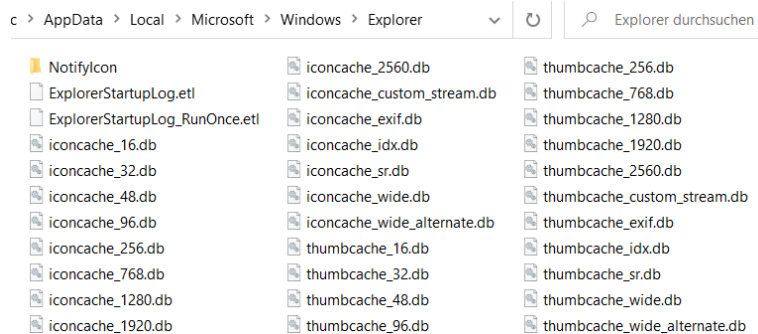


Abbildung 4.4.: Thumbcache-Ordner eines Windows 10 Systems

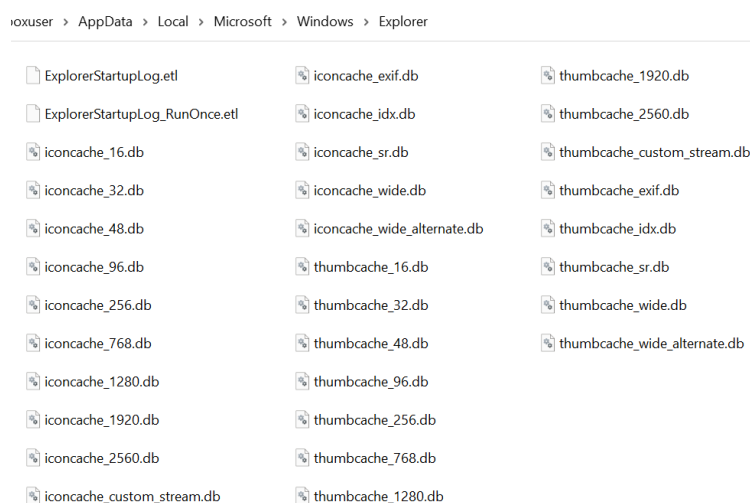


Abbildung 4.5.: Thumbcache-Ordner eines Windows 11 Systems

Wie in den Abbildungen 4.4 und 4.5 zu sehen ist, sind in den Ordnern nicht nur die Dateien des Thumbnail Cache angesiedelt. Zusätzliche Dateien, wie die des Windows Icon Cache, sind hier ebenfalls vorhanden. Diese zusätzlichen Dateien sind nicht Teil der Thesis.

Die Thumbcache-Datenbank *thumbcache_exif.db* dient nach McKeown et al. [22] dazu, die in die EXIF Informationen der Originaldateien eingebetteten Thumbnails aufzunehmen.

Zu den Thumbcache-Datenbanken

- thumbcache_custom_stream.db
- thumbcache_wide.db
- thumbcache_wide_alternate.db

konnten während der Durchführung der Thesis keine Informationen zu deren Zweck oder Funktion gewonnen werden. Sie enthalten lediglich den Standard Fileheader für Thumbcache-Datenbanken. Während der Experimente fand keine Veränderung des Ausgangszustands der Dateien statt.

4.4.2. Aufbau der Thumbcache-Datenbanken

Die Thumbcache-Datenbanken von Windows 10 und 11 haben den gleichen Aufbau wie die von Metz [25] dokumentierten Versionen von Windows Vista und Windows 7. Nach einem einleitenden Dateiheader folgen die einzelnen Thumbcache-Einträge, die ihrerseits über einen eigenen Eintragsheader verfügen, nach dem die Thumbnail Bilddatei folgt. Die Bedeutung einzelner Datenfelder der Header hat dagegen mehrere Änderungen erfahren.

Datenbank Header

Windows 10 und Windows 11 verwenden zueinander identische Dateiheader von 24 Bytes Länge und entsprechen so den von Metz dokumentierten Dateiheadern von Windows Vista und Windows 7. Auch in Anzahl und Größe der Datenfelder bestehen hier keine Differenzen. Die Nutzung des vierten und letzten Datenfeldes unterscheidet sich jedoch von der von Metz festgestellten Bedeutung der Felder.

Wert	Beschreibung
0x00	thumbcache_16.db
0x01	thumbcache_32.db
0x02	thumbcache_48.db
0x03	thumbcache_96.db
0x04	thumbcache_256.db
0x05	thumbcache_768.db
0x06	thumbcache_1280.db
0x07	thumbcache_1920.db
0x08	thumbcache_2560.db
0x09	thumbcache_sr.db
0x0A	thumbcache_wide.db
0x0B	thumbcache_exif.db
0x0C	thumbcache_wide_alternate.db
0x0D	thumbcache_custom_stream.db

Tabelle 4.10.: Cache Versionen für Windows 10 und Windows 11

Offset zum Dateifooter

Die deutlichste Änderung des Dateih-Headers der Thumbcache-Datenbanken unter Windows 10 und Windows 11 gegenüber Windows Vista und Windows 7 betrifft das letzte Datenfeld des Dateih-Headers. Während Metz [25] hier Informationen zur Anzahl der in der Datei vorhandenen Cache Einträge dokumentiert, stellt bei Windows 10 und Windows 11 dieser 4 Bytes lange Eintrag einen Offset vom Dateibeginn aus dar. An der im Offset angegebenen Adresse befindet sich ein 56 Byte langer Dateifooter, der das Ende der vorhandenen Einträge markiert.

Der Dateifooter und damit die zugehörige Offsetangabe im Dateih-Header sind in jeder Thumbcache-Datenbank vorhanden, die über Einträge verfügt. Bei leeren Datenbanken dagegen beträgt der Offsetwert immer 0x18, die Länge des Dateih-Headers.

Dateifooter

Die 56 Bytes des Dateifooters unterteilen sich in folgende Datenfelder:

Offset (Byte)	Größe (Byte)	Wert	Beschreibung
0	4	CMMM	Signatur
4	4		Anzahl der Bytes bis zum Dateiende, ausgehend vom Beginn des Footers
8	40		unbekannt, immer 0x00
48	8		unbekannt

Tabelle 4.11.: Thumbcache-Datenbank Footer bei Windows 10 und Windows 11

Bei jeder untersuchten Thumbcache-Datenbank folgten nach dem Footer ausschließlich 0x00 Werte bis zum Dateiende.

Ein beispielhafter Durchlauf vom Dateiheder bis zum Dateiende findet sich im Anhang A.2.

4.4.3. Einträge in der Thumbcache-Datenbank

Die Einträge in den Thumbcache-Datenbanken von Windows 10 und Windows 11 folgen dem Muster der Vorgängerversionen. Jeder Eintrag verfügt über einen eigenen Eintragsheader. Entsprechend der im Dateiheder angegebenen Format Version 0x20 weisen die Eintragsheader eine Länge von 56 Bytes auf, unterscheiden sich jedoch in den vorhandenen Datenfeldern von dem ursprünglichen Headerformat *Windows Vista*. Der Eintrag, der vormals die Dateierweiterung enthielt, ist nicht mehr vorhanden. Die dadurch weggefallenen 8 Bytes finden sich, mit unterschiedlicher Aufteilung und Verwendung, vor den Prüfsummen für Daten und Eintragsheader.

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Signatur (CMMM)
4	4	Gesamtgröße des Cache Eintrags
8	8	Thumbcache-ID
16	4	Größe des Eintragsnamens (identifier)
20	4	Größe des Paddings
24	4	Größe der Daten
28	4	Breite des Thumbnails in Pixeln
32	4	Höhe des Thumbnails in Pixeln
36	4	Unbekannt
40	8	Prüfsumme der Daten (CRC-64)
48	8	Prüfsumme des Eintragsheaders (CRC-64)
56	Größe des Eintragsnamens	Eintragsname (UTF-16 String)
...	Paddinggröße	Padding aus 0x00 Bytes (nur vorhanden wenn Paddinggröße > 0)
...	Datengröße	Daten (Thumbnail Bilddaten)

Tabelle 4.12.: Windows 10 / 11: Eintragsheader einer Thumbcache-Datenbank

4.4.4. Aufbau der Indexdatei

Auch in der Indexdatei ergeben sich Änderungen gegenüber der Dokumentation von Metz [25] verändert. Deren Aufbau ist bei Windows 10 und Windows 11 identisch.

Dateiheader

Während Metz die Länge des Dateiheders mit 24 Bytes angibt, ist er bei den beiden betrachteten Betriebssystemen auf 144 Bytes angewachsen. Erst nach dieser Marke findet man den ersten Indexeintrag. Auch die Indexeinträge selbst sind von 40 Bytes (Windows Vista), bzw. 32 Bytes (Windows 7) auf 72 Bytes angewachsen, wobei das angegebene Cache Format weiterhin 0x20 (Windows Vista) lautet.

Um die Funktion der unbekannten Datenfelder zu erkennen, wurden die Einträge des Dateiheders der Indexdatei mit den in Thumbcache Viewer [18] und im Hexeditor [26] geöffneten Thumbcache-Datenbanken verglichen und gezielt Manipulationen getätigt. Durch diese an beiden untersuchten Betriebssystemen durchgeführten Experimente, wurde die Funktion diverser Datenfelder im Dateiheader und den In-

dexeinträgen geklärt, wobei jedoch nicht für alle Datenfelder schlüssige Erkenntnisse gewonnen werden konnten.

Der Aufbau des Dateiheaders der Indexdatei ist in Anhang A.3 nachzuschlagen.

Indexeinträge

Die Indexeinträge messen bei Windows 10 und Windows 11 52 Bytes Länge. Neben dem Dateiheader ist also auch hier ein Zuwachs gegenüber früheren Versionen zu verzeichnen. Obwohl im Dateiheader die Format Version *Windows Vista* (0x20) angegeben ist, stimmt der Aufbau der Indexeinträge nicht mit dieser überein. Die Angabe des Zeitpunkts der letzten Modifikation und die von Metz [25] dokumentierten Flags sind nicht mehr vorhanden.

Um die Funktion der bestehenden Datenfelder zu erkennen, wurden Testdateien sukzessive mit wechselnden Vorschauansichten betrachtet und die Veränderungen in der Indexdatei verfolgt. Hierzu wurden mehrere Testdateien mit je einer Vorschaureihenfolge betrachtet: von der kleinsten zur größten Vorschau und umgekehrt.

Die Erkenntnisse sind in Anhang A.4 nachzuschlagen.

4.5. Nutzungsverhalten der Thumbcache-Datenbanken

Im Rahmen der durchgeführten Versuche am Windows Thumbnail Cache konnte festgestellt werden, dass bei beiden Betriebssystemen die involvierten Dateien auf dem Dateisystem einem bestimmten Verhaltensmuster folgen.

Nach der Installation des Betriebssystems weisen die Datenbankdateien eine Größe von 24 Bytes (Dateiheader) bis maximal 1024 KB auf. Zu diesem Zeitpunkt vorhandene Thumbnails stammen von Dateien des Betriebssystems. Die Zeitstempel der Thumbcache-Dateien (Erstellung und letzte Änderung) zeigen dabei den Installationszeitpunkt des Betriebssystems. Beim Befüllen der Dateien bleiben die Zeitstempel konstant, solange eine Datenbankdatei nicht vergrößert werden muss. Die Vergrößerung erfolgt dabei in Schritten von 1024 KB, zeitgleich dazu wird der Stempel der letzten Änderung aktualisiert.

Je nach Benutzerverhalten sind so Thumbcache-Datenbanken mit unterschiedlichen Änderungszeitstempeln vorhanden, die vom Installationszeitpunkt bis zum aktuellen Datum reichen können (siehe Abbildung 4.7).
















 thumbcache_16.db	08.11.2022 21:55	1.024 KB
 thumbcache_32.db	31.05.2023 07:41	3.072 KB
 thumbcache_48.db	01.02.2023 15:14	2.048 KB
 thumbcache_96.db	13.07.2023 09:42	38.912 KB
 thumbcache_256.db	30.06.2023 09:26	20.480 KB
 thumbcache_768.db	30.06.2023 06:43	6.144 KB
 thumbcache_1280.db	05.07.2023 08:57	36.864 KB
 thumbcache_1920.db	24.05.2023 08:17	1.024 KB
 thumbcache_2560.db	07.07.2023 11:12	16.384 KB
 thumbcache_custom_stream.db	08.11.2022 21:55	1 KB
 thumbcache_exif.db	19.12.2022 15:10	1.024 KB
 thumbcache_idx.db	01.06.2023 15:21	910 KB
 thumbcache_sr.db	08.11.2022 21:55	1 KB
 thumbcache_wide.db	08.11.2022 21:55	1 KB
 thumbcache_wide_alterate.db	08.11.2022 21:55	1 KB

Abbildung 4.7.: Windows 10: Natürlich gewachsener Thumbnail Cache

Werden Thumbcache-Datenbankdateien gelöscht, z.B. manuell oder durch die Festplattenbereinigung, ersetzt das Betriebssystem diese durch Dateien ohne Inhalt. Der Zeitstempel der Dateierstellung wird dabei jedoch nicht mit der aktuellen Zeit befüllt, sondern zeigt erneut den Installationszeitpunkt. Das Datum der letzten Änderung hingegen zeigt die aktuelle Zeit.

5. Forschungsumgebung und verwendete Software

5.1. Forschungsumgebung

Die Umsetzung der Experimente im Rahmen der Thesis, erfolgte in einer virtualisierten Umgebung, da durch den Einsatz von Snapshots und virtuellen Festplatten ein schneller Wechsel in den Ausgangszustand ermöglicht wird. Dies reduzierte vor allem den zeitlichen Aufwand bei der Durchführung der Experimente.

Als Hostsystem kam Debian 11 zum Einsatz. Die Spezifikationen des Hosts sind Anhang A.5 zu entnehmen.

Die Virtualisierung erfolgte mit Virtualbox 7.0.8 unter Verwendung des Oracle Virtualbox Extension Pack 7.0.8r156879 und den zugehörigen Gasterweiterungen.

5.2. Virtuelle Maschinen

Sowohl Windows 10 als auch Windows 11 wurden mit weitgehend identischen Konfigurationen der virtuellen Maschinen (VM) betrieben. Diese sind in Anhang A.6 gelistet.

Die VMs wurden am 20.06.2023 installiert und mit den zu diesem Zeitpunkt verfügbaren Updates versehen. Im Anschluss wurde die virtuelle Netzverbindung getrennt, um weitere Updates zu unterbinden und so Veränderungen des Betriebssystems während der Experimente zu verhindern. Details zu den eingesetzten Betriebssystemen sind in Anhang A.7 vermerkt. Bei Versuchen, die eine Onlineverbindung voraussetzen, wurde die Netzverbindung gewährt. Es wurden lokale Benutzerkonten verwendet, um mögliche Probleme von Onlinekonten durch die fehlende Internetverbindung zu vermeiden.

5.3. Verwendete Software

Die zur Umsetzung der Thesis verwendete Software besteht neben frei verfügbaren Werkzeugen auch aus proprietären Produkten, die untenstehend aufgelistet werden.

Name	Zweck	Version
DB Browser for SQLite [21]	Betrachten von SQLite Datenbanken	3.12.2
ESEDatabaseView [36]	Betrachten von ESE Datenbanken	1.73
HxD [26]	HEX-Editor	2.5.0.0 (x86-64)
paint.net [3]	Bildbearbeitung	5.0.7
RegistryChangesView [37]	Untersuchung der Windows Registry	1.2.9
Thumbcache Viewer [18]	Einsehen von Thumbcache Datenbanken	1.0.3.7
TrueNAS CORE 2023 [15]	NAS Software	TrueNAS-13.0-U5.1
Virtualbox [39]	Virtualisierungsumgebung	7.0.8
WinPrefetchView [38]	Untersuchung der Windows Prefetch Dateien	1.3.7
WinSearchDBAnalyzer [17]	Datenwiederherstellung in der Windows.edb	1.0.0.6
X-Ways Forensics [53]	Forensische Auswertungen	20.8

Tabelle 5.1.: Auflistung der verwendeten Software

6. Analyse der Mechanismen

Ziel der Analyse der Thumbcache-Mechanismen ist es festzustellen, auf welche Weise verschiedene Nutzeraktionen das Entstehen von Einträgen im Windows Thumbnail Cache beeinflussen. Durch die gewonnenen Erkenntnisse ist es unter Umständen möglich, bei der forensischen Untersuchung von Asservaten von dem Zustand des Thumbnail Cache auf Aktivitäten des Benutzers zu schließen.

6.1. Vorschaufunktionen des Datei-Explorers

Wie in 3.3 beschrieben, verfügt der Datei-Explorer über verschiedene Optionen, um Vorschaubilder von Dateien anzeigen zu lassen. Im Folgenden wird für jede Vorschauoption untersucht, welche der 14 Thumbcache-Datenbanken sie beeinflusst. Im Laufe der Versuche wurde festgestellt, dass die Fenstergröße des Datei-Explorers Auswirkungen auf den Thumbcache bei der Nutzung des Vorschaufensters hat. Aus der enormen Anzahl von Möglichkeiten, die Fenstergröße zu wählen, ergeben sich ebenso viele Möglichkeiten den Thumbcache zu beeinflussen. Aus diesem Grund wird in der Thesis die Vorschaufensterfunktion ausschließlich bei maximiertem Fenster untersucht.

6.1.1. Vorgehen

Die Untersuchung erfolgt in zwei Schritten. Zunächst wird im Datei-Explorer jede Vorschauoption für sich allein betrachtet. Hierzu wird eine Testdatei mit einer Vorschauoption angezeigt und im Anschluss nach Veränderungen in den Thumbcache-Datenbanken gesucht. Um auszuschließen, dass unterschiedliche Vorschauoptionen einander in der Wirkung beeinflussen, erfolgt vor jedem Testdurchlauf ein Zurücksetzen der virtuellen Maschine auf den Ausgangszustand.

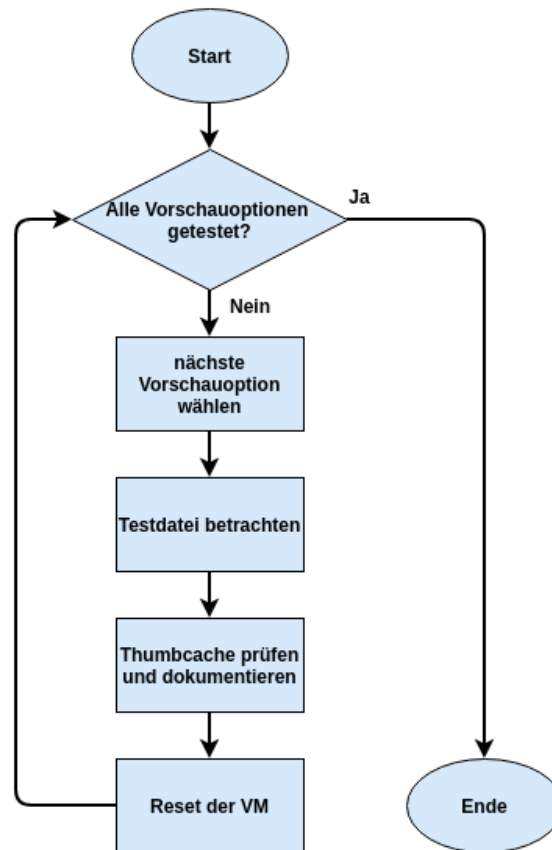


Abbildung 6.1.: Ablauf des ersten Untersuchungsschrittes

Im zweiten Schritt werden Wechselwirkungen zwischen den einzelnen Vorschaup Optionen untersucht. Dazu wird die Testdatei im Datei-Explorer von der geringsten zur höchsten Vorschaudarstellung betrachtet. Nach jedem Wechsel der Vorschau erfolgt eine Prüfung der Thumbcache-Datenbanken und der Indexdatei nach Änderungen bei den Einträgen der Testdatei. Dieses Vorgehen wird nach dem Zurücksetzen der virtuellen Maschine mit umgekehrter Vorschaureihenfolge wiederholt und die Testdatei von der höchsten bis zur niedrigsten Vorschaudarstellung betrachtet.

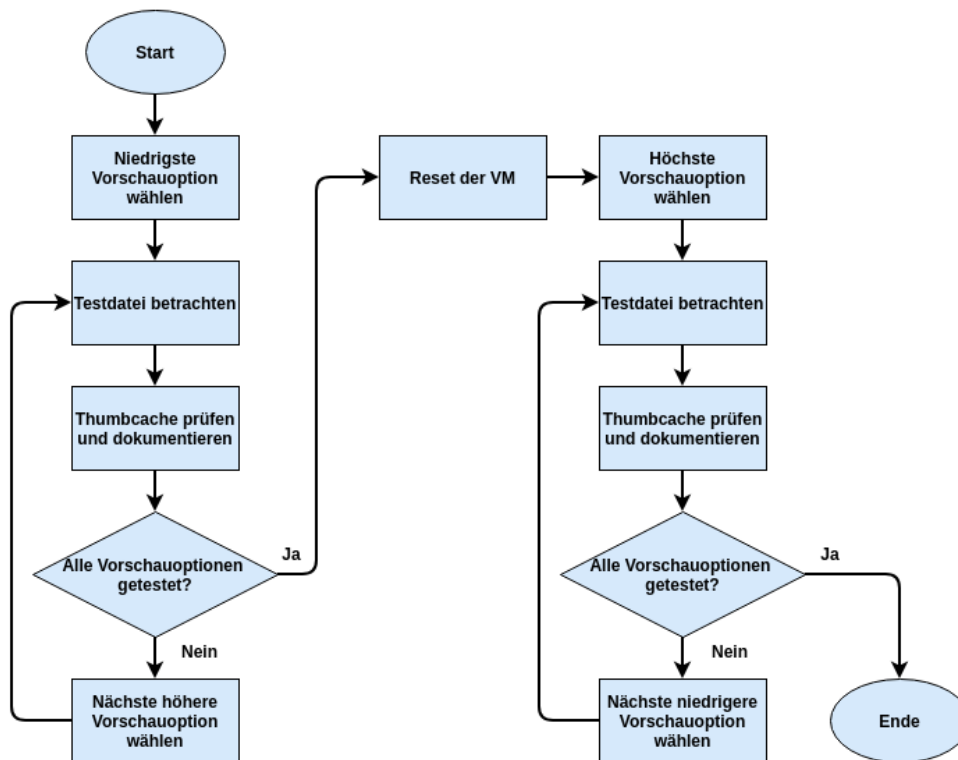


Abbildung 6.2.: Ablauf des zweiten Untersuchungsschrittes

Zur Durchführung der Tests wurden ursprünglich folgende Dateien verwendet (jeweils im JPEG und PNG Format):

Beschreibung	Dimensionen Breite × Höhe (Pixel)
Quadrat	50 × 50
Quadrat	500 × 500
Quadrat	5000 × 5000
Logo der HS Wismar, FIW	1181 x 1543

Tabelle 6.1.: Liste der ursprünglich verwendeten Testdateien

Die Dateien wurden dabei im Bilderverzeichnis des Benutzerprofils abgelegt.

Im Lauf der Experimente wurden weitere Testdateien hinzugefügt, um Feststellungen verifizieren zu können. Diese sind in Tabelle 6.2 aufgeführt.

Beschreibung	Dimensionen Breite × Höhe (Pixel)
Schmales Rechteck	300 × 500
Breites Rechteck	500 × 300
Logo der HS Wismar, FIW	skaliert auf 1000 × 1000

Tabelle 6.2.: Liste weiterer Testdateien

Die Testdateien wurden derart gewählt, um einen möglichen Einfluss der Quelldatei auf den Thumbcache identifizieren zu können (Größe, Dateiformat, Bildinhalt). Jede Testdatei befindet sich in einem eigenen Dateiordner, der sich wiederum in einem Hauptordner befindet.

6.1.2. Betrachtung der einzelnen Vorschauoptionen

Die Untersuchung der einzelnen Vorschauoptionen ergab folgende Ergebnisse, wobei beide betrachteten Betriebssysteme identische Werte lieferten:

Vorschauoption	Betroffene Thumbcache-Datei
Inhalt	thumbcache_32.db
Kacheln	thumbcache_48.db
Mittelgroße Symbole	thumbcache_48.db
Große Symbole	thumbcache_256.db
Extra Große Symbole	thumbcache_256.db

Tabelle 6.3.: Vorschauoptionen und die von ihnen veränderten Thumbcache-Dateien

Hierauf haben die Bilddimensionen, Dateiformate und Bildinhalte keinen Einfluss.

Die Vorschauoptionen

- Liste
- Details
- Kleine Symbole

verursachen keine Einträge im Thumbcache.

Sonderfall Vorschaufenster

Als Sonderfall zeigt sich die Anzeige im Vorschaufenster. Während ein maximal großes Vorschaufenster Änderungen in der thumbcache_2560.db verursacht, sind die betroffenen Thumbcache-Dateien bei minimalem Vorschaufenster abhängig von den Dimensionen des Originalbildes.

Quadratische und breit rechteckige Grundformen hinterlassen Werte in thumbcache_256.db und thumbcache_1280.db. Handelt es sich jedoch um eine schmal rechteckige Grundform, wird lediglich die thumbcache_1280.db verändert.

6.1.3. Betrachtung der Wechselwirkungen

Bei diesen Experimenten wurden nur noch die Vorschauoptionen berücksichtigt, die im vorangegangenen Schritt Einträge im Thumbcache hinterlassen haben. Als Testdateien wurden alle Dateien aus dem vorherigen Schritt verwendet.

Die Resultate der Versuche sind für Windows 10 und Windows 11 identisch.

Von der kleinsten zur höchsten Vorschauoption

Der Verlauf der gewählten Vorschauoptionen wurde wie folgt festgelegt:

Inhalt → Kacheln → Mitttelgroße Symbole → Große Symbole → Extra Große Symbole → Minimales Vorschaufenster → Maximales Vorschaufenster

Alle Testdateien erbrachten identische Ergebnisse, unabhängig von Bildinhalt, Dimensionen oder Bildgröße:

Vorschauoption	32	48	256	1280	2560
Inhalt	×				
Kacheln	×	×			
Mitttelgroße Symbole	×	×			
Große Symbole	×	×	×		
Extra Große Symbole	×	×	×		
Minimales Vorschaufenster	×	×	×	×	
Maximales Vorschaufenster (Endzustand)	×	×	×	×	×

Tabelle 6.4.: Änderungsverlauf des Thumbcache *klein* → *groß*

In keiner der übrigen Thumbcache-Datenbanken wurden Änderungen ausgelöst.

Von der höchsten zur kleinsten Vorschauoption

Hier wurde der Verlauf der Vorschauoptionen umgekehrt zum Vorangegangenen gewählt:

Maximales Vorschaufenster → Minimales Vorschaufenster → Extra Große Symbole → Große Symbole → Mitttelgroße Symbole → Kacheln → Inhalt

Die Umkehrung der Reihenfolge machte sich in den Thumbcache-Einträgen bemerkbar. Zum einen erfolgten keine Einträge mehr in der thumbcache_1280.db. Zum an-

deren erzeugten schmal rechteckige Grundformen keine Einträge in der thumbcache_256.db, wenn das minimale Vorschaufenster genutzt wurde, während breit rechteckige oder quadratische Grundformen einen Eintrag in dieser Datenbank setzen.

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster			×		×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt	×	×	×		×

Tabelle 6.5.: Änderungsverlauf des Thumbcache *groß* → *klein* bei quadratischen / breit rechteckigen Dateien

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster					×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt	×	×	×		×

Tabelle 6.6.: Änderungsverlauf des Thumbcache *groß* → *klein* bei schmal rechteckigen Dateien

6.1.4. Bewertung

Die durchgeführten Untersuchungen zielten darauf ab, die Frage nach dem Einfluss der einzelnen Vorschauoptionen des Datei-Explorers zu beantworten.

Die Resultate zeigen, dass die jeweiligen Thumbnail Caches von Windows 10 und Windows 11 in gleicher Weise auf Änderungen der Vorschauoptionen im Windows Explorer reagieren. Bildinhalte und die generelle Größe der Bilder haben auf diesen Prozess keinen Einfluss. Die Form der Bilddateien dagegen bestimmt das Verhalten der Thumbcache-Funktion, wenn Dateien im Vorschaufenster des Datei-Explorers betrachtet werden.

Aus den Erkenntnissen wurden insgesamt sechs Muster in Tabellenform erstellt, die

das Verhalten einzelner Vorschauoptionen und deren Kombination aufzeigen. Die Muster sind in Anhang A.8 zu finden.

6.2. Einfluss des Speicherortes der Originaldateien

Bei der vorangegangenen Untersuchung der Vorschauoptionen des Datei-Explorers befanden sich die Testdateien konstant im Bilderverzeichnis des Benutzerprofils. Bilddateien können jedoch an beliebigen Orten des Dateisystems gespeichert werden. Die Speicherung auf zusätzlichen oder portablen Datenträgern ist ebenso möglich, wie die Nutzung von Clouddiensten. Deshalb wird im Folgenden der Blick auf den Einfluss des Speicherortes auf das Verhalten des Windows Thumbnail Cache gelegt.

Vorgehen

Aus der Untersuchung der Vorschauoptionen ist bekannt, dass weder die Bildgröße, das Dateiformat noch Bildinhalte Einfluss auf das Verhalten des Thumbcache ausüben. Auswirkungen hat lediglich die geometrische Form der Bilddateien. Aus diesem Grund umfasst das Set an Testdateien für die nachfolgenden Untersuchungen nur noch drei Dateien im PNG Format.

Beschreibung	Dimensionen Breite × Höhe (Pixel)
Quadrat	500 × 500
Schmales Rechteck	300 × 500
Breites Rechteck	500 × 300

Tabelle 6.7.: Liste der verwendeten Testdateien

Diese werden an festgelegten Orten im Dateisystem der beteiligten Datenträger abgelegt und anschließend mit wechselnden Vorschauoptionen im Datei-Explorer betrachtet. Zunächst liegt das Augenmerk auf den einzelnen Vorschauoptionen und anschließend auf möglichen Wechselwirkungen derselben. Die Untersuchungsschritte sind also identisch zu den in Punkt 6.1.1 beschriebenen Verfahren.

6.2.1. Lokale Speicherung

Im Folgenden werden die Auswirkungen der Speicherung von Bilddateien auf lokalen Datenträgern untersucht. Der Ordner mit den Testdateien befand sich bei der Durchführung der Experimente an folgenden ausgewählten Speicherorten innerhalb und außerhalb des Benutzerprofils:

- C:\
- C:\Benutzer\Benutzername\
- C:\Benutzer\Benutzername\Desktop\
- C:\Benutzer\Benutzername\Dokumente\
- C:\Benutzer\Benutzername\Downloads\
- C:\Benutzer\Benutzername\Videos\
- C:\Windows\System32\
- E:\ (zusätzliche Festplatte)

Ergebnis

In beiden Betriebssystemen können Dateien auf dem Desktop in drei Vorschaugrößen angezeigt werden: Kleine Symbole, Mittlere Symbole, Große Symbole. Bei der Betrachtung der Testdateien gab es keine Abweichungen von dem Verhalten des Thumbnail Cache im Datei-Explorer bei gleicher Vorschauoption.

An den übrigen Speicherorten konnten ebenfalls keine Abweichungen zu den Mustertabellen in Anhang A.8 festgestellt werden.

6.2.2. Cloudspeicher

Bereits 2020 nutzten 35 % der Deutschen Clouddienste zum Speichern und Teilen ihrer Daten und liegen damit im europäischen Durchschnitt [2]. Anlässlich dieser Statistik werden im Folgenden die drei meistgenutzten Clouddienste [47] hinsichtlich ihrer Auswirkungen auf den Windows Thumbnail Cache untersucht.

Vorgehen

Für jeden Cloudanbieter wurde eine eigene virtuelle Maschine erstellt. Die Installation der nötigen Software erfolgte mit den vorgelegten Standardeinstellungen. Die Testdateien wurden anschließend in das vorgegebene Verzeichnis der jeweiligen Cloudsoft-

ware kopiert. Die Untersuchung des Thumbnail Cache erfolgte analog zu dem bisherigen Vorgehen. Da für diese Untersuchung eine Onlineverbindung nötig war, wurde diese den virtuellen Maschinen gewährt. Die Betriebssysteme wurde dabei nicht mit Updates versorgt.

Google Drive

Google Drive erscheint als neues Laufwerk G:\ im Betriebssystem. Die Testdateien wurden darin im vorgegebenen Order *Ablage* gespeichert. Die in beiden Betriebssystemen gleiche Standardkonfiguration der Cloudsoftware bewirkt, dass Dateien in diesem Ordner nur online gespeichert und erst bei Bedarf auf den Datenträger übertragen werden. Die Vorschau im Datei-Explorer erfolgt ausschließlich als Quadrat. Nicht quadratische Bilddateien werden dabei nur teilweise abgebildet. Von schmal rechteckigen Bilddateien entfällt z.B. der untere Bildteil, wie in Abbildung 6.3 zu sehen ist. Die Darstellung im Vorschaufenster ist hiervon ebenfalls betroffen.

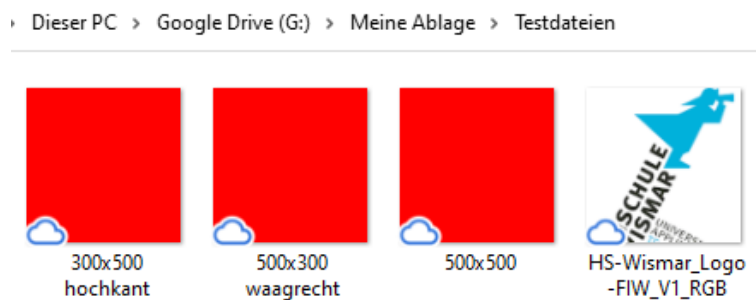


Abbildung 6.3.: Windows 11: Durchgehend quadratische Darstellung bei Google Cloud Dateien

Durch diesen Umstand werden alle Bilddateien vom Thumbcache behandelt, als wäre ihre Grundform quadratisch. So konnte bei den Versuchen beobachtet werden, dass schmal rechteckige Grundformen identische Thumbcache-Einträge erzeugen wie quadratische oder breit rechteckige Grundformen. Die Untersuchungsergebnisse sind für beide Betriebssysteme identisch zu denen quadratischer und breit rechteckiger Dateien.

Dropbox

Die Dropbox Software verfügt über zwei Nutzungspläne. Die kostenlose Standardvariante spiegelt lokale Dateien in den Onlinespeicher. So sind alle Dateien auch Offline verfügbar. Die Plus Variante bietet die Möglichkeit, Daten ausschließlich Online zu speichern. Bei der Installation wurde die kostenlose Standardvariante gewählt.

In dieser Konfiguration erstellt die Dropbox Software einen Dateiordner *Dropbox* direkt im Benutzerverzeichnis, dessen Inhalt in den Onlinespeicher gespiegelt wird. Hier wurde der Ordner mit den Testdateien platziert.

Das Untersuchungsergebnis weicht nicht von dem bisher beobachteten Verhalten des Thumbnail Cache ab und unterscheidet sich auch nicht bei den betrachteten Betriebssystemen.

Microsoft OneDrive

Die in beide Windowsversionen integrierte Software zur Nutzung der Microsoft Cloud *OneDrive* ist standardmäßig so konfiguriert, dass Dateien im Onlinespeicher abgelegt und nur bei Bedarf heruntergeladen werden. Zur Synchronisierung von Nutzerdaten wird ein Dateiordner *OneDrive* direkt im Benutzerverzeichnis erstellt. In diesem wurden die Testdateien platziert.

Die einzelnen Vorschauoptionen und deren Wechselwirkungen untereinander weichen nicht vom bisher beobachteten Verhalten des Thumbnail Cache ab. Auch konnten keine Unterschiede zwischen den beiden untersuchten Betriebssystemen festgestellt werden.

Ergebnis

Das Verhalten des Windows Thumbnail Cache weicht bei keinem der drei Cloudanbieter von dem in den Mustertabellen (Anhang A.8) dokumentierten Verhalten ab. Eine Besonderheit stellt die Cloud des Anbieters Google dar. Hier wird für sämtliche Bilddateien eine quadratische Vorschau im Datei-Explorer erzeugt. Dadurch verhalten sich auch schmal rechteckige Dateien, die im Normalfall leicht abweichende Einträge im Thumbcache erzeugen, wie quadratische Dateien. Windows 10 und Windows 11 verhalten sich bei der Cloudnutzung identisch.

6.2.3. Externe Datenträger

Der Umstand, dass im Jahr 2022 alleine auf dem deutschen Markt 10,85 Millionen USB-Sticks an Konsumenten verkauft wurden [12] macht deutlich, welchen Stellenwert USB-Datenträger haben. Aus diesem Grund wird im Folgenden untersucht, ob die Speicherung auf einem USB-Datenträger das Verhalten des Windows Thumbnail Cache beeinflusst.

Vorgehen

Um die Untersuchung umzusetzen, wurde ein USB-Stick (SanDisk Ultra USB 3.0) über das Geräte-Menü mit der jeweiligen virtuellen Maschine verbunden. Um zu prüfen, ob die virtuelle Maschine den USB-Datenträger tatsächlich als Wechselmedium behandelt, wurde die Logfunktion für USB-Geräte im Ereignisprotokoll aktiviert (Anwendungs- und Dienstprotokolle → Microsoft → Windows → DriverFrameworks-UserMode → Betriebsbereit) aktiviert. Nachdem dies sichergestellt war, wurde mit den Versuchen an den Testdateien begonnen, die auf dem USB-Stick abgelegt waren.



Abbildung 6.4.: Windows 11: Ereignismeldung zum Laden des Treibers für Wechsel-datenträger

Ergebnis

Bei den Versuchen wurden keine Abweichungen von dem bisher dokumentierten Verhalten des Windows Thumbnail Cache und den Tabellen des Anhang A.8 festgestellt. Beide Betriebssysteme verhielten sich dabei identisch.

6.2.4. Netzwerkspeicher

Eine weitere Möglichkeit, Daten zu speichern und zuzugreifen, stellen Netzwerkspeicher dar. Deren möglicher Einfluss auf den Windows Thumbnail Cache wird im Folgenden untersucht.

Vorgehen

Zur Umsetzung wurde eine virtuelle Maschine mit dem frei verfügbaren NAS System TrueNAS erstellt und deren Speicherfreigabe als Netzlaufwerk in Windows 10 und Windows 11 eingebunden. Anschließend folgte die Überprüfung des Verhaltens des Thumbnail Cache mithilfe der auf dem Netzlaufwerk abgelegten Testdateien.

Ergebnis

Das Verhalten des Thumbnail Cache weicht hier bei beiden Betriebssystemen teilweise deutlich von den bisherigen Feststellungen ab. Bei der Betrachtung der einzelnen Vorschauoptionen wurde keine Thumbcache-Datenbank unterhalb der thumbcache_256.db genutzt. Auch die beobachteten Wechselwirkungen bei einem Vorschauwechsel von der kleinsten zur größten Option weist Unterschiede auf. Lediglich beim Wechsel von der größten zur kleinsten Vorschauoption waren keine Unterschiede zu den bisherigen Ergebnissen festzustellen. Auch hier produzieren schmal rechteckige Bilddateien Thumbcache-Einträge, die von denen von quadratischen oder breit rechteckigen Bilddateien abweichen.

Die dokumentierten Ergebnisse sind dabei für Windows 10 und Windows 11 identisch und sind in Tabellenform in Anhang A.9 dokumentiert.

6.2.5. Bewertung

Die durchgeführten Untersuchungen hatten zum Ziel, einen möglichen Einfluss des Speicherortes der Originaldateien auf das Verhalten des Windows Thumbnail Cache festzustellen. Hierbei wurden unterschiedliche Speicherorte innerhalb und außerhalb des Benutzerverzeichnisses auf einem lokalen Datenträger, die Speicherung auf externen Datenträgern und Netzlaufwerken sowie die Nutzung von Clouddiensten in Betracht gezogen.

Die gewonnenen Erkenntnisse zeigen, dass der Speicherort der Originaldateien grundsätzlich keinen Einfluss auf den Windows Thumbnail Cache hat. Einträge in den Thumbcache-Datenbanken erfolgen immer nach dem Muster, wie es schon bei der Untersuchung der einzelnen Vorschauoptionen erkannt wurde. Lediglich bei Dateien, die sich auf Netzlaufwerken befinden, gibt es Abweichungen vom üblichen Verhalten der Funktion des Thumbnail Cache.

Die Sonderstellung von schmal rechteckigen Originaldateien ist bei jedem anderen Speicherort ebenfalls zu beobachten.

Windows 10 und Windows 11 verhielten sich dabei in jedem der Versuche identisch zueinander.

6.3. Sonstige Erkenntnisse

Während der Durchführung der Versuchsreihen wurde festgestellt, dass abseits der gewählten Vorschauoptionen und auch ohne Nutzeraktivität Einträge in die Thumbcache-Datenbanken aufgenommen wurden. Diese Feststellungen betreffen Windows 10 und Windows 11 gleichermaßen.

6.3.1. Inhaltsvorschau von Dateiordnern

Wird ein Ordnersymbol mit einer der in Tabelle 6.3 angegebenen Vorschauoptionen betrachtet, so wird das Symbol so verändert, dass eine Vorschau des Ordnerinhalts sichtbar ist (siehe Abbildung 6.5).

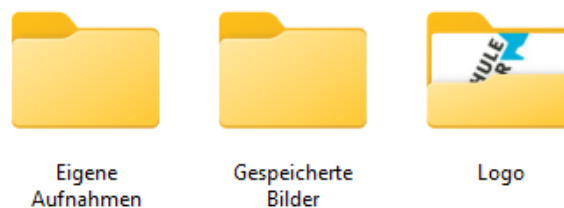


Abbildung 6.5.: Windows 11: Vorschau eines Ordnerinhalts

Ist als Vorschauoption Inhalt, Kacheln oder Mittelhöhe Symbole gewählt und befindet sich nur eine einzelne Bilddatei in dem Ordner, so wird die enthaltene Bilddatei als Vorschau in die Datei thumbcache_256.db aufgenommen. Ordner mit mehreren enthaltenen Bilddateien verursachen keine Thumbcache-Einträge.

Die Vorschauoptionen Große und Extra Große Symbole verursachen auch für Ordner mit mehreren enthaltenen Bilddateien Thumbcache-Einträge. Jedoch nur für die im Ordnersymbol angezeigte Bilddatei.

So werden Vorschaubilder im Thumbcache abgelegt, ohne dass der Ordner, in dem die Originaldateien gespeichert sind, selbst geöffnet wurde.

6.3.2. Drag-and-drop

Werden Bilddateien per drag-and-drop mit der Maus bewegt, so verursacht dies einen Eintrag in der thumbcache_96.db.



Abbildung 6.6.: Windows 10: Vorschau bei Drag-and-Drop

6.3.3. Automatisierte Thumbnails

Es wurde beobachtet, dass Einträge in den Thumbcache-Datenbanken thumbcache_16.db, thumbcache_48.db und thumbcache_256.db erstellt wurden, obwohl keine vorausgehende Nutzeraktion erfolgte. Dieses Verhalten war auch zu beobachten, wenn das Betriebssystem nach dem Start nicht weiter genutzt wurde. Die Erstellung der Thumbcache-Einträge erfolgt nicht sofort, nachdem die Dateien gespeichert wurden, sondern in zeitlichem Abstand und nicht allgemeingültig für alle vorhandenen Dateien. Betroffen sind ausschließlich Bilddateien, die im Bilderverzeichnis des Benutzerprofils gespeichert sind. Dieser Umstand wurde verifiziert, indem Testdateien auch außerhalb dieses Verzeichnisses platziert wurden.

Da die Thesis ausschließlich Auswirkungen von Nutzeraktionen betrachtet, wurde dieses Verhalten nicht tiefergehend analysiert.

6.3.4. Bewertung

Die Beobachtungen zeigen, dass Thumbcache-Einträge unabhängig von den Vorschauoptionen oder dem direkten Betrachten von Vorschaubildern entstehen können.

Vor allem das automatisierte Erstellen von Thumbnails für Bilddateien im Bilderverzeichnis, obwohl nach dem Systemstart keinerlei weitere Nutzeraktion erfolgte, ist für die forensische Arbeit relevant.

7. Gewinnung von Metadaten

Der forensische Mehrwert des Microsoft Windows Thumbnail Cache ist, dass dieser Thumbnails von Dateien enthalten kann, die im Original bereits gelöscht wurden. Auch Thumbnails von Dateien auf Datenträgern, die nicht mehr verfügbar sind, können darin nachweisbar sein [16]. Dies betrifft z.B. Dateien auf Wechseldatenträgern oder Netzlaufwerken, wie bei Untersuchungen im vorangegangenen Kapitel festgestellt wurde.

Nachteilig hierbei ist, dass in den Thumbcache-Datenbanken keine Metadaten zu den Originaldateien vorhanden sind (siehe 4). Für die forensische Auswertung wichtige Informationen, wie Dateinamen oder Speicherpfade, können aus dieser Quelle also nicht bestimmt werden.

Im Folgenden wird untersucht, ob und wie entsprechende Metadaten bei Windows 10 und Windows 11 gewonnen werden können.

7.1. Stand der Forschung

Sowohl Kävrestad [16] als auch Morris und Chivers [32, 33] nennen die zentrale Datenbank des Dienstes *Windows Search* als potenzielle Quelle, um Metadaten zu Thumbcache-Einträgen zu gewinnen.

Diese ist unter dem Speicherpfad

`%systemdrive%\ProgramData\Microsoft\Search\Data\Applications\Windows\`

als Datei *Windows.edb* zu finden [6].

In dieser ESE-Datenbank kann die Thumbcache-ID aus den Thumbcache-Datenbanken genutzt werden, um Metadaten zu dem entsprechenden Thumbnail zu finden [16, 32, 33].

7.2. Vorgehen

Für die beiden Betriebssysteme wurde je eine virtuelle Maschine erstellt. Anschließend wurden vier Bilddateien (JPG Format) im Datei-Explorer betrachtet, so dass sie Einträge im Thumbnail Cache hinterlassen. Bei diesen Bilddateien handelte es sich um:

- Bilddatei im Bilderverzeichnis (ungelöscht)
- Bilddatei im Bilderverzeichnis (nach der Betrachtung gelöscht)
- Bilddatei auf einem USB-Stick (nach der Betrachtung entfernt)
- Bilddatei auf einem Netzlaufwerk (nach der Betrachtung getrennt)



Abbildung 7.1.: Testdatei für das Bilder Verzeichnis

Die beiden VMs wurden anschließend heruntergefahren, deren virtuelle Festplatten im Hostsystem eingebunden und sowohl die Dateien des Thumbcache als auch die Dateien von Windows Search exportiert.

Das Mounten der virtuellen Festplatten wurde mit dem Terminalbefehl

```
vboximg-mount -i HDUUID -o allow_root /tmp/vdi/
```

durchgeführt, wobei HDUUID für die jeweilige ID der virtuellen Festplatte steht.

Das Mounten der nun verfügbaren Volumes (Partitionen) erfolgte mit dem Befehl

```
sudo mount -o ro VOLUME MOUNTPOINT
```

Die forensische Betrachtung der Datenexporte erfolgte in einer dritten VM unter Windows 10. Hierbei wurden die Programme Thumbcache Viewer und ESEDatabaseView genutzt. Bei der Betrachtung von Windows 11 wurde festgestellt, dass die ESE-Datenbank durch eine SQLite Datenbank ersetzt wurde, weshalb hier die Software DB Browser for SQLite zum Einsatz kam.

```
$ vboximg-mount -i b408f8f4-b1d8-4dba-86b7-a1393cadd4d -o allow_root /tmp/vdi
$ ls /tmp/vdi
vhdd vol0 vol1 vol2 vol3 W11_Stock.vdi
$ sudo mount /tmp/vdi/vol3 /mnt/w11
[sudo] Passwort für rba:
$ ls /mnt/w11
'$Recycle.Bin'          'Program Files (x86)'
'$WinREAgent'          Programme
'Dokumente und Einstellungen' Recovery
DumpStack.log.tmp      swapfile.sys
pagefile.sys           'System Volume Information'
PerfLogs               Users
ProgramData            vboxpostinstall.log
'Program Files'        Windows
$
```

Abbildung 7.2.: Einbinden der virtuellen Festplatte zur Analyse

7.3. Ergebnis Windows 10

Die Überprüfung der Thumbcache-Datenbanken ergab, dass alle vier Testdateien darin durch Thumbnails repräsentiert wurden. Auch die nach dem Betrachten im Vorschaumodus gelöschte Datei befand sich darin.

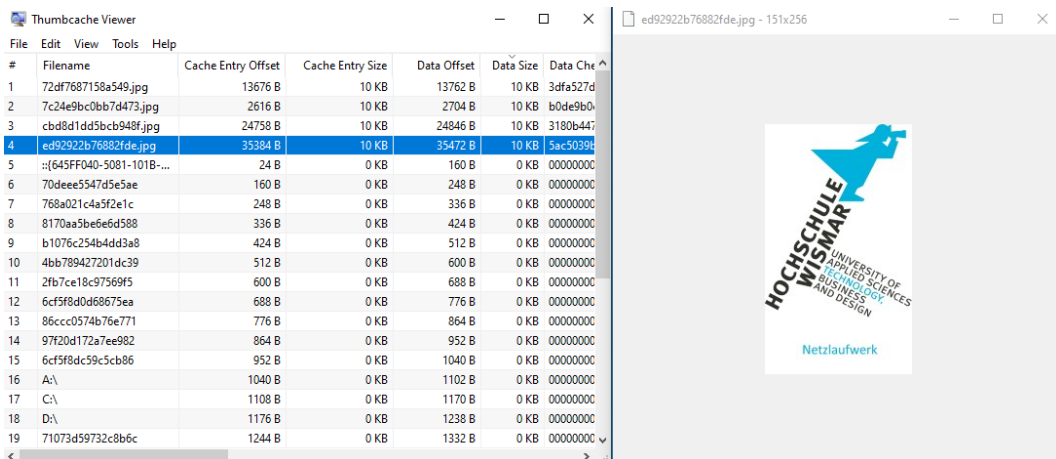


Abbildung 7.3.: Windows 10: Betrachtung eines Thumbnails im Thumbcache Viewer

Um nach den vier festgestellten Thumbcache-IDs zu recherchieren, wurde die Windows Search Datenbank Windows.edb in das Programm ESEDatabaseView importiert.

Die Suche erfolgte in allen vorhandenen Tabellen, wobei nur in der Tabelle *SystemIndex_PropertyStore* ein Treffer erzielt wurde.

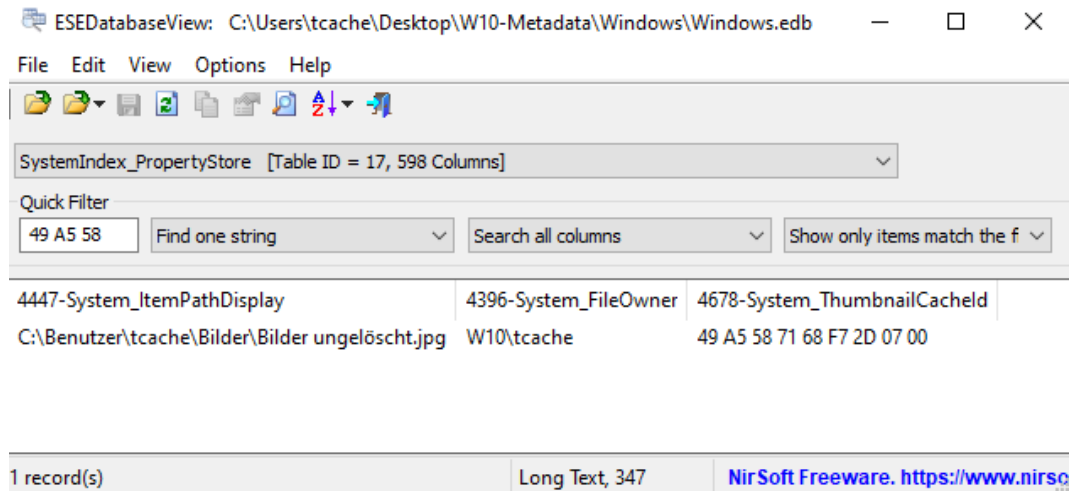


Abbildung 7.4.: Windows 10: Suchtreffer in ESEDatabaseView

Der einzelne Metadateneintrag in der Datei Windows.edb bezog sich auf die im Bilderordner abgelegte Testdatei, die nicht gelöscht wurde.

Um auch möglicherweise gelöschte Einträge in der Windows.edb mit einzubeziehen, wurde diese mit der Software WinSearchDBAnalyzer aufbereitet. Der Software gelang es, den Eintrag der aus dem Bilderordner gelöschten Testdatei wiederherzustellen und so deren Metadaten zugänglich zu machen.

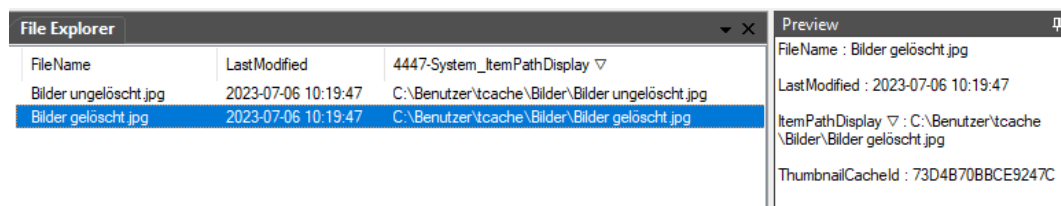


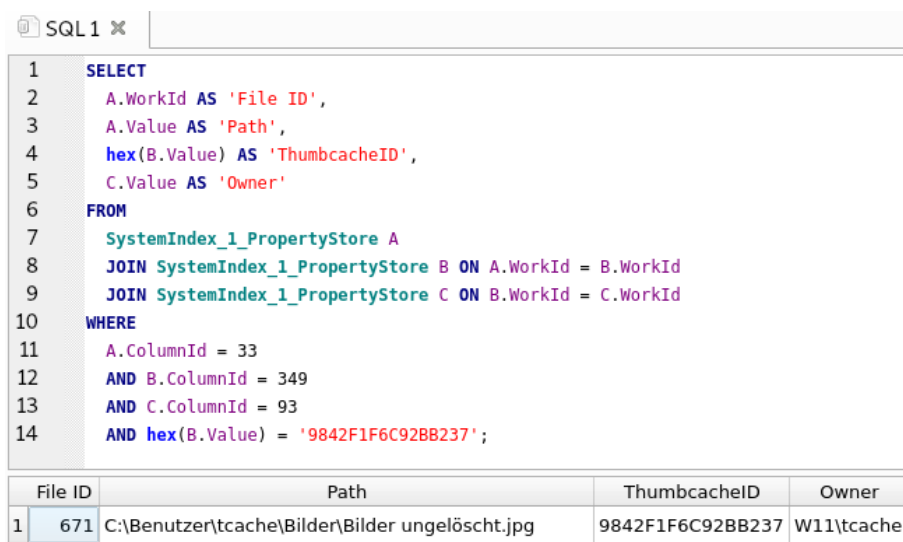
Abbildung 7.5.: WinSearchDBAnalyzer: Wiederhergestellter Eintrag

Da die Thumbcache-IDs als Hexadezimalwerte in der Windows.edb vorliegen, wurde abschließend mit einem Hexeditor nach den fehlenden beiden IDs gesucht. Hierbei wurde nach den vollständigen Thumbcache-IDs und Teilen davon gefiltert, wobei keine Treffer erzielt wurden.

7.4. Ergebnis Windows 11

Auch unter Windows 11 waren alle vier Testdateien als Thumbnails in den Thumbcache-Datenbanken vorhanden.

Zur Suche nach den vier Thumbcache-IDs wurde ein SQLite Query entworfen und in der Datei Windows.db angewendet. Das Query gibt zu einer gesuchten Thumbcache-ID den Speicherpfad und den Besitzer der Originaldatei sowie die interne Datenbank ID der Originaldatei aus. Bei der Suche nach den vier IDs der Testdateien wurde nur der Eintrag der Datei ausgegeben, die sich ungelöscht im Bilderordner des Benutzerprofils befand.



The screenshot shows a window titled 'SQL 1' with a query editor and a results table. The query is as follows:

```

1 SELECT
2   A.WorkId AS 'File ID',
3   A.Value AS 'Path',
4   hex(B.Value) AS 'ThumbcacheID',
5   C.Value AS 'Owner'
6 FROM
7   SystemIndex_1_PropertyStore A
8   JOIN SystemIndex_1_PropertyStore B ON A.WorkId = B.WorkId
9   JOIN SystemIndex_1_PropertyStore C ON B.WorkId = C.WorkId
10 WHERE
11   A.ColumnId = 33
12   AND B.ColumnId = 349
13   AND C.ColumnId = 93
14   AND hex(B.Value) = '9842F1F6C92BB237';

```

The results table below the query shows one entry:

	File ID	Path	ThumbcacheID	Owner
1	671	C:\Benutzer\tcache\Bilder\Bilder ungelöscht.jpg	9842F1F6C92BB237	W11\tcache

Abbildung 7.6.: Windows 11: Suchtreffer in DB Browser for SQLite

Auch bei Windows 11 wurden die Thumbcache-IDs der fehlenden Bilddateien mit einem Hexeditor in der Datei Windows.db gesucht. Dabei wurden sowohl die kompletten IDs als auch Teile davon verwendet. Die Suche verlief negativ.

7.5. Untersuchung von Windows Search

Die bei den Untersuchungen getroffenen Feststellungen stehen teilweise im Gegensatz zu den Aussagen der aktuellen Literatur (siehe 7.1):

Die Windows Search Datenbank wird als Quelle für Metadaten gelöschter Dateien und von Dateien auf nicht mehr verfügbaren Datenträgern (z.B. Wechseldatenträger, Netzlaufwerke, verschlüsselte Volumes) genannt. Im vorliegenden Fall konnten bei Windows 10 daraus aber nur Metadaten zu der ungelöschten und der gelöschten

Testdatei im Bilderverzeichnis des Benutzerprofils gewonnen werden. Bei Windows 11 tatsächlich nur für die nicht gelöschte Datei. Aus diesem Grund wurde bei beiden Betriebssystemen die Funktion von Windows Search näher betrachtet.

Dabei wurde festgestellt, dass Windows Search sowohl bei Windows 10 als auch bei Windows 11 standardmäßig so konfiguriert ist, dass nur Daten erfasst werden, die sich im Benutzerprofil oder im Startmenü befinden.

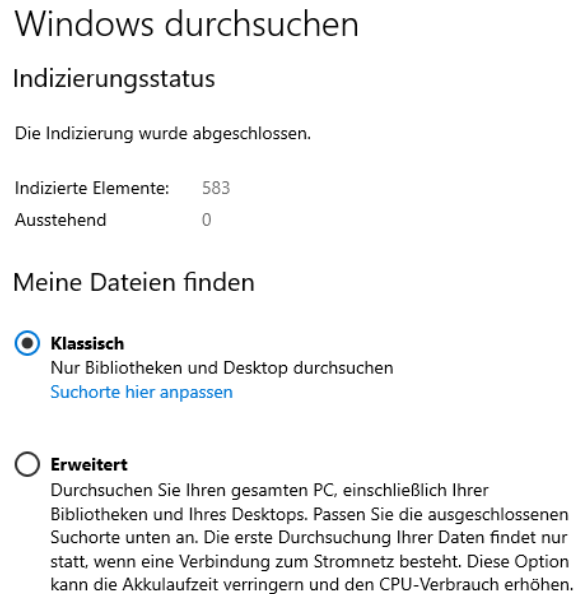


Abbildung 7.7.: Windows 10: Standardkonfiguration Windows Search

Es ist zwar möglich, sämtliche Speicherorte des Systems Windows Search zugänglich zu machen (Option *Erweitert*), dies bezieht sich aber nur auf lokale Datenträger. Verbundene Wechseldatenträger müssen in einem gesonderten Menü aktiviert werden, um auch diese in der Suche zu erfassen. Bei Netzlaufwerken ist dies nicht möglich. Sie sind Windows Search nicht zugänglich.

Diese Standardkonfiguration verhinderte also, dass die Bilddateien auf dem USB-Stick und dem Netzlaufwerk von Windows Search während der Versuche berücksichtigt wurden.

Durch Beobachtung des Indizierungsstatus, der auf Bild 7.7 zu sehen ist, konnte festgestellt werden, dass bei beiden Betriebssystemen Windows Search in Echtzeit arbeitet. Sobald eine Datei hinzugefügt oder entfernt wird, passt sich die Anzeige entsprechend an. Bei Windows 10 hat dieser Umstand nur geringe Auswirkungen, da gelöschte Einträge aus der ESE Datenbank Windows.edb wiederhergestellt werden können. Während der Experimente war dies bei Windows 11 und der hier verwendeten

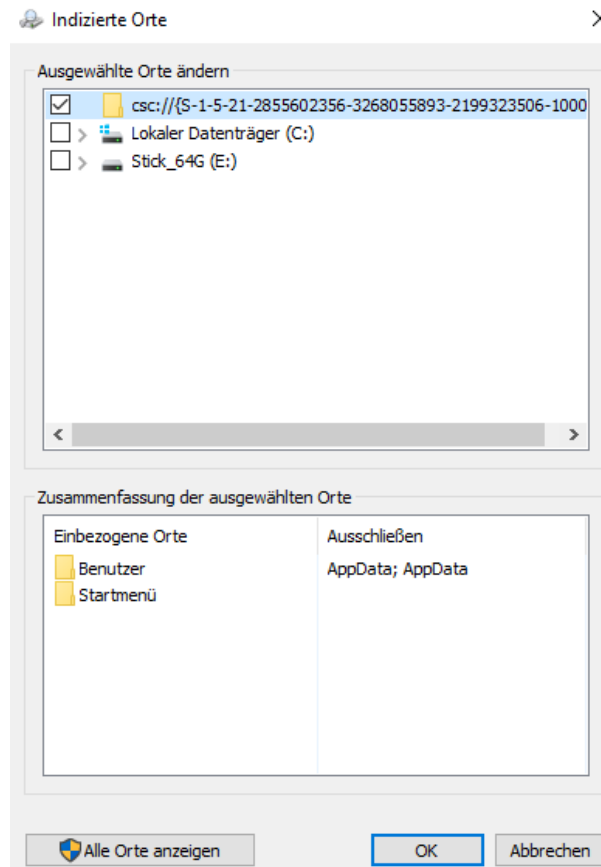


Abbildung 7.8.: Windows 10: Hinzufügen eines Wechseldatenträgers zu Windows Search

SQLite Datenbank Windows.db nicht möglich. Da seit SQLite Version 3.6.12 (März 2010) die Option *secure_delete* standardmäßig aktiviert ist [44], ist anzunehmen, dass dadurch auch in der Windows.db gelöschte Einträge mit Nullen überschrieben werden. Dadurch sind diese Daten nicht mehr der forensischen Auswertung zugänglich.

Um dies zu verifizieren, wurde eine Testdatei in das Dokumenten-Verzeichnis eines Benutzerprofils kopiert und die Speicherbereiche der zugehörigen Thumbcache-ID innerhalb der Windows.db vor und nach dem Löschen der Originaldatei betrachtet. Vor dem Betrachten wurde der Windows Search Dienst in einem Terminalfenster beendet (`net stop wsearch`). Durch das Experiment konnte festgestellt werden, dass die Speicherbereiche, in denen ursprünglich die Thumbcache-ID der Testdatei lag, nach deren Löschung entweder mit neuen Daten beschrieben oder gänzlich durch 0x00 belegt waren.

Die Einträge gelöschter Dateien werden also in der Windows.db unwiederbringlich gelöscht. Es finden sich nur noch Metadaten zu nicht gelöschten Dateien.

7.6. Weitere Quellen zur Gewinnung von Metadaten

Um weitere Quellen zu finden, anhand derer Metadaten zu vorhandenen Thumbcache-Einträgen gewonnen werden können, wurde eine Suche nach den Thumbcache-IDs in X-Ways Forensics durchgeführt. Hierzu wurden die virtuellen Festplatten als Datenträgerimages der Software hinzugefügt und über das Suchmenü *Parallele Suche* nach den Thumbcache-IDs gesucht. Dabei wurden die Textform und die Hexadezimalwerte, jeweils in Big und Little Endian Byte-Reihenfolge, als Suchbegriffe verwendet.

Bei dieser Suche konnten für beide Betriebssysteme außerhalb der Windows.edb und Windows.db keine Treffer erzielt werden, aus denen sich Metadaten für die Thumbcache-Einträge ergeben.

7.7. Bewertung

Die Durchführung der Untersuchungen hatte zum Ziel, Quellen zur Gewinnung von Metadaten zu den Einträgen des Windows Thumbnail Cache identifizieren.

Bei Windows 10 wurden entsprechende Daten ausschließlich in der Datenbank des Windows Search Dienstes vorgefunden. Durch dessen Standardkonfiguration beinhaltet diese aber lediglich Metadaten zu Dateien, die sich in einem Verzeichnis oder dem Startmenü eines Benutzerkontos befinden. Zuverlässig auffinden lassen sich Metadaten zu ungelöscht vorhandenen Bilddateien. Metadaten von zwischenzeitlich gelöschten Bilddateien sind insoweit verfügbar, als deren gelöschte Einträge in der Windows.edb noch rekonstruiert werden können.

Auch bei Windows 11 konnten Metadaten nur aus der Datenbank des Windows Search Dienstes gewonnen werden. Diese beschränken sich auf Metadaten zu ungelöscht vorhandenen Bilddateien auf internen Datenträgern. Dies liegt zum einen an der zu Windows 10 identischen Standardkonfiguration des Windows Search Dienstes. Zum anderen am Wechsel der Datenbanktechnik von ESE zu SQLite, bei der die Wiederherstellung von gelöschten Einträgen verhindert wird, da diese mit Nullen überschrieben werden.

So hält sich die Möglichkeit, nützliche Metadaten zu erlangen, bei beiden Betriebssystemen in engen Grenzen. Während bei Windows 10 zumindest noch Metadaten zu aus dem Benutzerprofil gelöschten Dateien gewonnen werden können, bietet Windows 11 diese Möglichkeit nicht mehr.

8. Manipulationsmöglichkeiten

Viele Computernutzer sind sich nicht bewusst, dass ihre Aktionen Spuren im Betriebssystem hinterlassen [52]. Dennoch stellen Techniken der Antiforensik ein immer größeres Hindernis für die IT-Forensik dar [7]. Im Folgenden wird deshalb auf die zur Verfügung stehenden Möglichkeiten eingegangen, die digitalen Spuren des Windows Thumbnail Cache zu verwischen, falsche Spuren zu setzen oder gar nicht erst entstehen zu lassen. Wege, die Anwendung solcher Techniken kennen, werden ebenfalls behandelt. Die nachfolgenden Ausführungen betreffen beide Betriebssysteme gleichermaßen, da bei den Untersuchungen keinerlei Differenzen zwischen Windows 10 und Windows 11 festgestellt werden konnten.

8.1. Deaktivierung des Thumbcache

Das Entstehen von Daten von Beginn an zu verhindern, ist Teil der Antiforensik. Rogers [43] siedelt dieses Vorgehen in der Kategorie Artifact Wiping (Vernichten von Daten) an. Im Folgenden werden Möglichkeiten aufgezeigt, wie Windows 10 oder Windows 11 System so konfiguriert werden können, dass keine Einträge im Thumbnail Cache erstellt werden. Anschließend wird erläutert, wie diese Konfigurationen bei der forensischen Auswertung erkannt werden können.

8.1.1. Vorgehen

In je einer virtuellen Maschine wurden Windows 10 und Windows 11 mit der entsprechenden Methode konfiguriert, nachdem der vorhandene Thumbnail Cache geleert wurde. Das Leeren wurde durch das manuelle Entfernen der Thumbcache-Datenbanken umgesetzt. Nachdem die Systeme konfiguriert waren, keine Einträge im Thumbnail Cache mehr zu erstellen, wurden Bilddateien mit sämtlichen Vorschauoptionen im Datei-Explorer betrachtet. Auch die Option des Vorschaufensters wurde genutzt. Dabei wurde auf Veränderungen der Dateigrößen und der Inhalte der Thumbcache-Datenbanken geachtet.

8.1.2. Methode 1: Explorer-Optionen

Bei dieser Methode wird das im Datei-Explorer integrierte Konfigurationsmenü verwendet, um die Nutzung des Thumbnail Cache zu deaktivieren. Hierzu wird unter

Optionen → Reiter *Ansicht*

die Option *Immer Symbole statt Miniaturansichten zeigen* aktiviert.

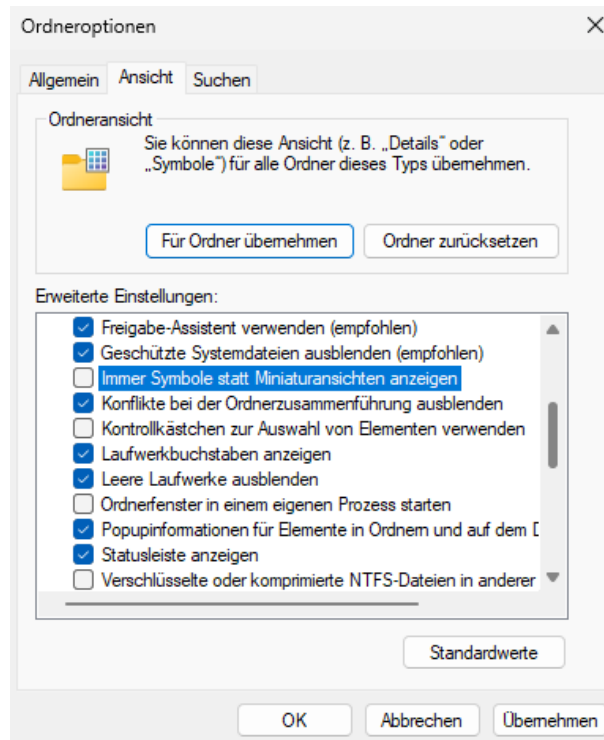


Abbildung 8.1.: Windows 11: Optionsmenü des Datei-Explorers

8.1.3. Methode 2: Systemeinstellungen

Diese Methode zur Deaktivierung des Thumbnail Cache bedient sich der zentralen Systemsteuerung des Windows Systems. Dazu wird unter

Systemsteuerung → System und Sicherheit → System → Erweiterte Systemeinstellungen → Reiter *Erweitert* → Einstellungen (Leistung) → Visuelle Effekte

die Option *Miniaturansichten anstelle von Symbolen anzeigen* deaktiviert.

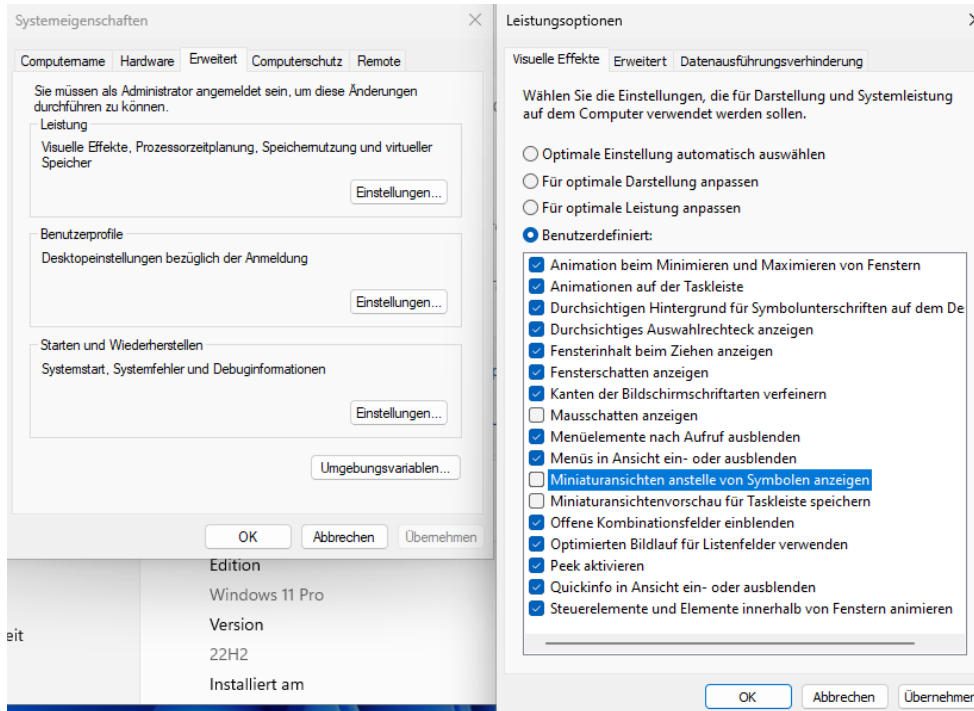


Abbildung 8.2.: Windows 11: Thumbcache deaktivieren über die Systemsteuerung

8.1.4. Methode 3: Gruppenrichtlinie

Die letzte Methode verwendet Gruppenrichtlinien, um die Nutzung des Thumbnail Cache zu deaktivieren. Hierzu existieren zwei Administrative Vorlagen, die diesen Zweck erfüllen, jedoch unterschiedlich wirken. Sie befinden sich im Gruppenrichtlinien Editor (gpedit.msc) unter

Benutzerkonfiguration → Administrative Vorlagen → Windows-Komponenten → Datei-Explorer.

Die Gruppenrichtlinie *Zwischenspeicherung von Bildern in Miniaturansicht deaktivieren* bewirkt, dass zwar im Datei-Explorer weiterhin Vorschaubilder angezeigt werden, diese jedoch nicht im Thumbcache zwischengespeichert werden.

Die Gruppenrichtlinie *Anzeige von Miniaturansichten deaktivieren und nur Symbole anzeigen* bewirkt, dass auch im Datei-Explorer keine Vorschaubilder angezeigt werden. Es sind nur Dateisymbole zu sehen und es erfolgen keine Einträge im Thumbnail Cache.

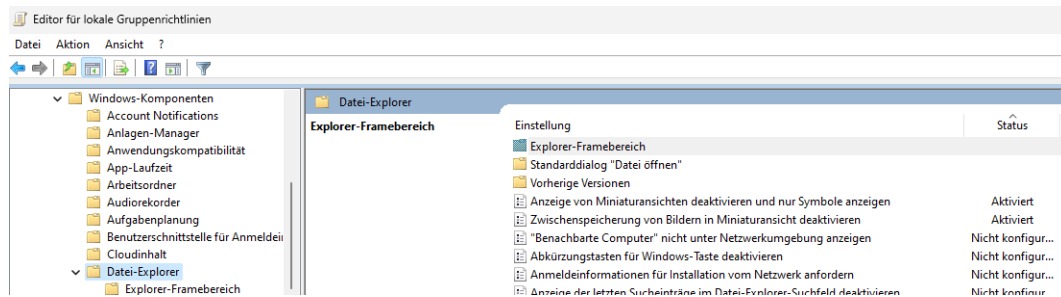


Abbildung 8.3.: Windows 11: Aktivierte Gruppenrichtlinien

8.1.5. Kontrolle der Wirksamkeit der Konfigurationen

Jede der getesteten Methoden unterdrückt bei Windows 10 und Windows 11 zuverlässig die Nutzung des Thumbnail Cache. In keiner Thumbcache-Datenbank konnten Thumbnails vorgefunden werden. Der Datei-Explorer zeigt dabei lediglich bei der zweiten getesteten Gruppenrichtlinie Vorschaubilder der Testdateien an. Die übrigen Methoden deaktivieren neben dem Thumbnail Cache auch die Vorschaufunktion des Datei-Explorers. Selbst im Vorschauenfenster wird lediglich das Dateisymbol der markierten Testdatei dargestellt.

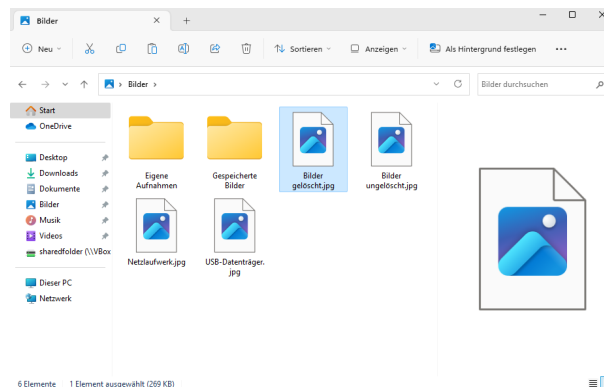


Abbildung 8.4.: Windows 11: Deaktivierter Thumbcache beim Datei-Explorer

8.1.6. Nachweis der Konfigurationsänderungen

Bei Windows Betriebssystemen werden die für die Konfiguration des Betriebssystems relevanten Einstellungen in der Registry gespeichert [9]. Zur Identifizierung der Registry Schlüssel, die die Nutzung des Windows Thumbnail Cache verhindern, wurde die Software RegistryChangesView verwendet. Um die Funktionalität der dabei festgestellten Schlüssel zu verifizieren, erfolgte anschließend deren Überprüfung durch manuelles Verändern ihrer Werte.

Die vorgestellten Methoden 1 und 2 können in der Datei NTUSER.DAT nachgewiesen werden, die sich im Stammverzeichnis von Benutzerprofilen befindet. In dieser Datei werden die Einstellungen des jeweiligen Benutzerkontos gespeichert [10]. Sie wird im Registry Editor als Hive HKEY_CURRENT_USER dargestellt [8]. Bei Aktivierung einer der beiden Methoden wird der dort vorhandene Schlüssel *IconsOnly* verändert. Trägt der Schlüssel den Wert 1, so deaktiviert dies die Nutzung des Windows Thumbnail Cache. Es erfolgen keine Einträge in den Thumbcache-Datenbanken. Zu finden ist der Schlüssel in der NTUSER.DAT unter dem Pfad:

Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\

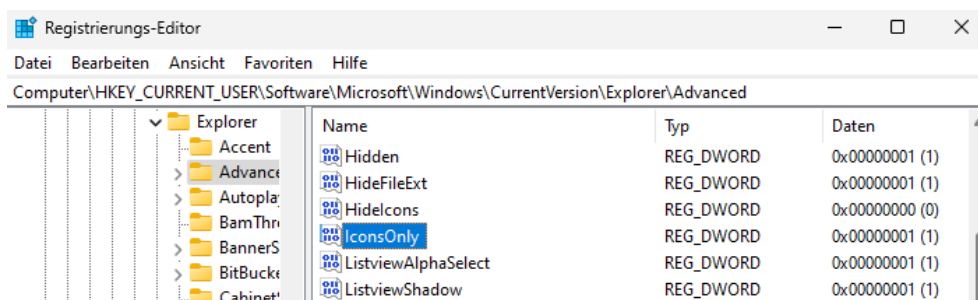


Abbildung 8.5.: Windows 11: Registry Schlüssel *IconsOnly*

Auch Methode 3 verändert bei der Aktivierung der beiden genutzten Gruppenrichtlinien die Datei NTUSER.DAT. Dabei nutzt *Anzeige von Miniaturansichten deaktivieren und nur Symbole anzeigen* den Schlüssel *DisableThumbnails*, während die Gruppenrichtlinie *Zwischenspeicherung von Bildern in Miniaturansicht deaktivieren* den Schlüssel *NoThumbnailCache* verwendet. Beide Schlüssel werden beim Aktivieren ihrer zugehörigen Gruppenrichtlinie unter dem Pfad

Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\

erstellt und ihr Wert jeweils auf 1 gesetzt.

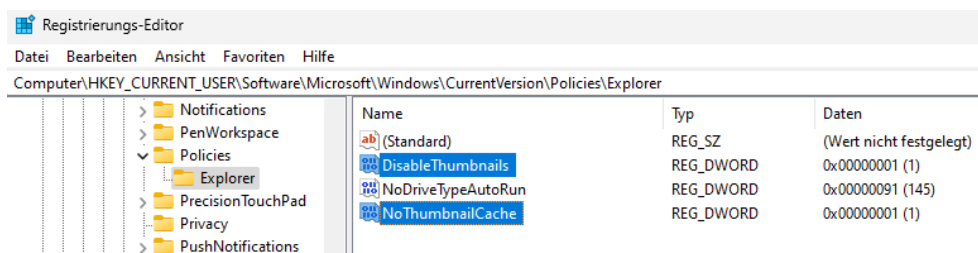


Abbildung 8.6.: Windows 11: Registry Schlüssel der beiden Gruppenrichtlinien

8.1.7. Bewertung

Die durchgeführten Untersuchungen zeigen, dass der Windows Thumbnail Cache mit den Bordmitteln des Betriebssystems deaktiviert werden kann. Zur Umsetzung verfügen beide Betriebssysteme über die gleichen Möglichkeiten, die jeweils zuverlässig verhindern, dass Einträge in den Thumbnail Cache aufgenommen werden.

Ebenfalls bei beiden Betriebssystemen kann die Deaktivierung des Thumbnail Cache in der Registrydatei NTUSER.DAT nachgewiesen werden, die im Benutzerprofil gespeichert ist.

Es ist anzumerken, dass jede der oben genannten Methoden zur Deaktivierung des Thumbnail Cache zwar seine weitere Nutzung effektiv unterbindet, bereits darin gespeicherte Inhalte jedoch nicht gelöscht werden. Es werden nur keine neuen Inhalte hinzugefügt.

8.2. Löschen der Thumbcache-Datenbanken

Entstandene Daten zu entfernen, ist Teil des Artifact Wiping (Vernichten von Daten) [43] und kann auch auf die Dateien des Windows Thumbnail Cache angewandt werden. Microsoft Windows bietet hierfür mehrere Wege, Thumbcache-Datenbanken zu entfernen, die im Folgenden beleuchtet werden. Die hierfür nötigen Experimente fanden pro Betriebssystem und Bereinigungstechnik in gesonderten virtuellen Maschinen statt.

8.2.1. Nutzung der Datenträgerbereinigung

Eine Möglichkeit, die Daten des Windows Thumbcache zu entfernen, ist die sog. Datenträgerbereinigung. Diese Wartungssoftware wird zur Entfernung von temporären und Systemdateien genutzt, um Speicherplatz freizugeben [27]. Eine dabei angebotene Option ist es, die Miniaturansichten (Thumbnail Cache) zu löschen.

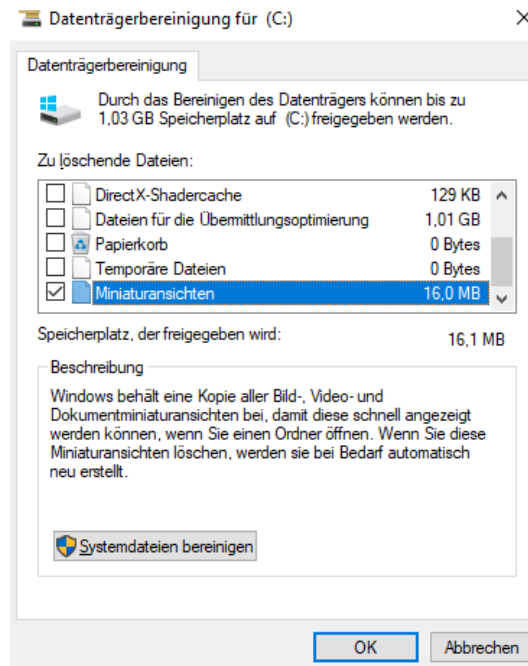


Abbildung 8.7.: Windows 10: Windows Datenträgerbereinigung

Auswirkungen

Nach der Ausführung der Datenträgerbereinigung enthalten fast alle Dateien des Windows Thumbnail Cache keine Daten mehr. Einzelne Thumbcache-Datenbanken beinhalten entweder nur den Dateihheader oder wenige Einträge zu Bilddateien des Betriebssystems, deren Verwaltungsdaten dann auch in der Indexdatei vorhanden sind.

Eine tiefergehende Betrachtung mit X-Ways Forensics ergab, dass sich die Sektoren, in denen die Dateien des Thumbnail Cache gespeichert sind, vor und nach der Ausführung der Datenträgerbereinigung unterscheiden. Die Dateien werden dem entsprechend nicht nur geleert, sondern als Ganzes neu erstellt, während die bisherigen Dateien durch Löschung entfernt werden. Ein Umstand, der auch von McKeown et al. [22] festgestellt wurde. Die neuen Dateien behalten dabei den Zeitstempel des Installationszeitpunktes des Betriebssystems, während das Änderungsdatum den Zeitpunkt der durchgeführten Festplattenbereinigung zeigt.

Nachweis der Datenträgerbereinigung

Der Einsatz der Datenträgerbereinigung kann durch die Analyse der Windows Prefetch Dateien belegt werden. Windows Prefetch bietet unter anderem Informationen über die letzten Ausführungszeiten und wie oft ein Programm insgesamt ausgeführt wurde [23]. Das ausführende Benutzerkonto kann mithilfe der Aufzeichnungen der Windows

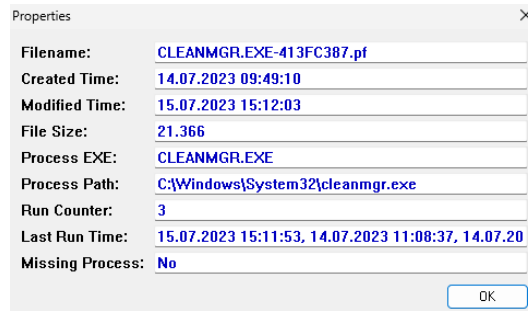


Abbildung 8.8.: WindowsPrefetchView: Details der cleanmgr.exe

Funktion Background Activity Moderator (BAM) bestimmt werden. BAM setzt bei der Ausführung von Programmen einen Eintrag in der Registrydatei SYSTEM und weist diesen anhand der SID dem ausführenden Benutzerkonto zu. Diese Informationen können unter dem Pfad

ControlSet001\Services\bam\State\UserSettings\{SID}

eingesehen werden.

Values BamDam	
Drag a column header here to group by that column	
Program	Execution Time
Program	=
windows.immersivecontrolpanel_cw5n1h2bxyewy	2023-07-15 14:09:15
\Device\HarddiskVolume4\Windows\explorer.exe	2023-07-15 14:20:47
\Device\HarddiskVolume4\Windows\System32\ApplicationFrameHost.exe	2023-07-15 14:09:15
MicrosoftWindows.Client.WebExperience_cw5n1h2bxyewy	2023-07-15 14:20:46
\Device\HarddiskVolume4\Windows\System32\cleanmgr.exe	2023-07-15 13:12:26
Microsoft.XboxGamingOverlay_8wekyb3d8bbwe	2023-07-14 09:07:46

Abbildung 8.9.: Registry Explorer: Details der cleanmgr.exe

Eine weitere, wenn auch nicht immer erfolgsversprechende Form, die Nutzung der Datenträgerbereinigung nachzuweisen, stützt sich auf Feststellungen von Morris und Chivers [32], die für Windows 10 von McKeown et al. [22] bestätigt wurden und im Rahmen dieser Thesis auch bei Windows 11 beobachtet werden konnten.

Bei den Untersuchungen zum Löschen des Windows Thumbnail Cache durch die Datenträgerbereinigung, aber auch durch die unten behandelte Nutzung der Systemeinstellungen werden die Thumbcache-Dateien nicht sofort gelöscht, sondern vorab temporär in einen Dateiordner *ThumbCacheToDelete* verschoben. Kann dieser Ordner, z.B. durch die Nutzung forensischer Software, nachgewiesen werden, ist dies ein Hin-

weis auf die Nutzung einer der beiden Methoden zur Vernichtung von Thumbcache-Datenbanken.

Name	Size	Existent	Path	Created
Windows (182)	52,3 MB	✓	\Users\tcache\AppData\Local\Microsoft	20.06.2023 07:04:01,0 +2
Explorer (92)	19,9 MB	✓	\Users\tcache\AppData\Local\Microsoft\Windows	20.06.2023 07:04:02,7 +2
IconCacheToDelete (3)	2,0 MB	X	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	18.07.2023 10:13:46 +2
NotifyIcon (0)	0 B	X	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	20.06.2023 07:06:35 +2
NotifyIcon (0)	0 B	✓	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	20.06.2023 07:06:35,1 +2
ThumbCacheToDelete	2,0 MB	X	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	18.07.2023 10:13:46
ExplorerStartupLog.etl	512 KB	✓	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	20.06.2023 07:04:02,7 +2
ExplorerStartupLog_RunOnce.etl	24,0 KB	✓	\Users\tcache\AppData\Local\Microsoft\Windows\Explorer	20.06.2023 07:04:04,4 +2

Abbildung 8.10.: X-Ways Forensics: Ordner *ThumbCacheToDelete*

8.2.2. Nutzung der Systemeinstellungen

Eine weitere Möglichkeit, die die beiden Betriebssysteme zum Löschen des Thumbnail Cache bietet, befindet sich im Einstellungsmenü:

Einstellungen → System → Speicher → Temporäre Dateien

Hier kann der Thumbnail Cache auf Knopfdruck geleert werden.

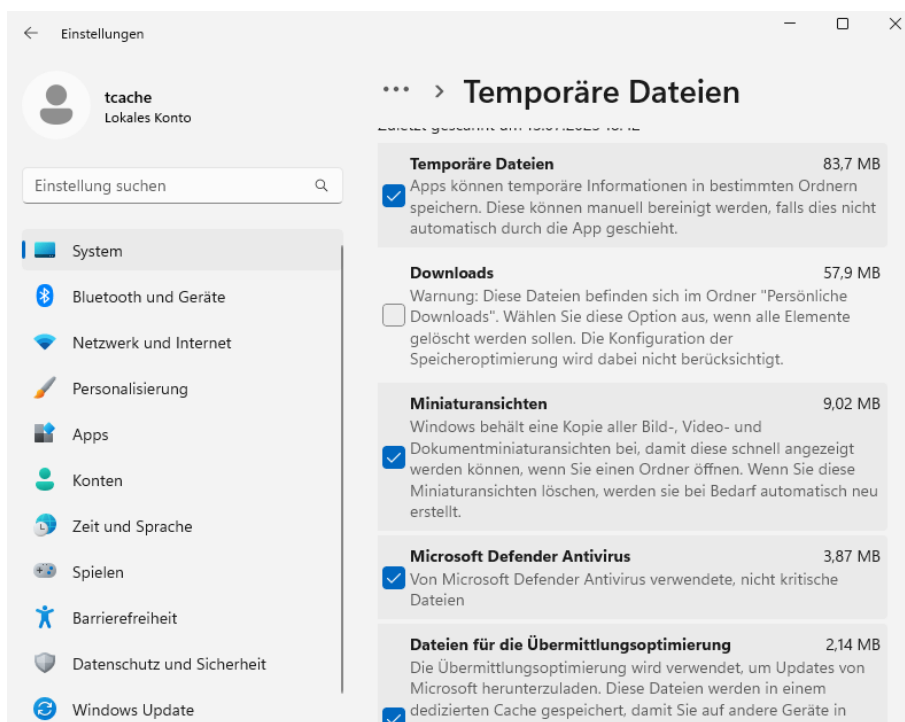


Abbildung 8.11.: Windows 11: Löschen des Thumbcache im Systemmenü

Nachweis der Datenträgerbereinigung

Für diese Art, die Dateien des Windows Thumbnail Cache zu löschen, konnten bei den durchgeführten Experimenten keine Möglichkeiten des Nachweises gefunden werden. Weder in den Dateien der Registry noch in der Ereignisanzeige oder den Prefetch Dateien wird die Nutzung dokumentiert. Einen Hinweis auf diese Art, den Thumbcache zu leeren, gibt lediglich der Nachweis des Dateiordners *ThumbCacheToDelete*, wie obenstehend beschrieben.

8.2.3. Manuelles löschen

Neben den Möglichkeiten, die das Betriebssystem hierfür bietet, ist auch das manuelle Löschen der Thumbcache-Dateien möglich. Bei der Überprüfung dieser Methode wurde festgestellt, dass manche Thumbcache-Dateien durch den Zugriff des Betriebssystems nicht gelöscht werden können. Dieser Umstand kann jedoch durch einfaches Verschieben der Dateien in einen anderen Ordner umgangen werden. Aus diesem können die Dateien anschließend gelöscht werden.

Nachweis des manuellen Löschens

Die manuelle Löschung und die darauf folgende Neuerstellung der Thumbcache-Dateien hinterlässt keine Spuren im System. Weder in der Registry noch dem Eventlog des Betriebssystems sind Spuren nachweisbar.

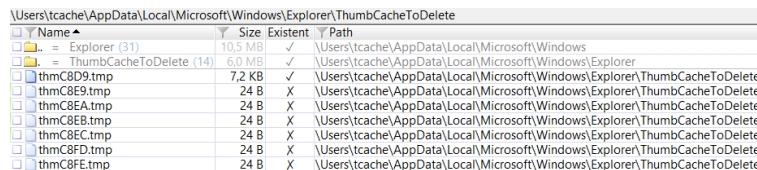
Um diese wie auch die bereits vorgestellten Methoden dennoch indirekt nachweisen zu können, kann das Verhalten des Windows Thumbnail Cache beim Erstellen der neuen Dateien (siehe 4.5) genutzt werden.

Die Zeitstempel der Erstellung und Änderung nicht genutzter Thumbcache-Datenbankdateien tragen das Datum und die Uhrzeit der Systeminstallation. Neu erstellte Datenbankdateien hingegen tragen als Änderungszeitstempel den Zeitpunkt der Löschung bzw. Neuerstellung. Sind also ungenutzte Thumbcache-Datenbankdateien vorhanden, die als Änderungszeitstempel nicht das Installationsdatum tragen, ist dies ein Zeichen dafür, dass eine Löschung des Thumbnail Cache durchgeführt wurde.

8.2.4. Wiederherstellung der gelöschten Daten

Die vorgestellten Möglichkeiten, den Windows Thumbnail Cache zu leeren bzw. zu löschen, bedienen sich dem einfachen logischen Löschen der Thumbcache-Datenbanken. Wie bereits gezeigt, werden diese (außer beim manuellen Löschen) in einen Ordner

ThumbcacheToDelete verschoben. Dabei erfahren sie auch eine Änderung des Dateinamens sowie der Dateierweiterung. Später wird der Ordner dann vom Betriebssystem gelöscht, wobei die Daten in den Speichersektoren jedoch erhalten bleiben. Die Löschung erfolgt lediglich durch Markierung der betroffenen Sektoren als *frei* im Dateisystem (hier NTFS). Dadurch gelang es mit Hilfe von X-Ways Forensics, durch die automatisierte Aufbereitung der NTFS Master File Table (MFT), die gelöschten Thumbcache-Datenbanken als ganzes wieder herzustellen.



The screenshot shows a file explorer window titled "\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete". It displays a list of files with columns for Name, Size, Existence, and Path. The files are:

Name	Size	Existence	Path
Explorer (31)	10,5 MB	✓	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer
ThumbCacheToDelete (14)	6,0 MB	✓	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8D9.tmp	7,2 KB	✓	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8E9.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8EA.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8EB.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8EC.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8FD.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete
thmC8FE.tmp	24 B	X	\\Users\\tcache\\AppData\\Local\\Microsoft\\Windows\\Explorer\\ThumbCacheToDelete

Abbildung 8.12.: X-Ways Forensics: Wiederhergestellte Thumbcache-Datenbanken

Sollten die entsprechenden MFT Einträge nicht mehr zur Verfügung stehen, ist hier auch Carving eine Lösung, um an Daten des Thumbnail Cache zu gelangen. Hier bietet sich zunächst die Möglichkeit, nach den eingebetteten Thumbnails selbst zu carven. Wie bereits behandelt, werden die Thumbnails als reguläre Bilddateien in verschiedenen Dateiformaten in die Thumbcache-Datenbanken eingebettet. Dies schließt ebenfalls die zugehörigen Header und Footer, die sog. *Magic Bytes*, mit ein, anhand derer das Carving durchgeführt werden kann.

Bei der Analyse der Thumbcache-Datenbanken von Windows 10 und Windows 11 wurde im Abschnitt 4.4.2 festgestellt, dass diese ebenfalls über einen Dateifooter verfügen, der für die Vorgängerbetriebssysteme nicht dokumentiert ist. Zusammen mit der immer vorhandenen Dateisignatur CMMM in den ersten Bytes der Datenbankdateien, eröffnet dies die Option, nicht nur nach den enthaltenen Bilddateien, sondern den Thumbcache-Datenbanken an sich zu carven. Dadurch werden auch Metadaten wie die Thumbcache-ID zugänglich.

Auch für den Fall, dass die gelöschten Thumbcache-Datenbanken inzwischen nur noch in Fragmenten vorliegen, da bereits einige ihrer Sektoren mit neuen Daten beschrieben wurden, ist es möglich, die Fragmente als Teil früherer Thumbcache-Datenbanken zu identifizieren. Morris und Chivers [34] nutzten hierfür etwa ein bayessches Netz, mit dem sie sehr hohe Identifikationsraten erzielten.

8.2.5. Bewertung

Windows 10 und auch Windows 11 verfügen über die gleichen Möglichkeiten, die Dateien ihres jeweiligen Thumbnail Cache zu löschen. Ob über die Datenträgerbereinigung, in einem Untermenü der Systemeinstellungen oder manuelles Löschen, das Resultat ist immer das Gleiche. Es werden neue Thumbcache-Dateien ohne Inhalt angelegt, während die alten durch logisches Löschen entfernt werden. Diesem Umstand ist es zu verdanken, dass Thumbcache-Dateien, die auf diese Arten entfernt wurden, bei der forensischen Auswertung im Ganzen oder zumindest teilweise wiederhergestellt werden können. Besonders hilfreich ist hier der Dateifooter der Thumbcache-Datenbanken, der das Erstellen einer Carving-Schablone ermöglicht.

Der detailreichste Nachweis der Nutzung einer Löschfunktion kann für die Datenträgerbereinigung geführt werden, indem Informationen aus Windows Prefetch und dem Background Activity Monitor zusammengeführt werden. Dadurch können sowohl der Zeitpunkt als auch das ausführende Benutzerkonto bestimmt werden.

Die beiden anderen Methoden können nur indirekt nachgewiesen werden. Entweder durch den Nachweis des ThumbcacheToDelete Ordners bei der Löschung über die Systemeinstellungen oder der Betrachtung der Zeitstempel der Thumbcache-Dateien an sich.

8.3. Löschen einzelner Einträge des Thumbnail Cache

Im vorhergehenden Abschnitt wurde gezeigt, dass das manuelle Löschen der Thumbcache-Dateien oder die Nutzung einer dafür vorgesehenen Funktion des Betriebssystems keine Möglichkeit darstellt, Daten sicher zu entfernen. Aus diesem Grund wird im Folgenden geprüft, ob es möglich ist, einzelne Einträge aus dem Thumbnail Cache gänzlich zu entfernen, sodass weder das Betriebssystem noch forensische Programme dies zur Kenntnis nehmen. Zusätzlich wird geprüft, ob durch dieses Vorgehen Daten sicher und unwiederbringlich gelöscht werden.

8.3.1. Vorgehen

In beiden untersuchten Betriebssystemen wurde zunächst eine Bilddatei so mit den Vorschaufunktionen des Datei-Explorers betrachtet, dass die Thumbcache-Datenbanken thumbcache_32.db, thumbcache_48.db und thumbcache_256.db Thumbnails der Bilddatei enthielten. Die Originaldatei wurde gelöscht, sobald die

Thumbcache-Einträge vorhanden waren. Anschließend wurden mit einem zweiten Benutzerprofil die Thumbcache-Datenbanken in einem Hexeditor bearbeitet. Die Nutzung eines zweiten Benutzerprofils war nötig, um die Sperre der Dateien zu umgehen, die eine Bearbeitung mit dem gleichen Benutzerprofil unmöglich macht. Die Verwendung dieser Methode wurde Alternativen wie der Bearbeitung der virtuellen Festplatte im offline Zustand vorgezogen, da so ein Zeitersparnis und ein flüssigeres Arbeiten an dem Experiment erzielt werden konnte.

In den Dateien wurden die jeweiligen Einträge entfernt, indem deren Hexadezimalwerte mit Nullen (0x00) überschrieben wurden. Anschließend erfolgte die Anpassung der Indexdatei thumbcache_idx.db. Darin wurde der Indexeintrag für die entsprechende Thumbcache-ID ebenfalls mit Nullen überschrieben und im Dateiheader die Anzahl der vorhandenen Indexeinträge angepasst. Abschließend erfolgte die Anmeldung mit dem Benutzerprofil des manipulierten Thumbnail Cache, zur Überprüfung des Resultats.

8.3.2. Ergebnis

Im Lauf des Experimentes wurde festgestellt, dass beide Betriebssysteme den Thumbnail Cache neu initialisieren, sobald sich in den beteiligten Dateien Unstimmigkeiten ergeben. Dabei werden alle Thumbcache-Dateien neu und ohne Inhalt erzeugt.

Werden die Einträge einer Originaldatei in den Thumbcache-Datenbanken auf die beschriebene Weise entfernt und die Indexdatei in korrekter Weise angepasst, ist dies nicht der Fall. Beide Betriebssysteme erkannten die Manipulation nicht.

Die Überprüfung der Thumbcache-Datenbanken mit Thumbcache Viewer und die Aufbereitung mit X-Ways Forensics ergaben keine Auffälligkeiten. Die noch vorhandenen Einträge wurden korrekt angezeigt. Das Carving nach den Thumbnails in X-Ways Forensics förderte die gelöschten Thumbcache-Einträge nicht zutage. Die Thumbcache-Einträge wurden also sicher und unwiederbringlich gelöscht.

Durch die Nutzung eines zweiten Benutzerkontos und der direkten Bearbeitung der Thumbcache-Dateien wurden Veränderungen der Zeitstempel und der Zugriffsrechte ausgelöst, die forensisch nachweisbar sind. Spuren, die bei der Bearbeitung der Rohdaten der Festplatte im offline Modus nicht vorhanden wären.

8.3.3. Bewertung

Das durchgeführte Experiment zeigt, dass das Entfernen von einzelnen Einträgen aus dem Thumbcache möglich ist, ohne dass die Betriebssysteme den Umstand bemerken. Voraussetzung hierfür ist, dass in den Verwaltungsdaten der Indexdatei keine Unstimmigkeiten hinterlassen werden. Dies verhindert auch, dass die Löschung bei der forensischen Auswertung entdeckt wird. Durch die Nullung der Bytes der Thumbcache-Einträge sind zudem die so entfernten Thumbnails auch durch Carving nicht wiederherstellbar. Ergebnis ist ein mit plausiblen Daten gefüllter Thumbnail Cache, aus dem unerwünschte Inhalte entfernt wurden.

8.4. Einfügen von Inhalten in Thumbcache-Datenbanken

Nachdem in den letzten Abschnitten die Löschung des Thumbnail Cache thematisiert wurde, beschäftigen sich die folgenden Ausführungen mit der Möglichkeit, Inhalte gezielt in den Thumbnail Cache einzufügen oder zu manipulieren. Das Ziel ist es, auszuloten, ob dabei Daten erzeugt werden können, die bei der forensischen Auswertung und dem Betriebssystem selbst als legitim erscheinen. Zur Anwendung kommen hier die bei der durchgeführten Analyse der aktuellen Versionen des Windows Thumbnail Cache (siehe Abschnitt 4) gewonnenen Erkenntnisse.

8.4.1. Erstellen neuer Einträge

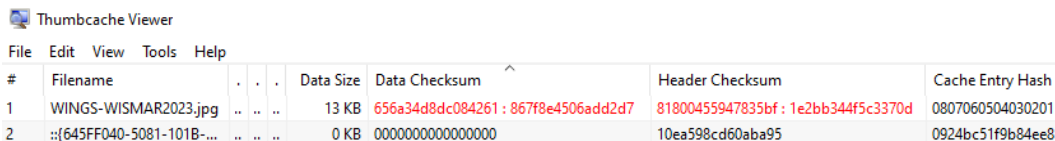
Zunächst wird untersucht, ob Einträge in den Thumbcache-Datenbanken eingefügt werden können, zu denen kein tatsächliches Originalbild vorliegt, so wie es etwa auch bei einem inzwischen gelöschten Original der Fall ist. Als Zieldatenbank des Experiments dient dabei die Datei thumbcache_256.db.

Vorgehen

Zur Manipulation der Zieldatenbank wurde auf dem Host-System vorab eine Bilddatei mit einer Höhe von 255 und einer Breite von 195 Pixeln erzeugt. Hierfür wurde das Logo der Fakultät für Ingenieurwissenschaften der Hochschule Wismar verwendet und im Format JPG gespeichert. Die Abmessungen berücksichtigen die maximal zulässigen Dimensionen der zu manipulierenden Thumbcache-Datenbank. Abschließend erfolgte das Kopieren der Dateien auf die virtuellen Maschinen.

Für den Versuch wurde die Manipulation des Thumbnail Cache mit einem Hexeditor an der Datenbankdatei thumbcache_256.db und der Indexdatei thumbcache_idx.db durchgeführt. Hierfür wurde aus den bereits dargelegten Gründen ein zweites Benutzerkonto verwendet.

In der Datenbankdatei wurde als Offset des einzufügenden Eintrags die Position des aktuellen Dateifooters gewählt, der selbst einstweilen zwischengespeichert wurde. Hier wurde ein Eintragsheader im Vista Format erstellt, wobei eine fiktive Thumbcache-ID und ein sprechender Eintragsname Anwendung fanden. An den Eintragsheader wurden die Hexadezimalwerte der verwendeten Bilddatei angefügt und im Header die Werte für die Größe der Daten, die Gesamtgröße des Eintrags und des Paddings angepasst. Die fehlenden korrekten Prüfsummen des Headers und der Daten des Eintrags wurden Thumbcache Viewer entnommen. In der Software können die Prüfsummen auf Korrektheit überprüft werden, wobei das Programm bei Unstimmigkeiten die tatsächlich korrekten Prüfsummen ausgibt. Da die Prüfsumme der Daten in die Berechnung der Prüfsumme des Eintragsheaders einfließt, wurden auf diese Weise zunächst die Datenprüfsumme angepasst und anschließend die korrekte Prüfsumme des Eintragsheaders übernommen. Das Ergebnis der Manipulation ist in Bild 8.14 zu sehen. Auf den korrekten Eintragsheader folgen nach dem Padding die ersten Bytes (Magic Bytes 0xFFD8FFE0) der eingefügten Bilddatei.



#	Filename	Data Size	Data Checksum	Header Checksum	Cache Entry Hash
1	WINGS-WISMAR2023.jpg	13 KB	656a34d8dc084261 : 867f8e4506add2d7	81800455947835bf : 1e2bb344f5c3370d	0807060504030201
2	::[645FF040-5081-101B-...	0 KB	0000000000000000	10ea598cd60aba95	0924bc51f9b84ee8

Abbildung 8.13.: Thumbcache Viewer: Eintrag mit inkorrekten Prüfsummen

Nachdem der Eintrag erstellt war, wurde der zwischengespeicherte Dateifooter unter die Hexadezimalwerte der Bilddatei eingefügt und die Anzahl der Bytes bis zum Dateiende angepasst. Im Dateihheader erfolgte abschließend noch die Aktualisierung der Startposition des Dateifooters.

In der Indexdatei thumbcache_idx.db wurde direkt im Anschluss an einen beliebigen vorhandenen Indexeintrag ein neuer Eintrag erstellt. Dieser wurde mit der fiktiven Thumbcache-ID und dem Offset des Eintrags in der thumbcache_256.db versehen. Zuletzt erfolgte die Anpassung des Dateihheaders. Hier wurde die Anzahl der insgesamt vorhandenen Einträge der Indexdatei und die Anzahl der Einträge, die die Datei thumbcache_256.db betreffen, um je 1 inkrementiert.

Offset(d)	00	01	02	03	04	05	06	07	Decoded text
00014824	37	00	00	00	00	00	00	00	7.....
00014832	43	4D	4D	4D	88	34	00	00	CMMM*4..
00014840	01	02	03	04	05	06	07	08
00014848	20	00	00	00	01	00	00	00
00014856	2F	34	00	00	00	01	00	00	/4.....
00014864	00	01	00	00	00	00	00	00
00014872	D7	D2	AD	06	45	8E	7F	86	*Ö..EŽ.†
00014880	8F	00	20	51	80	CD	2A	EC	.. Q€í*ì
00014888	57	00	49	00	4E	00	47	00	W.I.N.G.
00014896	53	00	2D	00	57	00	49	00	S.-.W.I.
00014904	53	00	4D	00	41	00	52	00	S.M.A.R.
00014912	32	00	30	00	32	00	33	00	2.0.2.3.
00014920	00	FF	D8	FF	E0	00	10	4A	.ÿØÿà..J

Abbildung 8.14.: thumbcache_256.db: Manuell erstellter Eintragsheader

Offset(h)	00	01	02	03	04	05	06	07	Decoded text
00005770	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
00005778	01	02	03	04	05	06	07	08
00005780	00	01	10	00	00	00	00	00
00005788	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
00005790	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
00005798	F0	39	00	00	FF	FF	FF	FF	89..ÿÿÿÿ
000057A0	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
000057A8	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
000057B0	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
000057B8	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿ
000057C0	41	4A	6E	71	17	BE	46	39	AJnq.*F9

Abbildung 8.15.: thumbcache_idx.db: Manuell erstellter Indexeintrag

Um die Persistenz des erzeugten Eintrags zu testen, erfolgten nach Abschluss der Manipulation mehrere Neustarts der Betriebssysteme. Zusätzlich wurden durch Betrachten von Vorschaubildern neue Einträge in der thumbcache_256.db erzeugt, bis die Datei aus Platzmangel vom Betriebssystem vergrößert wurde.

Ergebnis

Trotz mehrerer Neustarts der Betriebssysteme und der intensiven Nutzung der manipulierten Thumbcache-Datenbankdatei blieb der eingefügte Eintrag in der Datei vorhanden. Beim Öffnen der Datei mit Thumbcache Viewer und einer forensischen Aufbereitung durch X-Ways Forensics wurde der künstliche Eintrag wie jeder legitime Thumbcache-Eintrag behandelt. Lediglich der sprechende Eintragsname wurde als ungewöhnlicher Dateiname interpretiert.

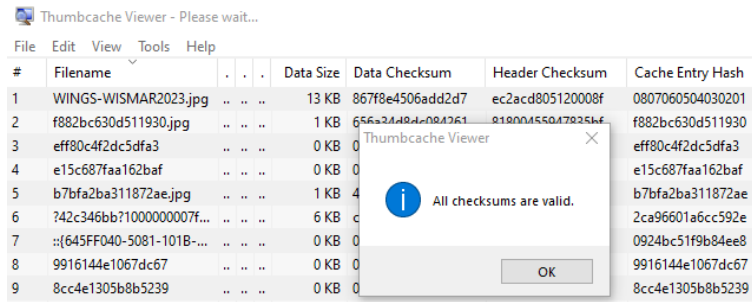


Abbildung 8.16.: Thumbcache Viewer: Manuell erstellter Thumbcache-Eintrag

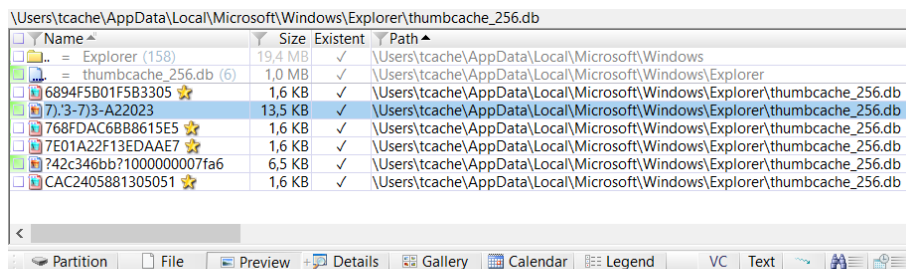


Abbildung 8.17.: X-Ways Forensics: Manuell erstellter Thumbcache-Eintrag

Nachweis des eingefügten Thumbnails

Der Nachweis eines manuell eingefügten Thumbnails in einer Thumbcache-Datenbank gestaltet sich schwierig. Werden plausibel scheinende Werte für Thumbcache-ID und Eintragsnamen (üblicherweise die UTF-16 Version der Thumbcache-ID) verwendet, lässt sich weder aus den Daten des Eintragsheaders noch aus dem zugehörigen Eintrag in der Indexdatei ableiten, dass ein Thumbnail nicht legitim ist. Der durchgeführte Versuch zeigt, dass weder die verwendeten forensische Programme noch die beiden Betriebssysteme die Manipulation erkennen.

Auffälligkeiten ergeben sich unter Umständen jedoch aus dem Thumbnail selbst. Bei dem durchgeführten Versuch wurde das Thumbnail mit der Software paint.net erstellt und erhielt dadurch entsprechende Exchangeable Image File Format (EXIF) Daten.

Solche dateiinternen Metadaten sind in regulären Thumbnails des Windows Thumbnail Cache nicht vorhanden. Abbildung 8.18 zeigt den Unterschied zwischen der manuell eingefügten Datei (links) und dem vom System erstellten Thumbnail der Originaldatei beider Thumbnails (rechts).

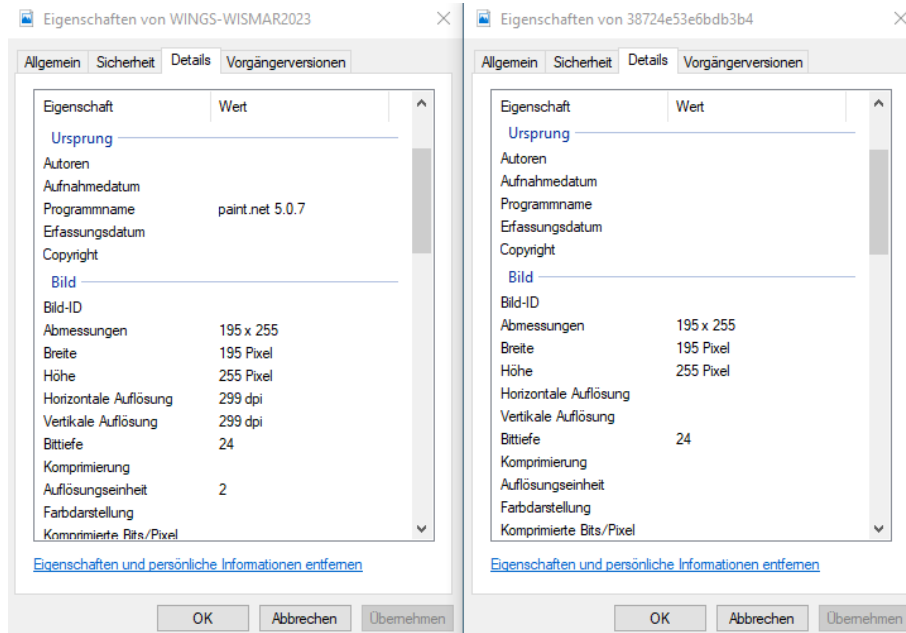


Abbildung 8.18.: Dateieigenschaften des manuell erstellten und des legitimen Thumbcache-Eintrags

8.4.2. Manipulation vorhandener Einträge

Das abschließende Experiment dient dem Zweck, die Grenzen der Manipulierbarkeit von Thumbcache-Einträgen auszuloten. Hierfür soll der vorhandene Thumbcache-Eintrag einer Originaldatei so manipuliert werden, dass im Datei-Explorer nicht das korrekte Thumbnail angezeigt wird. Als Ziel der Manipulation dient die Anzeige im minimalen Vorschaufenster bei maximiertem Datei-Explorer, also die Thumbcache-Datenbank thumbcache_1280.db. Die Manipulationen erfolgten wegen der Dateisperre wieder mithilfe eines zweiten Benutzerkontos.

Vorgehen

Zunächst wurde mit der Software paint.net eine Testdatei passend zu den vorausgesetzten Dimensionen der gewählten Thumbcache-Datenbank in Breite und Höhe erstellt. Anschließend wurde eine weitere Bilddatei im Datei-Explorer mit den entsprechenden Vorschauoptionen betrachtet, um Einträge in den Thumbcache-Datenbanken thumbcache_32.db, thumbcache_48.db und thumbcache_256.db zu erzeugen. Die Datenbank-

datei thumbcache_1280.db verblieb dabei ohne Einträge und enthielt zu diesem Zeitpunkt lediglich den ihr zugehörigen Dateiheader.

Sobald diese Ausgangslage bereitet war, wurde in einem zweiten Benutzerkonto mit der Manipulation der thumbcache_1280.db begonnen. Auf einen manuell erstellten Eintragsheader folgend, wurden die Hexadezimalwerte der verkleinerten Bilddatei angefügt. Nachdem der Eintragsheader mit korrekten Werten versehen war, wurde darunter ein Dateifooter erstellt und dessen Offsetangaben zusammen mit denen des Dateiheaders angepasst. Abschließend erfolgte der Eintrag im dafür vorgesehenen Offsetsfeld des Eintrags der Originaldatei in der thumbcache_idx.db.

Ergebnis

Durch die vorgenommene Manipulation präsentiert der Datei-Explorer bei beiden Betriebssystemen im Vorschauenfenster nicht eine verkleinerte Version der Originaldatei, sondern die manuell eingefügte Datei.

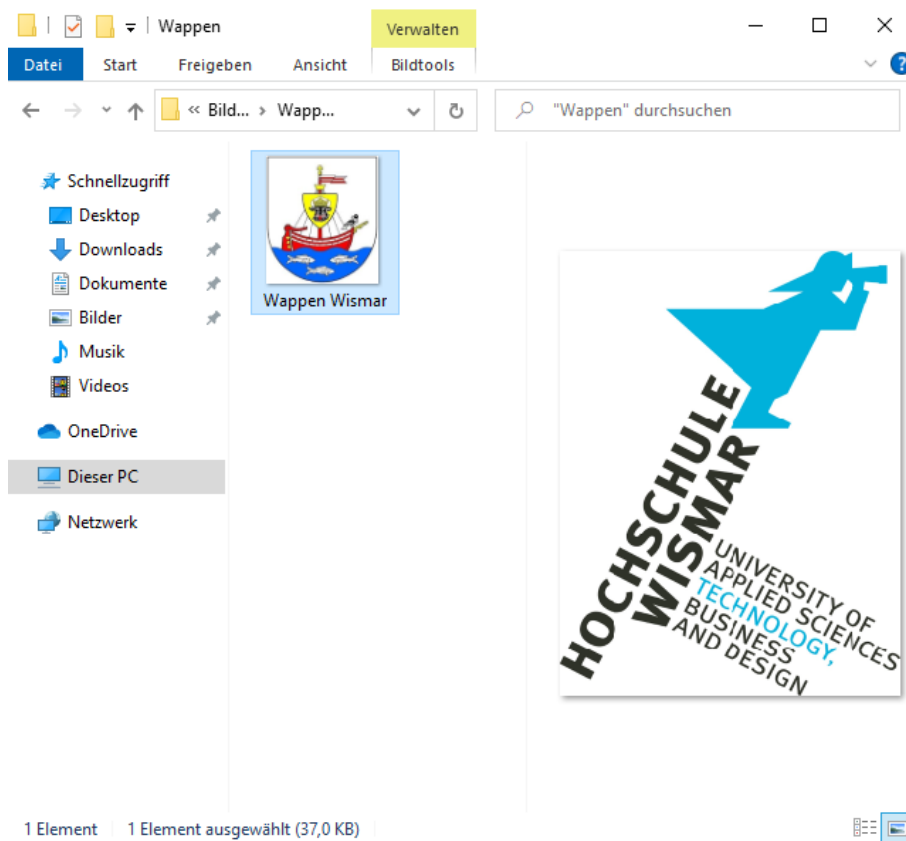


Abbildung 8.19.: Windows 10: Manipulierte Vorschauensteransicht

Dieser Zustand blieb trotz mehrerer Neustarts und der Aufnahme weiterer Thumbnails in der manipulierten Datei konstant. Erst als die Betrachtung nicht mehr im maximierten Fenster erfolgte, wurde wieder das korrekte Bild angezeigt. Das inkorrekte

Thumbnail verblieb dabei allerdings in der manipulierten Datenbank und wurde nicht aktualisiert.

Nachweis der Manipulation

Der Nachweis der durchgeführten Manipulation gelingt, indem für jede Thumbcache-ID die Inhalte aller Thumbnails der entsprechenden Thumbcache-Datenbanken miteinander verglichen werden. So können Abweichungen leicht festgestellt werden. Mit steigender Anzahl von Thumbcache-Einträgen nimmt die dafür benötigte Zeit jedoch ebenfalls merklich zu.

8.4.3. Bewertung

Die durchgeführten Versuche zum Einfügen neuer oder manipulieren bestehender Thumbcache-Einträge zeigen, dass es bei korrekter Ausführung so möglich ist, dass weder das Betriebssystem noch forensische Programme dies bemerken. Wird ein neuer Eintrag als Ganzes manuell hinzugefügt, scheint dieser bei der forensischen Auswertung wie ein Eintrag eines inzwischen gelöschten Originalbildes. Dies ist umso mehr der Fall, wenn die Manipulationen nicht an den Dateien selbst, sondern auf Dateisystemebene stattfinden und somit die Zeitstempel der Thumbcache-Dateien nicht verändert werden.

Der Austausch eines Thumbnails eines vorhandenen Eintrags durch ein gänzlich anderes dürfte bei der tatsächlichen forensischen Arbeit kaum jemals vorkommen, zeigt jedoch, wie weit Manipulationen des Thumbcache-Inhalts gehen können.

9. Empfohlenes Vorgehen bei der forensischen Auswertung

Basierend auf den Erkenntnissen der Thesis, wird in diesem Kapitel der empfohlene Ablauf der forensischen Auswertung des Windows Thumbnail Cache dargelegt. Die Ausführungen sind dabei als Handlungsempfehlung für die Betriebssysteme Windows 10 und Windows 11 zu verstehen. Da jedes zur Untersuchung vorliegende Asservat einzigartig hinsichtlich der darauf vorhandenen Spuren ist, kann und ggf. muss die Vorgehensweise der vorliegenden Situation angepasst werden. Eine veränderte Reihenfolge der Abarbeitung oder Überspringen einzelner Schritte ist hier möglich.

1. Datensicherung

Handelt es sich bei der forensischen Auswertung um eine Post-mortem-Analyse, liegt das Asservat also in ausgeschaltetem Zustand vor, ist ein Datenimage der vorhandenen Datenspeicher anzufertigen. Um Veränderungen am Asservat zu vermeiden, ist ein geeigneter Schreibblocker einzusetzen. Dadurch wird die geforderte Beweissicherheit berücksichtigt, indem ausschließlich lesende Zugriffe erfolgen [4].

Für den Fall, dass sich das Asservat noch in Betrieb befindet und ein Herunterfahren nicht in Betracht kommt oder dadurch ein Datenverlust zu befürchten ist, kann auf eine Live-Sicherung des Systems zurückgegriffen werden. Ist lediglich der Thumbnail Cache zu sichern, können die relevanten Dateien im Ausnahmefall auch manuell kopiert werden, was eine vorherige Dokumentation der vorhandenen Zeitstempel jedoch zwingend voraussetzt.

In beiden Szenarien ist die Absicherung und Verifizierung der erstellten Datensicherungen durch kryptographische Prüfsummen angeraten.

2. Prüfung der Thumbcache-Dateien

Nach der Datensicherung sollten zunächst die Zeitstempel und Speichergrößen der Dateien des Thumbnail Cache näher betrachtet werden. Ein besonderes Augenmerk ist hier auf Dateien zu legen, die keine Daten oder nur den Dateiheader beinhalten.

Stimmen deren Zeitstempel für Erstellung und Änderung nicht überein, ist dies ein Indiz für eine vorgenommene Löschung des Thumbnail Cache.

Die Dateigrößen der Thumbcache-Datenbanken können einen Hinweis auf eine durchgeführte Manipulation geben. Je nach Verwendung bestimmter Vorschauoptionen wachsen die Datenbanken ungleichmäßig an, wie in Abbildung 4.7 zu sehen ist. Verfügen die Datenbanken dagegen über (nahezu) die gleiche Dateigröße, z.B. 1024 KB, kann das ebenfalls ein Hinweis auf eine kürzlich durchgeführte Löschung des Thumbnail Cache sein.

Je nach verwendeter Technik lässt sich die Löschung durch verschiedene Methoden, die in Abschnitt 8.2 dargelegt sind, nachvollziehen.

3. Prüfung auf Deaktivierung des Thumbcache Cache

Bei Deaktivierung des Thumbnail Cache wird zwar dessen Weiternutzung unterbunden, vorhandene Inhalte aber nicht entfernt. So kann durch die gefüllten Thumbcache-Datenbanken das Bild entstehen, der Thumbnail Cache sei aktiv, während er tatsächlich nicht weiter genutzt wird. Aus diesem Grund ist es immer sinnvoll, auf eine mögliche Deaktivierung des Thumbnail Cache zu prüfen. Abschnitt 8.1.6 behandelt die diesbezüglichen Möglichkeiten.

4. Suche nach gelöschten Thumbcache-Datenbanken

Sind Hinweise auf die Löschung von Thumbcache-Datenbanken vorhanden, sollte versucht werden, deren Inhalte ganz oder teilweise wiederherzustellen. Abschnitt 8.2.4 bietet hierfür eine Hilfestellung.

5. Identifizierung relevanter Inhalte

Dieser Schritt ist der erste, der sich mit den Inhalten der Thumbcache-Datenbanken beschäftigt. Je nach Ziel der Auswertung und zugrunde liegendem Delikt sind nur bestimmte Thumbnails innerhalb der Thumbcache-Datenbanken relevant. Um diese zu identifizieren, werden die Thumbcache-Datenbanken mit der bevorzugten forensischen Software geöffnet und gesichtet.

6. Prüfung auf Manipulationen

Ergeben sich aus den Aussagen der beteiligten Personen oder bei der Auswertung des Asservats Hinweise auf eine mögliche Manipulation des Inhalts des Thumbnail Cache, sind Schritte zu unternehmen, dies zu untersuchen. Hier ist vor auszuschicken,

dass die Chance, korrekt durchgeführte Manipulationen forensisch nachzuweisen, äußerst gering ist. Die nachfolgenden Techniken sind deshalb nur bei mangelhaft durchgeführten Manipulationen erfolgversprechend.

Besteht der Verdacht, dass Inhalte in den Thumbnail Cache eingefügt wurden, sind zunächst die Prüfsummen für den Eintragsheader und das Thumbnail selbst zu betrachten. Die Software Thumbcache Viewer kann dabei genutzt werden, um Unstimmigkeiten aufzudecken.

Die Thumbcache-ID der einzelnen Einträge kann ebenfalls zum Nachweis einer Inhaltsmanipulation verwendet werden. Hierfür kann jede Thumbcache-Datenbank nach der ID durchsucht und die zugehörigen Thumbnails gegenübergestellt werden. Haben nicht alle Thumbnails einer ID den gleichen Inhalt, ist von einer Manipulation auszugehen.

Eine weitere Möglichkeit, die Manipulation nachzuweisen, ist die Betrachtung der eingebetteten Thumbnails selbst. Legitime Thumbnails verfügen über keine EXIF-Daten, die Informationen zum verwendeten Kameramodell oder einer Bildbearbeitungssoftware liefern. Thumbnails mit solchen Einträgen sind ein starkes Indiz für eine Manipulation.

Wurde die Manipulation des Inhalts auf Dateiebene durchgeführt, wie es etwa mit einem zweiten Benutzerkonto möglich ist, wird der Zeitstempel der letzten Änderung dadurch beeinflusst. Da aber auch das natürliche Anwachsen oder die Löschung der Thumbcache-Datenbanken diese Wirkung haben, kann durch den Zeitstempel alleine nur im Ausnahmefall eine Manipulation belegt werden.

7. Identifizierung der genutzten Vorschauoptionen

Je nach Delikt und Tatumständen kann der Nachweis, dass der Beschuldigte Kenntnis vom Inhalt einer Datei haben konnte, von erheblicher Bedeutung für den Fall sein. Gelingt dies, wird das Abstreiten dieser Kenntnis erschwert. Deshalb ist es sinnvoll, für relevante Thumbcache-Einträge festzustellen, durch welche Umstände sie erstellt wurden.

Durch das in Abschnitt 6.3.3 festgestellte automatisierte Erstellen von Thumbcache-Einträgen ist zunächst festzuhalten, dass bestimmte Thumbcache-Datenbanken hierfür nicht geeignet sind. Nur Datenbanken, in denen nachweislich keine automatisierten Einträge erfolgen, können sicher zu diesem Zweck herangezogen werden.

Kann der Nachweis geführt werden, dass eine relevante Bilddatei im Vorschaufenster angezeigt wurde, stellt dies das stärkste Indiz über die Kenntnis des Bildinhalts dar, da hierfür die entsprechende Originaldatei im Datei-Explorer markiert werden muss.

Weitere Informationen zu den Spuren der Vorschauoptionen sind Kapitel 6 zu entnehmen.

Dokumentation der Ergebnisse

Die bei der Auswertung gewonnenen Erkenntnisse sind abschließend für die weitere Verwendung zu dokumentieren.

10. Zusammenfassung und Ausblick

Ziel dieser Bachelor Thesis war es, diverse Aspekte des Windows Thumbnail Cache zu untersuchen und Unterschiede zwischen den Betriebssystemen Windows 10 und Windows 11 aufzuzeigen. Im Laufe der Auswertung der Ergebnisse kristallisierte sich heraus, dass beide Betriebssysteme bei keinem der durchgeführten Experimente voneinander abweichendes Verhalten zeigen. Es konnten keine Unterschiede bei den Abläufen und den Reaktionen des Windows Thumbnail Cache festgestellt werden. Dieses identische Verhalten ist leicht nachzuvollziehen, da beide Betriebssysteme nahezu die gleiche Version der Systemdatei *thumbcache.dll* teilen (siehe Anhang A.1).

Nichtsdestoweniger konnten im Rahmen der durchgeführten Arbeit und der damit verbundenen Untersuchungen diverse Erkenntnisse gewonnen und sämtliche Forschungsfragen beantwortet werden. Hier ist zunächst die Analyse des inneren Aufbaus der Thumbcache-Datenbanken und der zugehörigen Indexdatei zu nennen. Sinn und Zweck der darin vorhandenen Datenfelder konnte durch zahlreiche Versuche zum größten Teil ermittelt werden. Hierbei wurden weitreichende Unterschiede zu dem bisherigen Stand der Forschung festgestellt, der sich auf die Betriebssysteme Windows Vista und Windows 7 beschränkt. Die hierbei gewonnenen Erkenntnisse erwiesen sich bei der späteren Manipulation der Thumbcache-Datenbanken als unverzichtbar.

Die Untersuchung der Mechanismen zeigte, dass die Funktion des Windows Thumbnail Cache unabhängig vom Dateiformat, dem Inhalt und der Größe einer Bilddatei ist. Je nach gewählter Vorschauoption im Datei-Explorer werden Einträge in den dafür vorgesehenen Datenbanken erzeugt. So kann aufgrund der Verteilung der Thumbnails in den einzelnen Datenbanken darauf geschlossen werden, welche Vorschauoptionen bemüht wurden, um eine bestimmte Originaldatei zu betrachten.

Ebenfalls keinen Einfluss auf das Verhalten des Thumbnail Cache übt der Speicherort einer Bilddatei aus, solange diese entweder auf einem lokalen oder einem USB-Datenträger gespeichert wird. Auf einem Netzwerkspeicher abgelegte Dateien führen dagegen zu teilweise deutlichen Abweichungen, was die Nutzung der jeweiligen Thumbcache-Datenbanken angeht. Ähnliches konnte auch für die Software des Cloud

Anbieters Google festgestellt werden. Durch dessen ausnahmslos quadratischer Darstellung von Vorschaubildern ist hier die sonst übliche Sonderbehandlung von schmal rechteckigen Bilddateien nicht vorhanden, was ein Alleinstellungsmerkmal unter den betrachteten Cloudanbietern darstellt.

Metadaten zu vorhandenen Thumbnails lassen sich nur den Datenbanken von Windows Search entnehmen. Sie beschränken sich jedoch im ersten Schritt auf Informationen zu nicht gelöschten Originaldateien, da zugehörige Einträge beim Löschen von Dateien in Echtzeit aus der Suchdatenbank entfernt werden. Die Wiederherstellung dieser gelöschten Einträge ist nur bei Windows 10 und seiner ESE Datenbank möglich. Die bei Windows 11 eingesetzte SQLite Datenbank verwehrt die Suche nach gelöschten Einträgen durch die vermutlich aktivierte Option *secure_delete*.

Beide untersuchten Betriebssysteme bieten die gleichen Möglichkeiten, die Funktion des Thumbnail Cache zu deaktivieren. Egal, ob hierfür die Explorer Optionen, die Systemeinstellungen oder Gruppenrichtlinien genutzt werden, das Ergebnis ist die zuverlässige Unterbindung der Nutzung des Thumbnail Cache. Es werden keine Einträge in den Datenbanken hinterlegt, die für eine forensische Auswertung von Nutzen sein könnten. Die Anwendung dieser Techniken kann jedoch anhand der Registry Datei NTUSER.DAT im betroffenen Benutzerverzeichnis nachgewiesen werden.

Durch Versuche wurde gezeigt, dass der Thumbnail Cache nicht vor Manipulationen der Datenbankinhalte abgesichert ist. So ist es möglich, bestehende Einträge so zu entfernen, dass auch durch Carving nicht wiederherstellbar sind. Fiktive Einträge können erstellt und vorhandene Einträge so abgeändert werden, dass falsche Vorschaubilder das Resultat sind. Werden hierbei die Verwaltungsdaten in der Indexdatei und den Eintragsheadern in den Thumbcache-Datenbanken korrekt gesetzt, ist es weder den Betriebssystemen noch den verwendeten forensischen Programmen möglich, die Veränderungen festzustellen.

Allgemein waren die durchgeführten Experimente sehr umfangreich und zeitaufwendig, konnten aber dennoch nur Teilbereiche des Windows Thumbnail Cache beleuchten. Dies ist vor allem bei der Betrachtung der Mechanismen der Fall. Hier wurden die Wirkungsweisen der Vorschauoptionen aufgrund der endlos scheinenden Möglichkeiten, die Fenstergröße zu wählen, ausschließlich bei maximiertem Fenster des Datei-Explorers untersucht. Beliebige Fenstergrößen oder auch abweichende Bildschirmauflösungen könnten zu anderen Ergebnissen führen, die bei Bedarf im

Rahmen einer forensischen Auswertung gezielt untersucht werden müssen.

Als Nebenprodukt der durchgeführten Experimente wurde erkannt, dass Thumbcache-Einträge gänzlich ohne vorherige Aktion des Benutzers erstellt werden. Eine Feststellung, die Parsonage 2012 auch bei seiner Analyse von Windows 7 erwähnt [40]. Er schließt daraus, dass fortan nicht mehr automatisch davon ausgegangen werden kann, dass der Inhalt einer Originaldatei einem Benutzer durch die Vorschaufunktion des Datei-Explorers angezeigt wurde und dadurch bekannt sein muss, sobald Thumbnails der Datei im Thumbnail Cache vorhanden sind. Durch die Feststellung des gleichen Verhaltens bei Windows 10 und Windows 11 hat dies auch für deren Thumbnail Caches zu gelten.

Neben dieser automatischen Erstellung von Thumbnails sollte auch die nachgewiesene Manipulierbarkeit des Thumbnail Cache bei der forensischen Auswertung bedacht werden. Die durchgeführten Manipulationen wurden im Rahmen der Thesis manuell im Hexeditor durchgeführt, eine Automatisierung mittels einer hierfür entwickelten Software ist jedoch denkbar.

Die gezeigten Ergebnisse spiegeln insgesamt den Status quo wider. Da es sich beim Windows Thumbnail Cache um ein oft genutztes forensisches Artefakt handelt, sollte die zukünftige Entwicklung des Windows Betriebssystems daher nicht aus dem Blick gelassen werden. Alleine schon Microsofts die Ankündigung, das Windows Betriebssystem künftig gänzlich in die Cloud auszulagern [46] sollte zum Anlass genommen werden, die Forschung auf diesem Gebiet weiterzubetreiben.

Literaturverzeichnis

- [1] S. Bhosale and P. Patil. SQLite: Light Database System. *International Journal of Computer Science and Mobile Computing*, 44:882–885, April 2015.
- [2] R. Bocksch. Jeder dritte Deutsche speichert in der Cloud, Januar 2021. <https://de.statista.com/infografik/3077/nutzung-von-cloud-speichern-in-europa/> (Zugriff am: 05.05.2023).
- [3] Brewster, Rick. paint.net 5.0.7. dotPDN LLC. <https://www.getpaint.net/> (Zugriff am: 18.07.2023).
- [4] Bundesamt für Sicherheit in der Informationstechnik. Leitfaden „IT-Forensik“, März 2011. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf?__blob=publicationFile&v=2 (Zugriff am: 14.05.2023).
- [5] E. Casey. *Handbook of Computer Crime Investigation*. Elsevier, Oktober 2001.
- [6] H. Chivers and C. Hargreaves. Forensic data recovery from the Windows Search Database. *Digital Investigation*, 7(3-4):114–126, April 2011.
- [7] K. J. Conlan, I. Baggili, and F. Breitingner. Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy. *Digital Investigation*, 18:S66–S75, 8 2016.
- [8] C. De Alwis. Windows Registry Analysis 101. *Forensic Focus*, Mai 2020.
- [9] B. Dolan-Gavitt. Forensic analysis of the Windows registry in memory. *Digital Investigation*, 5:S26–S32, September 2008.
- [10] D. J. Farmer. A Windows Registry Quick Reference: For the Everyday Examiner. Januar 2007.
- [11] P. Fischer and P. Hofer. *Lexikon der Informatik*. Springer-Verlag, Dezember 2010.

- [12] gfu. Absatz von USB-Sticks auf dem Konsumentenmarkt in Deutschland von 2004 bis 2022 (in Millionen Stück), Februar 2023. <https://de.statista.com/statistik/daten/studie/151613/umfrage/absatz-von-usb-sticks-seit-2004-in-deutschland/> (Zugriff am 03.07.2023).
- [13] D. Heinson. *IT-Forensik*. Mohr Siebeck, September 2015.
- [14] D. Hurlbut. AccessData Corporation: Thumbs DB Files Forensic Issues, 2005. <https://usermanual.wiki/Pdf/WpThumbsDbFilesEnUs.1606461609/view> (Zugriff am: 28.05.2023)
Originalveröffentlichung (http://accessdata.com/media/en_us/print/papers/wp.Thumbs_DB_Files.en_us.pdf) nicht mehr verfügbar.
- [15] iXsystems. TrueNAS CORE 13.0-U5.1. <https://www.truenas.com/> (Zugriff am: 03.07.2023).
- [16] J. Kävestad. *Fundamentals of Digital Forensics: Theory, Methods, and Real-Life Applications (English Edition)*. Springer, 2 edition, Mai 2020.
- [17] J. Kim. WinSearchDBAnalyzer 1.0.0.6. <https://moaistory.blogspot.com/2018/10/winsearchdbanalyzer.html> (Zugriff am: 08.07.2023).
- [18] Kutcher, Eric. Thumbcache Viewer 1.0.3.7. <https://thumbcacheviewer.github.io/> (Zugriff am: 02.06.2023).
- [19] D. Labudde and M. Spranger. *Forensik in der digitalen Welt*. Springer-Verlag, März 2017.
- [20] T. Larson. Windows 7 Thumbnail Cache. Juni 2011. <https://www.slideshare.net/ctin/windows-7-forensics-thumbnaildtlr4/> (Zugriff am: 25.07.2023).
- [21] Martin Kleusberg u. a. DB Browser for SQLite 3.12.2. <https://sqlitebrowser.org/> (Zugriff am: 06.07.2023).
- [22] S. McKeown, G. Russell, and P. Leimich. Fast Forensic Triage Using Centralised Thumbnail Caches on Windows Operating Systems. *The Journal of Digital Forensics, Security and Law*, Januar 2019. <https://commons.erau.edu/cgi/viewcontent.cgi?article=1591&context=jdfsl> (Zugriff am: 01.03.2023).
- [23] J. McQuaid. Forensic analysis of prefetch files in Windows. *Magnet Forensics*, August 2014. <https://www.magnetforensics.com/>

- blog/forensic-analysis-of-prefetch-files-in-windows/ (Zugriff am: 15.07.2023).
- [24] J. Metz. Windows Search forensics: Analyzing the Windows (Desktop) Search Extensible Storage Engine database, 2010. <https://raw.githubusercontent.com/libyal/documentation/main/Forensic%20analysis%20of%20the%20Windows%20Search%20database.pdf> (Zugriff am: 16.05.2023).
- [25] J. Metz. Windows Explorer Thumbnail Cache database file format specification, 2021. <https://github.com/libyal/libwtcdb/blob/main/documentation/Windows%20Explorer%20Thumbnail%20Cache%20database%20format.asciidoc> (Zugriff am: 31.05.2023).
- [26] mh-nexus. HxD 2.5.0.0. <https://mh-nexus.de/de/> (Zugriff am: 02.06.2023).
- [27] Microsoft Corporation. Disk cleanup in Windows. <https://support.microsoft.com/en-us/windows/disk-cleanup-in-windows-8a96ff42-5751-39ad-23d6-434b4d5b9a68> (Zugriff am: 15.07.2023).
- [28] Microsoft Corporation. Search for anything, anywhere. https://support.microsoft.com/en-us/windows/search-for-anything-anywhere-b14cc5bf-c92a-1e73-ea18-2845891e6cc8#ID0EDF=Windows_10 (Zugriff am: 18.05.2023).
- [29] Microsoft Corporation. Windows-Explorer hat einen neuen Namen. <https://support.microsoft.com/de-de/windows/windows-explorer-hat-einen-neuen-namen-c95f0e92-b1aa-76da-b994-36a7c7c413d7> (Zugriff am: 11.06.2023).
- [30] Microsoft Corporation. Esentutl: Management and Tools / Command-Line Reference, August 2016. [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh875546\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh875546(v=ws.11)) (Zugriff am: 17.05.2023).
- [31] Microsoft Press. *Microsoft Computer Dictionary*. 2002.
- [32] S. Morris and H. Chivers. An analysis of the structure and behaviour of the Windows 7 operating system thumbnail cache. Proceedings from 1st International Conference on Cybercrime, Security and Digital Forensics, 27-28 June 2011, Juni 2011.

- [33] S. Morris and H. Chivers. Forming a Relationship between Artefacts identified in thumbnail caches and the remaining data on a storage device, September 2011.
- [34] S. Morris and H. Chivers. Thumbnail Cache File Fragment Identification using a Bayesian Network. Juni 2013.
- [35] S. L. A. Morris. *An Investigation into the identification, reconstruction, and evidential value of thumbnail cache file fragments in unallocated space*. PhD thesis, Cranfield University, 2013.
- [36] NirSoft. ESEDatabaseView v1.73. https://www.nirsoft.net/utils/ese_database_view.html (Zugriff am: 05.07.2023).
- [37] NirSoft. RegistryChangesView v1.29. https://www.nirsoft.net/utils/registry_changes_view.html (Zugriff am: 13.07.2023).
- [38] NirSoft. WinPrefetchView v1.37. https://www.nirsoft.net/utils/win_prefetch_view.html (Zugriff am: 15.07.2023).
- [39] Oracle. VirtualBox 7.0.8 r156879. <https://www.virtualbox.org/> (Zugriff am: 02.06.2023).
- [40] H. Parsonage. Under My Thumbs - Revisiting Windows thumbnail databases and some new revelations about the forensic implications. Januar 2012. <http://computerforensics.parsonage.co.uk/downloads/UnderMyThumbs.pdf> (Zugriff am: 25.07.2023).
- [41] D. M. Purcell and S.-D. Lang. Forensic Artifacts of Microsoft Windows Vista System. In C. C. Yang, H. Chen, M. Chau, K. Chang, S.-D. Lang, P. S. Chen, R. Hsieh, D. Zeng, F.-Y. Wang, K. Carley, W. Mao, and J. Zhan, editors, *Intelligence and Security Informatics*, pages 304–319, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [42] Quick, Darren and Tassone, Christopher and Choo Kim-Kwang Raymond. Forensic Analysis of Windows Thumbcache Files. *Social Science Research Network*, April 2014. https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID2429795_code2138273.pdf?abstractid=2429795 (Zugriff am: 01.03.2023).
- [43] M. Rogers. Anti-Forensics, präsentiert bei Lockheed Martin, September 2005. https://www.researchgate.net/publication/268290676_Anti-Forensics_Anti-Forensics (Zugriff am: 15.05.2023).
- [44] P. Sanderson. *SQLite Forensics*. Mai 2018.

- [45] E. Schicker. *Datenbanken und SQL*. Springer-Verlag, Januar 2017.
- [46] F. Schräier. Microsoft will Windows 11 komplett in die Cloud verlagern. Juni 2023. <https://www.heise.de/news/Microsoft-will-Windows-11-komplett-in-die-Cloud-verlagern-9200869.html> (Zugriff am: 25.07.2023).
- [47] N. Sebastian. Usage & Trends of Personal Cloud Storage: GoodFirms Research. <https://www.goodfirms.co/resources/personal-cloud-storage-trends> (Zugriff am: 01.07.2023).
- [48] B. Stewart. Forensic Implications of Windows Vista, 2007. <https://whereismydata.files.wordpress.com/2009/09/forensic-implications-of-windows-vista.pdf> (Zugriff am: 28.05.2023).
- [49] L. Taylor. ESE Deep Dive: Part 1: The Anatomy of an ESE database, April 2019. <https://techcommunity.microsoft.com/t5/ask-the-directory-services-team/ease-deep-dive-part-1-the-anatomy-of-an-ese-database/ba-p/400496> (Zugriff am: 01.03.2023).
- [50] Technische Hochschule Köln. Datenbanken Online Lexikon - ISAM. <https://wikis.gm.fh-koeln.de/Datenbanken/ISAM> (Zugriff am: 16.05.2023).
- [51] The SQLite Consortium. Database File Format. <https://www.sqlite.org/fileformat.html> (Zugriff am 25.05.2023).
- [52] A. C. F. Thomson. Windows 8 Forensic Guide, 2012. http://www.csc.villanova.edu/~dprice/extra_handouts/thomson_windows-8-forensic-guide.pdf (Zugriff am 11.07.2023).
- [53] X-Ways Software Technology AG. X-Ways Forensics 20.8. <https://www.x-ways.net/> (Zugriff am: 18.07.2023).

Abbildungsverzeichnis

3.1. Windows 10: Verfügbare Vorschaumodi im Datei-Explorer	13
3.2. Windows 10: Bestandteile des Datei-Explorer	13
3.3. Magic Bytes einer SQLite Datenbank	14
3.4. Magic Bytes einer ESE Datenbank	15
3.5. ESE Datenbank im Zustand <i>Clean Shutdown</i>	16
3.6. ESE Datenbank im Zustand <i>Dirty Shutdown</i>	16
3.7. Reparatur einer ESE Datenbank	17
4.1. Funktionsweise des Windows Thumbnail Cache	19
4.2. Aufbau der Thumbcache-Datenbank	20
4.3. Aufbau der Thumbcache-Indexdatei	23
4.4. Thumbcache-Ordner eines Windows 10 Systems	25
4.5. Thumbcache-Ordner eines Windows 11 Systems	25
4.6. Windows 11: Format Version im Dateiheder einer Thumbcache-Datenbank	27
4.7. Windows 10: Natürlich gewachsener Thumbnail Cache	32
6.1. Ablauf des ersten Untersuchungsschrittes	36
6.2. Ablauf des zweiten Untersuchungsschrittes	37
6.3. Windows 11: Durchgehend quadratische Darstellung bei Google Cloud Dateien	43
6.4. Windows 11: Ereignismeldung zum Laden des Treibers für Wechsel-datenträger	45
6.5. Windows 11: Vorschau eines Ordnerinhalts	47
6.6. Windows 10: Vorschau bei Drag-and-Drop	48
7.1. Testdatei für das Bilder Verzeichnis	50
7.2. Einbinden der virtuellen Festplatte zur Analyse	51
7.3. Windows 10: Betrachtung eines Thumbnails im Thumbcache Viewer .	51
7.4. Windows 10: Suchtreffer in ESEDatabaseView	52
7.5. WinSearchDBAnalyzer: Wiederhergestellter Eintrag	52
7.6. Windows 11: Suchtreffer in DB Browser for SQLite	53
7.7. Windows 10: Standardkonfiguration Windows Search	54

7.8. Windows 10: Hinzufügen eines Wechseldatenträgers zu Windows Search	55
8.1. Windows 11: Optionsmenü des Datei-Explorers	58
8.2. Windows 11: Thumbcache deaktivieren über die Systemsteuerung . .	59
8.3. Windows 11: Aktivierte Gruppenrichtlinien	60
8.4. Windows 11: Deaktivierter Thumbcache beim Datei-Explorer	60
8.5. Windows 11: Registry Schlüssel <i>IconsOnly</i>	61
8.6. Windows 11: Registry Schlüssel der beiden Gruppenrichtlinien	61
8.7. Windows 10: Windows Datenträgerbereinigung	63
8.8. WindowsPrefetchView: Details der cleanmgr.exe	64
8.9. Registry Explorer: Details der cleanmgr.exe	64
8.10. X-Ways Forensics: Ordner <i>ThumbCacheToDelete</i>	65
8.11. Windows 11: Löschen des Thumbcache im Systemmenü	65
8.12. X-Ways Forensics: Wiederhergestellte Thumbcache-Datenbanken . .	67
8.13. Thumbcache Viewer: Eintrag mit inkorrekten Prüfsummen	71
8.14. thumbcache_256.db: Manuell erstellter Eintragsheader	72
8.15. thumbcache_idx.db: Manuell erstellter Indexeintrag	72
8.16. Thumbcache Viewer: Manuell erstellter Thumbcache-Eintrag	73
8.17. X-Ways Forensics: Manuell erstellter Thumbcache-Eintrag	73
8.18. Dateieigenschaften des manuell erstellten und des legitimen Thumbcache-Eintrags	74
8.19. Windows 10: Manipulierte Vorschaufensteransicht	75

Tabellenverzeichnis

4.1. Thumbcache-Datenbanken nach Betriebssystem [42]	19
4.2. Thumbcache-Datenbank Header bei Windows Vista und Windows 7 [25]	21
4.3. Cache Typen für Windows Vista und Windows 7 [25]	21
4.4. Thumbcache-Eintrag im Windows Vista Format [25]	22
4.5. Thumbcache-Eintrag im Windows 7 Format [25]	22
4.6. Dateiheader der Indexdatei thumbcache_idx.db [25]	24
4.7. Indexeintrag im Windows Vista Format [25]	24
4.8. Indexeintrag im Windows 7 Format [25]	24
4.9. Thumbcache-Datenbank Header bei Windows 10 und Windows 11	27
4.10. Cache Versionen für Windows 10 und Windows 11	28
4.11. Thumbcache-Datenbank Footer bei Windows 10 und Windows 11	29
4.12. Windows 10 / 11: Eintragsheader einer Thumbcache-Datenbank	30
5.1. Auflistung der verwendeten Software	34
6.1. Liste der ursprünglich verwendeten Testdateien	37
6.2. Liste weiterer Testdateien	37
6.3. Vorschauoptionen und die von ihnen veränderten Thumbcache-Dateien	38
6.4. Änderungsverlauf des Thumbcache <i>klein</i> → <i>groß</i>	39
6.5. Änderungsverlauf des Thumbcache <i>groß</i> → <i>klein</i> bei quadratischen / breit rechteckigen Dateien	40
6.6. Änderungsverlauf des Thumbcache <i>groß</i> → <i>klein</i> bei schmal rechte- ckigen Dateien	40
6.7. Liste der verwendeten Testdateien	41

Verzeichnis der Abkürzungen

z.B.	zum Beispiel	11
bzw.	beziehungsweise	11
BSI	Bundesamt für Sicherheit in der Informationstechnik	12
ISAM	Index Sequential Access Method	15
VM	Virtuelle Maschine	33
NAS	Network Attached Storage	34
FIW	Fakultät für Ingenieurwissenschaften	37
BAM	Background Activity Moderator	64
NTFS	New Technology File System	67
MFT	Master File Table	67
EXIF	Exchangeable Image File Format	73
ggf.	gegebenenfalls	77

A. Anhang

Anhangsverzeichnis

A.1. Versionen der thumbcache.dll nach Betriebssystem	95
A.2. Thumbcache-Datenbank: Header → Footer → EOF	96
A.3. Dateiheader der Indexdatenbank - Windows 10 / 11	97
A.4. Eintrag der Indexdatenbank - Windows 10 / 11	98
A.5. Hostsystem der Virtualisierung	99
A.6. Konfiguration der Virtuellen Maschinen	99
A.7. Betriebssysteme der Virtuellen Maschinen	100
A.8. Tabellen zum Verhalten der Vorschauoptionen	101
A.9. Tabellen zum Verhalten der Vorschauoptionen (Netzlaufwerk)	103

A.1. Versionen der thumbcache.dll nach Betriebssystem

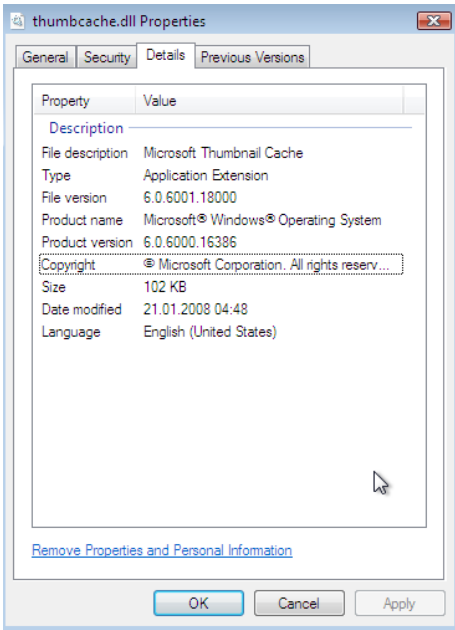


Abbildung A.1.: Windows Vista

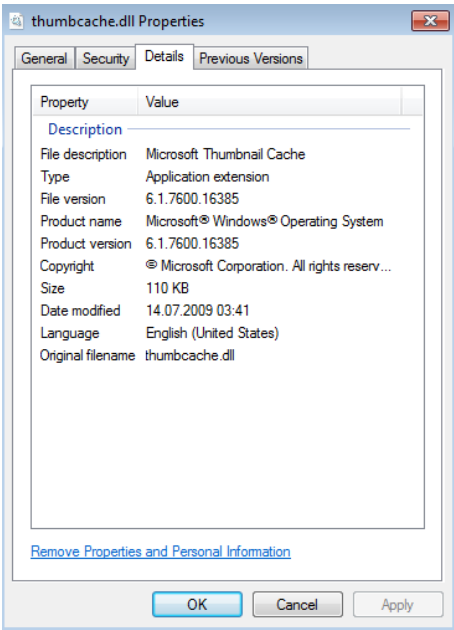


Abbildung A.2.: Windows 7

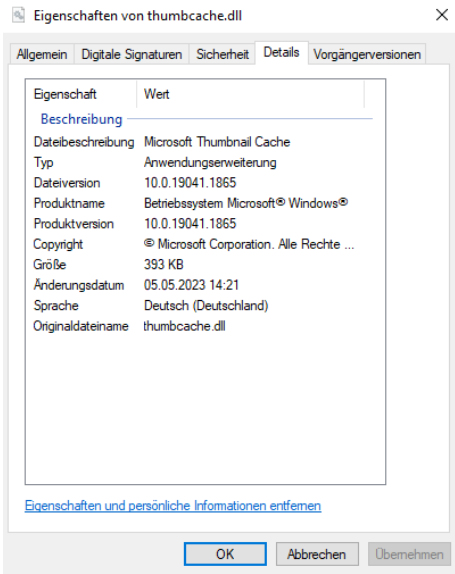


Abbildung A.3.: Windows 10

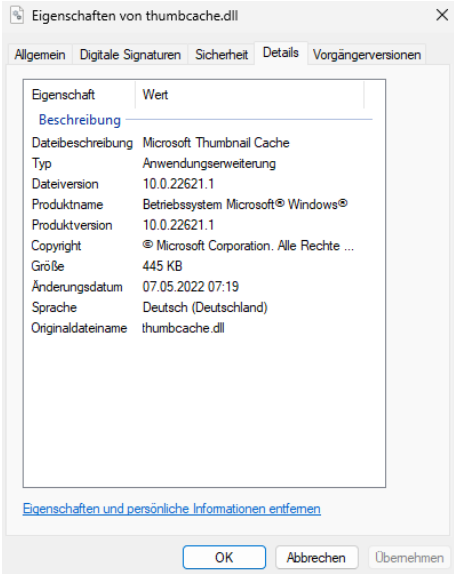
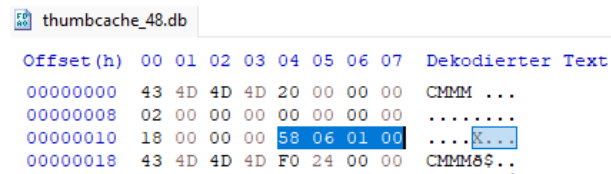


Abbildung A.4.: Windows 11

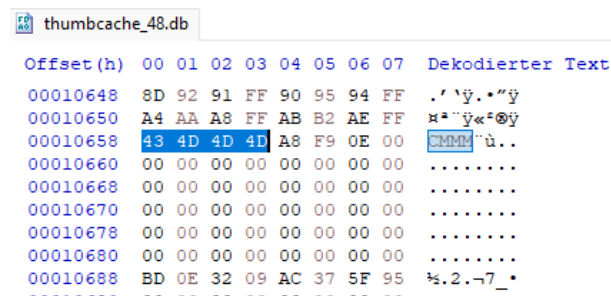
A.2. Thumbcache-Datenbank: Header → Footer → EOF

Der unten gezeigte Eintrag in der Datei *thumbcache_48.db* hat in seinem Header den Offset 0x10658 als Position seines Footers angegeben. Nach dem Identifier des Footers (CMMM) ist angegeben, dass in 0xEF9A8 Bytes die Datei endet.



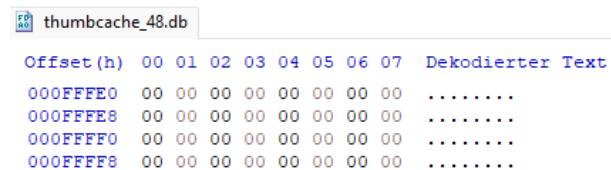
Offset (h)	00	01	02	03	04	05	06	07	Dekodierter Text
00000000	43	4D	4D	4D	20	00	00	00	CMMM ...
00000008	02	00	00	00	00	00	00	00
00000010	18	00	00	00	58	06	01	00X...
00000018	43	4D	4D	4D	F0	24	00	00	CMMM\$S..

Abbildung A.5.: Windows 11: Dateiheader einer *thumbcache_48.db*



Offset (h)	00	01	02	03	04	05	06	07	Dekodierter Text
00010648	8D	92	91	FF	90	95	94	FF	. 'ÿ. "ÿ
00010650	A4	AA	A8	FF	AB	B2	AE	FF	 "ÿ« "ÿ
00010658	43	4D	4D	4D	A8	F9	0E	00	CMMM"ü..
00010660	00	00	00	00	00	00	00	00
00010668	00	00	00	00	00	00	00	00
00010670	00	00	00	00	00	00	00	00
00010678	00	00	00	00	00	00	00	00
00010680	00	00	00	00	00	00	00	00
00010688	BD	0E	32	09	AC	37	5F	95	.2.¬7_*

Abbildung A.6.: Windows 11: Footer einer *thumbcache_48.db*



Offset (h)	00	01	02	03	04	05	06	07	Dekodierter Text
000FFFFE0	00	00	00	00	00	00	00	00
000FFFFE8	00	00	00	00	00	00	00	00
000FFFFF0	00	00	00	00	00	00	00	00
000FFFFF8	00	00	00	00	00	00	00	00

Abbildung A.7.: Windows 11: Dateiende einer *thumbcache_48.db*

A.3. Dateiheader der Indexdatenbank - Windows 10 / 11

Offset (Byte)	Größe (Byte)	Beschreibung
0	4	Unbekannt
4	4	Signatur (IMMM)
8	4	Format Version (0x20)
12	4	unbekannt
16	4	unbekannt
20	4	Anzahl der vorhandenen Indexeinträge
24	4	Anzahl der möglichen Indexeinträge bei der aktuellen Dateigröße
28	4	Dateigröße in Byte der thumbcache_16.db
32	4	Dateigröße in Byte der thumbcache_32.db
36	4	Dateigröße in Byte der thumbcache_48.db
40	4	Dateigröße in Byte der thumbcache_96.db
44	4	Dateigröße in Byte der thumbcache_256.db
48	4	Dateigröße in Byte der thumbcache_768.db
52	4	Dateigröße in Byte der thumbcache_1280.db
56	4	Dateigröße in Byte der thumbcache_1920.db
60	4	Dateigröße in Byte der thumbcache_2560.db
64	20	unbekannt
84	4	Anzahl der Thumbnails der thumbcache_16.db
88	4	Anzahl der Thumbnails der thumbcache_32.db
92	4	Anzahl der Thumbnails der thumbcache_48.db
96	4	Anzahl der Thumbnails der thumbcache_96.db
100	4	Anzahl der Thumbnails der thumbcache_256.db
104	4	Anzahl der Thumbnails der thumbcache_768.db
108	4	Anzahl der Thumbnails der thumbcache_1280.db
112	4	Anzahl der Thumbnails der thumbcache_1920.db
116	4	Anzahl der Thumbnails der thumbcache_2560.db
120	24	unbekannt

Tabelle A.1.: Dateiheader einer Thumbcache-Indexdatei bei Windows 10 und Windows 11

Es ist anzunehmen, dass die Datenfelder, die auf die beiden Einträge der thumbcache_2560.db folgen, die noch fehlenden Thumbcache-Datenbanken repräsentieren. Da diese Datenbanken von den durchgeführten Experimenten jedoch nicht verändert wurden, konnte die Annahme nicht untermauert werden.

A.4. Eintrag der Indexdatenbank - Windows 10 / 11

Offset (Byte)	Größe (Byte)	Beschreibung
0	8	Thumbcache-ID
8	4	vermutlich Indikator der bisher größten genutzten Vorschau
12	4	unbekannt, immer 0x00
16	4	Offset in thumbcache_16.db
20	4	Offset in thumbcache_32.db
24	4	Offset in thumbcache_48.db
28	4	Offset in thumbcache_96.db
32	4	Offset in thumbcache_256.db
36	4	Offset in thumbcache_768.db
40	4	Offset in thumbcache_1280.db
44	4	Offset in thumbcache_1920.db
48	4	Offset in thumbcache_2560.db
52	20	unbekannt

Tabelle A.2.: Eintrag in der Indexdatei bei Windows 10 und Windows 11

Wird einem Eintrag ein neuer Offset hinzugefügt und existiert kein vorhandener Offset in einer Thumbcache-Datenbank mit höheren Pixeldimensionen, so erhöht sich der Indikatorwert an Offset 8 des Indexeintrags. Wird das Vorschaufenster genutzt, verändert sich der Indikatorwert ab diesem Zeitpunkt nicht mehr, selbst wenn später ein Offset in einer Thumbcache-Datenbank gesetzt wird, die eine neue maximale Pixeldimension darstellt. Die Indikatorwerte sind dabei immer ein Vielfaches von 0x1001 (dezimal: 4097).

Es ist anzunehmen, dass die Datenfelder, die auf den Offset in thumbcache_2560.db folgen, die noch fehlenden Thumbcache-Datenbanken repräsentieren. Da diese Datenbanken von den durchgeführten Experimenten jedoch nicht verändert wurden, konnte die Annahme nicht belegt werden.

A.5. Hostsystem der Virtualisierung

Betriebssystem	Debian GNU/Linux 11.7 (bullseye), Kernel 5.10.0-23-amd64
CPU	Intel™ Core® i5-10210U CPU @ 1.60GHz × 8
RAM	16 GB
Datenspeicher	NVMe SSD 1 TB

Tabelle A.3.: Spezifikationen des Hostsystems

A.6. Konfiguration der Virtuellen Maschinen

Windows 10

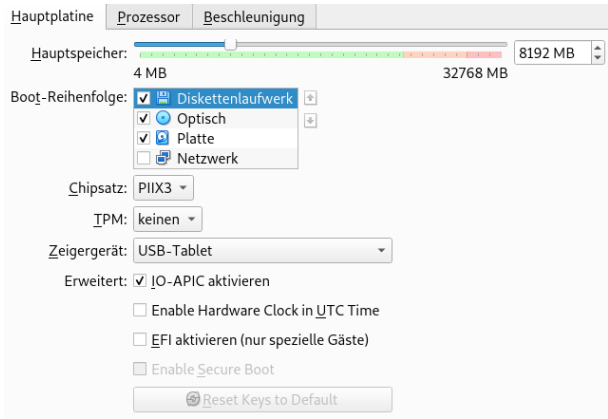


Abbildung A.8.: Windows 10: Systemkonfiguration

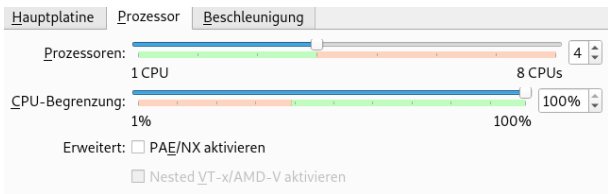


Abbildung A.9.: Windows 10: CPU Konfiguration

Windows 11

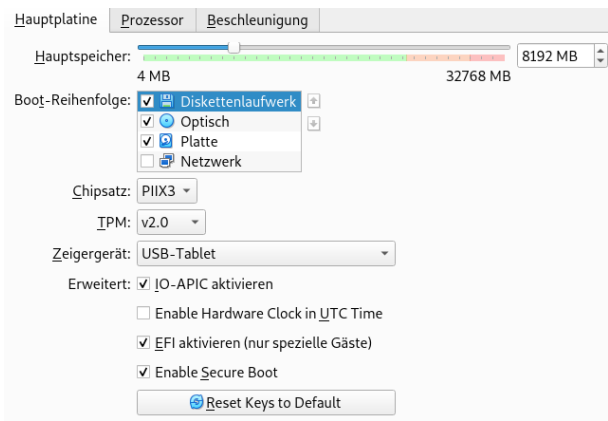


Abbildung A.10.: Windows 11: Systemkonfiguration

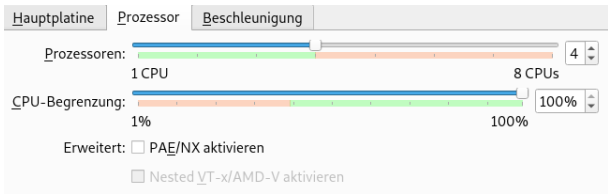


Abbildung A.11.: Windows 11: CPU Konfiguration

A.7. Betriebssysteme der Virtuellen Maschinen

Windows 10

Betriebssystem	Windows 10 Pro
Version	22H2 (Build 19045.3086)
Bildschirmauflösung	1680 x 1050
Dateisystem	NTFS

Tabelle A.4.: Details des Windows 10 Testsystems

Windows 11

Betriebssystem	Windows 11 Pro
Version	22H2 (Build 22621.1848)
Bildschirmauflösung	1680 x 1050
Dateisystem	NTFS

Tabelle A.5.: Details des Windows 11 Testsystems

A.8. Tabellen zum Verhalten der Vorschauoptionen

Quadratische und breit rechteckige Dateien

Vorschauoption	Betroffene Thumbcache-Datei
Inhalt	thumbcache_32.db
Kacheln	thumbcache_48.db
Mittelgroße Symbole	thumbcache_48.db
Große Symbole	thumbcache_256.db
Extra Große Symbole	thumbcache_256.db
Vorschauenfenster minimal	thumbcache_256.db thumbcache_1280.db
Vorschauenfenster maximal	thumbcache_2560.db

Tabelle A.6.: Vorschauoptionen und die von ihnen veränderten Thumbcache-Dateien

Vorschauoption	32	48	256	1280	2560
Inhalt	×				
Kacheln	×	×			
Mittelgroße Symbole	×	×			
Große Symbole	×	×	×		
Extra Große Symbole	×	×	×		
Minimales Vorschauenfenster	×	×	×	×	
Maximales Vorschauenfenster (Endzustand)	×	×	×	×	×

Tabelle A.7.: Änderungsverlauf des Thumbcache *klein* → *groß*

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster			×		×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt (Endzustand)	×	×	×		×

Tabelle A.8.: Änderungsverlauf des Thumbcache *groß* → *klein*

Schmal rechteckige Dateien

Vorschauoption	Betroffene Thumbcache-Datei
Inhalt	thumbcache_32.db
Kacheln	thumbcache_48.db
Mittelgroße Symbole	thumbcache_48.db
Große Symbole	thumbcache_256.db
Extra Große Symbole	thumbcache_256.db
Vorschaufenster minimal	thumbcache_1280.db
Vorschaufenster maximal	thumbcache_2560.db

Tabelle A.9.: Vorschauoptionen und die von ihnen veränderten Thumbcache-Dateien

Vorschauoption	32	48	256	1280	2560
Inhalt	×				
Kacheln	×	×			
Mittelgroße Symbole	×	×			
Große Symbole	×	×	×		
Extra Große Symbole	×	×	×		
Minimales Vorschaufenster	×	×	×	×	
Maximales Vorschaufenster (Endzustand)	×	×	×	×	×

Tabelle A.10.: Änderungsverlauf des Thumbcache *klein* → *groß*

A.9. TABELLEN ZUM VERHALTEN DER VORSCHAUPTIONEN (NETZLAUFWERK)

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster					×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt (Endzustand)	×	×	×		×

Tabelle A.11.: Änderungsverlauf des Thumbcache *groß* → *klein*

A.9. Tabellen zum Verhalten der Vorschaupptionen (Netzlaufwerk)

Quadratische und breit rechteckige Dateien

Vorschauoption	Betroffene Thumbcache-Datei
Inhalt	thumbcache_256.db
Kacheln	thumbcache_256.db
Mittelgroße Symbole	thumbcache_256.db
Große Symbole	thumbcache_256.db
Extra Große Symbole	thumbcache_256.db
Verschaufenster minimal	thumbcache_256.db thumbcache_1280.db
Verschaufenster maximal	thumbcache_2560.db

Tabelle A.12.: Vorschaupoptionen und die von ihnen veränderten Thumbcache-Dateien

Vorschauoption	32	48	256	1280	2560
Inhalt			×		
Kacheln		×	×		
Mittelgroße Symbole		×	×		
Große Symbole		×	×		
Extra Große Symbole		×	×		
Minimales Vorschaufenster		×	×	×	
Maximales Vorschaufenster (Endzustand)		×	×	×	×

Tabelle A.13.: Änderungsverlauf des Thumbcache *klein* → *groß*

A.9. TABELLEN ZUM VERHALTEN DER VORSCHAUOPTIONEN (NETZLAUFWERK)

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster			×		×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt (Endzustand)	×	×	×		×

Tabelle A.14.: Änderungsverlauf des Thumbcache *groß* → *klein*

Schmal rechteckige Dateien

Vorschauoption	Betroffene Thumbcache-Datei
Inhalt	thumbcache_256.db
Kacheln	thumbcache_256.db
Mittelgroße Symbole	thumbcache_256.db
Große Symbole	thumbcache_256.db
Extra Große Symbole	thumbcache_256.db
Vorschaufenster minimal	thumbcache_1280.db
Vorschaufenster maximal	thumbcache_2560.db

Tabelle A.15.: Vorschauoptionen und die von ihnen veränderten Thumbcache-Dateien

Vorschauoption	32	48	256	1280	2560
Inhalt			×		
Kacheln		×	×		
Mittelgroße Symbole		×	×		
Große Symbole		×	×		
Extra Große Symbole		×	×		
Minimales Vorschaufenster		×	×	×	
Maximales Vorschaufenster (Endzustand)		×	×	×	×

Tabelle A.16.: Änderungsverlauf des Thumbcache *klein* → *groß*

**A.9. TABELLEN ZUM VERHALTEN DER VORSCHAUOPTIONEN
(NETZLAUFWERK)**

Vorschauoption	32	48	256	1280	2560
Maximales Vorschaufenster					×
Minimales Vorschaufenster					×
Extra Große Symbole			×		×
Große Symbole			×		×
Mittelgroße Symbole		×	×		×
Kacheln		×	×		×
Inhalt (Endzustand)	×	×	×		×

Tabelle A.17.: Änderungsverlauf des Thumbcache *groß* → *klein*