Hochschule Wismar
Fakultät für Ingenieurwissenschaften
Studiengang IT-Forensik

IT-Forensik-Projekt II

DISSECT Tools zur effizienten

Triage von DISK-Images

von: F. Zeilhofer

Modulverantwortliche Dozentin: Frau Prof. Dr. Antje Raab-Düsterhöft

Datum: 16.10.2023

Inhaltsverzeichnis

Abkürzungsverzeichnis

bzgl.	bezüglich
CFReDS	Computer Forensic Reference Data Sets
DD	Disk Dump / auch: Disk Duplicator
EWF	Expert Witness Format
NIST	National Institute of Standards and Technology
o.g.	oben genannt
OS	Operating System (Betriebssystem)
u.a.	unter anderem
USA	United States of America
USD	US-Dollar
VHD	Virtual Hard Disk
VMDK	Virtual Machine Disk
WSL	Windows Subsystem für Linux
z.B.	zum Beispiel

Aufgabenstellung

Diese Arbeit beschäftigt sich mit der raschen und effizienten Extraktion von relevanten Informationen aus forensischen Datenträger-Abbildern, sogenannten Disk-Images. Die extrahierten Informationen sollen dazu dienen einen schnellen Überblick über eine Vielzahl von zu untersuchenden Datenträgern zu erhalten, um eine Sortierung nach Relevanz bzw. Dringlichkeit zu ermöglichen. Dieser als Triage bezeichnete Vorgang soll den Arbeitsablauf in Digitalen Forensik Laboren erleichtern und dies im Idealfall ohne großen Mehraufwand oder Verursachung von Kosten.

Hierzu wurden diverse auf dem Markt verfügbare Softwarelösungen zur Auswahl herangezogen. Nach Filterung der Optionen fiel der Fokus dabei auf die Open Source Tools „Dissect“ von Fox-IT, die auf den ersten Blick alle gewünschten Kriterien erfüllen.

Diese Arbeit widmet sich daher der Fragestellung, ob sich die „Dissect Tools“ als Triage-Werkzeug im forensischen Alltag eignen und sie die benötigten Anforderungen hierfür erfüllen können.

Nicht Gegenstand dieser Ausarbeitung sind hingegen die Darstellung und Bewertung aller Funktionen die von der Dissect Toolsammlung zur Verfügung gestellt werden. Die forensischen Möglichkeiten der Dissect Toolsammlung, der durch die Entwickler darüber hinaus auch noch laufend erweitert wird, ist weitaus umfangreicher, als wie er für die Triage von Disk-Images benötigt wird.

Für die Betrachtung wurden die benötigten Anforderungskriterien definiert und mithilfe eines Testaufbaus mit unterschiedlichen Disk-Images geprüft. Hierfür wurden Referenzwahrheiten aus den Disk-Images manuell extrahiert und analysiert, um die Ergebnisse aus Dissect damit abgleichen zu können.

Die daraus gewonnenen Erkenntnisse sollen Aufklärung darüber bringen, ob sich Dissect als Triage-Werkzeug im forensischen Alltag eignet und in einen bestehenden Arbeitsablauf ohne großen Mehraufwand integrieren lässt.

Die Ergebnisse dieser Untersuchung sollen dazu beitragen, die Korrektheit und die Effizienz von Dissect in der IT-Forensik zu bewerten und die Entscheidungsfindung über die Integration von Dissect in bestehende Workflows für die digitale Forensik zu unterstützen.

1. Einleitung

Dieser Abschnitt soll Informationen zu der Motivation hinter der Arbeit geben, die verfolgte Zielsetzung darstellen sowie einen Überblick bieten über die gewählte Vorgehensweise. Darüber hinaus wird aufgezeigt, was diese Arbeit nicht darstellen soll.

1.1. Motivation

Wenn die Anzahl der zu untersuchenden EDV-Geräte die personellen oder zeitlichen Grenzen von IT-Forensik-Laboren überschreitet, müssen Maßnahmen ergriffen werden, den wachsenden Mengen an digitalen Beweismitteln begegnen zu können.

Eine Triage der eingehenden Beweismittel stellt hierbei einen vielversprechenden Optimierungsansatz dar. Leider kann einem Computer oder Datenträger „von außen“ nicht angesehen werden, ob sich darauf relevante Informationen befinden, somit bleibt ein Blick auf die Inhalte unvermeidlich. Dies setzt unter forensischen Gesichtspunkten die Anfertigung von Datensicherung aller Asservate in Form von Disk-Images voraus, auf die nicht verzichtet werden sollte. [1]

Die für die Sicherungen benötigte Zeit ist von dem zu sichernden Quellmedium (Lesegeschwindigkeit), dem Sicherungsziel (Schreibgeschwindigkeit) sowie der Sicherungshardware (z.B. Forensik-Workstation oder autarker Hardware-Schreibschutz) abhängig. Die Sicherungsdauer eines einzelnen Beweismittels lässt sich aufgrund der drei genannten Faktoren bei beschränkten Hardware-Ressourcen nicht beschleunigen, daher muss hier eine andere Stellschraube zur Workflow-Optimierung gefunden werden. *

Mit der Einführung einer geeigneten Software zur Durchführung einer Triage können Disk Images nach deren Erstellung rasch nach festgelegten Kriterien sortiert und je nach Dringlichkeit oder Expertise zugeteilt werden. Diese Vorgehensweise erleichtert es, auch bei einer großen Menge an Fällen, den Überblick zu behalten und Ressourcen effektiv zu nutzen. Ebenfalls ermöglicht es die priorisierte Behandlung von zeitkritischen Fällen.

** Einzig eine Parallelisierung der Sicherungsvorgänge kann die Gesamtdauer der Sicherungen für eine Vielzahl an Datenträgern reduzieren. Je mehr Sicherungs-Hardware verfügbar ist, desto mehr kann parallel anstatt nacheinander gesichert werden. Der Sicherungsvorgang an sich benötigt, außer der Konfiguration und der abschließenden Prüfsummenverifikation, kaum menschliche Aufmerksamkeit.*

Durch die Erkenntnisse einer Triage können Entscheidungen über die weitere Vorgehensweise der Auswertung getroffen und die Notwendigkeit bestimmter Verarbeitungsschritte festgelegt werden. Durch die Triage wäre auch eine umgehende Zuordnung zu speziellen Abteilungen (z.B. Windows- / Linux-Forensik) oder Spezialisten durch die aus der Triage gewonnenen Erkenntnisse möglich.

Aus diesen Beobachtungen entstand die Motivation eine geeignete Triage-Lösung zu finden, um den forensischen Alltag effizienter zu gestalten.

1.2. Zielsetzung der Arbeit

Das primäre Ziel dieser Arbeit soll die Identifizierung eines geeigneten Werkzeugs zur Triage von Disk Images in der digitalen Forensik sein und ob sich diese ergänzende Maßnahme der Triage im Forensik-Alltag als unnötige und zeitaufwändige Mehrarbeit entpuppt oder als gewinnbringende und effiziente Workflowergänzung herausstellt.

Die Betrachtung liegt hier insbesondere auf dem Open Source Tool „Dissect“ der Firma Fox-IT und ob es die erforderlichen Kriterien im Vergleich zu anderen Triage-Programmen erfüllt. „Dissect“ wird dabei eingehend auf die Genauigkeit seiner Ergebnisse geprüft und der für den Einsatz der Software benötigte Zeitaufwand mit den Forensikprogrammen „X-Ways Forensics“ und „Magnet Axiom“ verglichen, um die Anwendbarkeit im forensischen Kontext zu beurteilen.

Des Weiteren strebt diese Arbeit an, zu evaluieren, ob und inwieweit „Dissect“ in den bestehenden Workflow eines forensischen Labors integriert werden kann, um eine nahtlose und effiziente Triage von Disk Images zu ermöglichen. Hierbei soll nicht nur die technische Machbarkeit, sondern auch der potenzielle Mehrwert einer solchen Integration beleuchtet werden.

1.3. Abgrenzung

Zur Verdeutlichung der Intention dieser Arbeit werden nachfolgend die Grenzen dieses Vorhabens definiert. So beschränkt sich die Untersuchung und Analyse der Software „Dissect“ ausschließlich auf die für die Triage von Disk Images relevanten Funktionen. Eine Vorstellung und Prüfung aller verfügbaren Features dieses auch als Framework dienenden Open Source Projekts wird nicht vorgenommen. Weiterhin befasst sich die Arbeit nicht mit den Möglichkeiten der Live-Triage, also der Informationsgewinnung von laufenden Systemen. Stattdessen fokussiert sich diese Arbeit konsequent auf den Aspekt der Triage von bereits erstellten Disk-Images. Um die rasche Integration in einen bestehenden Workflow zu prüfen, erfolgte daher auch keine Bearbeitung oder Anpassung des Quellcodes von „Dissect“ an spezielle Gegebenheiten. Das Tool wird in seiner standardmäßigen, „Out-of-the-Box“ verfügbaren Version betrachtet, um die Anwendbarkeit für möglichst viele Forensik-Labor-Umgebungen sicherstellen zu können.

1.4. Grundlagen und Begriffe

Im Fachgebiet der Informatik und insbesondere im Bereich der Digitalen Forensik haben sich zahlreiche Begriffe etabliert, die häufig im täglichen Diskurs genutzt werden und ihren Ursprung oftmals in der englischen Sprache haben. Die im Rahmen dieser Arbeit regelmäßig verwendeten Fachbegriffe und auch Konzepte werden nachfolgend erläutert.

1.4.1. Open Source

Als Open Source wird eine Software bezeichnet, wenn der für die Erzeugung oder den Betrieb der Software zugrunde liegende Programmiercode (Quelltext) von den Entwicklern offen zugänglich gemacht wird. Dies bedeutet nicht gleichzeitig, dass eine Open Source Software immer kostenlos ist. Um Missverständnisse zu vermeiden und die Nutzungsbedingungen zu definieren, werden Quelltexte vom Urheber üblicherweise unter einer spezifischen Lizenz veröffentlicht, die bestimmt, wie der veröffentlichte Programmtext verwendet werden darf. Der Vorteil an Open Source Software besteht darin, dass Anwender mit den erforderlichen Kenntnissen den Code überprüfen und die Funktionsweise der Programme nachvollziehen können. [2]

1.4.2. Disk-Images

Als Disk Images oder auch Datenträgerabbilder werden Datenspiegelungen bezeichnet, die in der Digitalen Forensik vor jeglichen weiteren Schritten von den zu untersuchenden Datenträgern angefertigt werden. Im Kontext der Digitalen Forensik geschieht die Anfertigung einer solchen Datenspiegelung unter der Verwendung eines Soft- oder Hardware-Schreibschutzes, der potenzielle Schreibvorgänge auf den zu sichernden Datenträger durch den Auswerte-PC verhindert, um Änderungen an den Beweisquellen ausschließen zu können.

Bei den resultierenden Disk Images handelt es sich um Dateien, die von speziellen Programmen zur forensischen Analyse ausgelesen werden können. Dabei gibt es sowohl 1:1 Roh-Datensicherungen (Bit für Bit – Sicherung) als auch unterschiedliche forensische Formate, zum Teil mit integrierter Prüfsumme, um die Disk Images auf Ihre Integrität verifizieren zu können. Während den Untersuchungen wird ausschließlich mit den forensischen Kopien der Daten gearbeitet, sodass das Original nach der Sicherung unberührt bleibt.

1.4.3. Forensische Artefakte

Als forensische Artefakte werden in dieser Arbeit digitale Spuren bezeichnet, die potenziellen forensischen Mehrwert darstellen können. Dies können sowohl Dateien sein, die vom Benutzer bewusst abgelegt wurden, als auch die im System automatisch im Hintergrund erzeugten Protokolle, Logs oder Einstellungen. Für die Digitale Forensik können verschiedene Artefakte wertvolle Informationen liefern, um beispielsweise Einblick in die Art und Weise der Verwendung eines Computers zu erlangen. [3]

1.4.4. Triage

Der üblicherweise im Bereich der Notfallmedizin gebräuchliche Begriff „Triage“ zur „Sortierung“ verwundeter Menschen nach Dringlichkeit der medizinischen Behandlung anhand ihrer Überlebenschancen, kann in ähnlicher Weise auch in der Digitalen Forensik Anwendung finden. Hierbei wird jedoch nicht über Leben und Tod entschieden, sondern versucht, die Dringlichkeit bei der Auswertung einer Vielzahl von Geräten systematisch einzuordnen, insbesondere, wenn die zur Verfügung stehenden Ressourcen begrenzt sind. [4] [5]

1.4.5. Workflow

Als Workflow wird der typische Arbeitsablauf bezeichnet, den eine Person in der IT-Forensik bei der Verarbeitung der Untersuchungsgegenstände üblicherweise durchführt. Der Workflow umfasst dabei grundsätzlich alle Bearbeitungsschritte einer Untersuchung, die bereits bei der Entgegennahme des zu untersuchenden Objekts beginnt und mit der abschließenden Aushändigung des Untersuchungsgegenstandes mit dem entstandenen Bericht und den extrahierten Daten an den Auftraggeber endet. Weiterführende unterstützende Maßnahmen für die Auftraggeber nach Abschluss der Untersuchung, z.B. bei der Auswertung der Daten ist dagegen kein Bestandteil des typischen Workflows mehr.

Die in der Betrachtung dieser Arbeit liegende Fokus liegt folglich nach den vorbereitenden Schritten wie der Dokumentation des äußeren Zustands der Untersuchungsobjekte und der forensischen Datensicherung der Datenträgerinhalte als Disk-Images.

Nach der Erstellung dieser Datensicherung soll mithilfe einer Triage die Wahl des geeigneten Auswerte-Workflows getroffen werden. Dabei kann je nach Gerät auf praxiserprobte Standard-Workflows zurückgegriffen werden oder gar die Notwendigkeit entstehen, einen völlig neuen Workflow mit zielgerichteten Auswertezielen zu entwickeln.

Ohne Triage ist die Wahl eines geeigneten Workflows zur eigentlichen Auswertung jedoch kaum möglich und es kann passieren, dass die Untersuchungsgegenstände nicht sofort mit der effizientesten Vorgehensweise bearbeitet werden.

2. Triage Werkzeuge

Auf dem Markt existieren bereits Triage-Werkzeuge, die im Bereich der IT-Forensik eingesetzt werden, um effiziente und zielgerichtete Analysen digitaler Beweismittel vorzunehmen. Die Auswahl eines geeigneten-Tools kann dabei eine Herausforderung darstellen. Durch Festlegung von speziellen Anforderungskriterien, soll es gelingen, eine informierte Entscheidung zu treffen, die den benötigten Bedürfnissen gerecht wird. Diese werden nachfolgend beschriebenen.

2.1. Anforderungen

Um zu einem geeigneten Tool zu gelangen, wurden Kernanforderungen formuliert, die zum Vergleich der angebotenen Lösungen herangezogen wurden.

Open Source

Um die Funktionsweise der Software nachvollziehen zu können und bei Bedarf zu erweitern, wäre eine Open Source Lösung wünschenswert. Dies stellte jedoch kein Ausschlusskriterium dar.

Unterstützung von Offline Disk Images (Ausschlusskriterium)

Die Software muss die Fähigkeit besitzen, „offline“ Disk-Images direkt zu analysieren. Im Idealfall kann die Lösung bestehende Datenträgersicherungen direkt einlesen, ohne, dass diese vorab durch eine andere Software gemountet oder extrahiert werden müssen. In diesem Zusammenhang wird nochmals darauf hingewiesen, dass eine Live Triage nicht benötigt wird, da die Software nicht auf laufenden Rechnern zum Einsatz kommen wird.

Offline-Funktionalität (Ausschlusskriterium)

Ein weiterer entscheidender Aspekt ist die Fähigkeit, offline, also ohne aktive Internetverbindung funktionieren zu können. Dies ist relevant, da die Auswerte-Stationen in Forensik-Laboren aus Sicherheitsgründen meist nicht mit dem Internet verbunden sind.

Kompatibilität mit Windows (Ausschlusskriterium)

Die gesuchte Lösung muss nativ unter dem Betriebssystem Microsoft Windows laufen oder mit dem Windows Subsystem für Linux (WSL) ausführbar sein, um eine Installation und Integration in bestehende Systemlandschaften zu ermöglichen. Da das Triage Tool als Ergänzung für X-Ways Forensics und Magnet AXIOM dienen soll (beides reine Windows Applikationen), muss es somit auf dem gleichen System ausführbar sein.

Kosteneffizienz (Ausschlusskriterium)

Da das Tool lediglich zur Ergänzung und Optimierung des Workflows dienen und nicht die bestehenden, bewährten Forensik-Suiten ersetzen soll, darf es keine hohen Kosten verursachen.

2.2. Marktüberblick – potenzielle Kandidaten

Eine Marktschau brachte folgende Kandidaten hervor, die gemäß der unter Punkt 2.1 genannten Anforderungen betrachtet wurden. Alle betrachteten Programme können unter Windows bzw. WSL betrieben werden und sind ohne Internetverbindung nutzbar (Offline), daher werden diese beiden Kriterien als erfüllt betrachtet und in der nachfolgenden Tabelle nicht als separate Spalten aufgeführt.

			Ausschlusskriterien		
Hersteller	**Produkt**	**Open Source**	**Verarbeitet Disk-Images**	**Preis**	
Fox-IT	**Dissect**	ja	Ja	Kostenlos	
Sleuth Kit Labs (Brian Carrier)	**CYBER TRIAGE**	Nein	Ja	Lite kostenlos	$2.500 USD/Jahr
KROLL (Eric Zimmermann)	**KAPE**	Nein (nur „Module“)	Nein	Kostenlos für: „interne Ermittlungen“, Law Enforcement & akademisch	
Magnet Forensics	**OUTRIDER**	Nein	Nein	„auf Anfrage“	
GetData	**FEX Triage**	Nein	Nein	$ 495 USD / Jahr	

Tabelle 1: Übersicht der erfüllten Anforderungen der potenziellen Triage Software

- Grün hinterlegte Zellen erfüllen das gewünschte Kriterium vollumfänglich.
- Gelb hinterlegte Zellen erfüllen das Kriterium unter bestimmten Bedingungen.
- Rote Zellen können die Anforderungen nicht erfüllen.

Die letzten beiden Spalten stellen dabei Ausschlusskriterien dar. Somit wurden Programme, die mindestens eine der beiden Kategorien nicht erfüllt, in der weiteren Betrachtung nicht weiter berücksichtigt, da diese den festgelegten Kernanforderungen nicht entsprechen.

2.3. Entscheidung für Dissect

Ein Blick auf die Tabelle in Bild 1 zeigt, dass lediglich „Dissect“ von Fox-IT alle Kriterien vollumfänglich erfüllt. Im Zuge der Ausarbeitung wurde auch eine Testversion von „CYBER TRIAGE“ von Sleuth Kit Labs geprüft. Während des gesamten Zeitraums der Erstellung dieser Arbeit wurde vom Hersteller jedoch nur eine eingeschränkte veraltete Version (3.6) als Lite Variante angeboten – die aktuelle Version (3.8), war nur gegen Bezahlung erhältlich. Nachdem „CYBER TRIAGE“ zudem auch dem optionalen Kriterium „Open Source“ nicht gerecht wird, wurde letztendlich entschieden, den Fokus dieser Arbeit vollständig auf „Dissect“ von Fox-IT zu legen, bei der in Hinblick auf die festgelegten Kriterien keine Abstriche gemacht werden müssen. Im Anhang befinden sich zur Vollständigkeit Screenshots von der Installation und Bedienung von CYBER TRIAGE.

3. Dissect als Open Source Lösung

3.1. Über Dissect

Bei Dissect handelt es sich um ein auf der Programmiersprache Python basierendes Framework und Toolset für die Bereiche Digitalforensik und Incident Response. Es wurde vom „Cyber-Sicherheitsunternehmen „Fox-IT Holding B.V.“ (Niederlande) entwickelt, das seit 2015 Teil der „NCC Group plc“ (England/Wales) ist.[6]

Zunächst für „In-House“-Zwecke für den Eigengebrauch bzw. für den Einsatz bei Kunden entwickelt, wurde Dissect am 04. Oktober 2022 auf der Online-Plattform GitHub veröffentlicht.[7] Das Projekt wurde dabei Open Source unter der AGPL-3.0-Lizenz (GNU Affero General Public License v3.0) veröffentlicht, die auch eine kommerzielle Nutzung und Modifikation der angebotenen Werkzeuge gestattet. Dissect ermöglicht es, schnell auf forensische Artefakte aus verschiedenen Festplattensicherungsformaten zuzugreifen und diese zu extrahieren.

Gemäß der Projektseite „https://github.com/fox-it/dissect“ wird Dissect als eine Lösung mit „einheitlichem Ansatz“ beschrieben, da es gemäß Angaben unabhängig vom zugrundeliegenden Container (wie zum Beispiel: E01, VMDK, QCoW), dem Dateisystem (z.B.: NTFS, ExtFS, FFS) oder Betriebssystem (Windows, Linux, ESXi) eingesetzt werden kann.

3.2. Installation in Testumgebung

Um die Tests für diese Arbeit durchzuführen wurde entschieden, Dissect nicht direkt auf dem Windows-Host-Betriebssystem zu installieren, sondern diese stattdessen im so genannten „Windows-Subsystem für Linux Version 2“ (kurz: WSL) einzurichten. Bei WSL handelt es sich um eine ab 2016 in Windows 10 eingeführte Kompatibilitätsschicht, die die Ausführung einer Linux-Distribution in einer ressourcenschonenden Virtualisierungsumgebung direkt unter Windows ermöglicht. Dadurch wird die Ausführung von Linux-Befehlen und -software auf Kommandozeilenebene (unter Linux „Terminal“ genannt) möglich.

Ein weiterer Vorteil dabei ist, dass ohne großen Aufwand auf Dateien des Windows-Host-Systems zugegriffen werden kann und die Disk-Images somit nicht zwischen den beiden Systemen transferiert werden müssen. Externe USB-Speicher werden ebenfalls unterstützt.

Die wichtigsten Hard- und Software Parameter der Testumgebung sind nachfolgender Tabelle zu entnehmen.

Systemtyp (Bauform)	Tower-PC
CPU	Intel Core i7-5820K
Arbeitsspeicher	96 GB DDR 4
Datenträger (Betriebssystem)	SATA SSD 250 GB
Datenträger (Test-Image-Dateien)	M.2 SSD 1 TB PCIe 3.0 NVMe
Host-Betriebssystem	Windows 10 22H2 (Build 19045.3570)
Windows Subsystem für Linux	WSL Version 2 (Kernel Version 5.10.16)
WSL-OS	Ubuntu 22.04.2 LTS

Tabelle 2: Wesentliche Konfiguration der Testumgebung

Da WSL nicht standardmäßig in Windows-Systemen mitinstalliert wird, musste dies zunächst aktiviert werden. Der Installations- und Konfigurationsvorgang von WSL (Version 2) wird im Anhang mit Screenshots dargestellt. An dieser Stelle wird nur betont, dass Microsoft diesen Prozess sehr nutzerfreundlich gestaltet hat und es zu keinerlei Komplikationen dabei kam.

Von den als Linux-Distributionen im Microsoft Store zur Verfügung stehenden Linux-Systemen wurde sich für die populäre Variante Ubuntu 22.04.2 LTS entschieden, die sich durch ihre gute Dokumentation und große Kompatibilität auszeichnet.

Die Installation von Dissect selbst gestaltete sich ebenfalls einfach, da diese mithilfe des PIP-Installationsmanagers geladen werden kann. Eine Screenshot-Sammlung diesbezüglich ist dem Anhang zu entnehmen. Auch hier soll an dieser Stelle nur erwähnt werden, dass die Installation ohne Schwierigkeiten auf Anhieb funktioniert hat. Dies wird auch in den Entscheidungsprozess der Erwägung von DISSECT als Triage-Werkzeug zur Workflow-Ergänzung positiv mit einfließen.

Anmerkung: Die Installation von Dissect hätte auch direkt auf dem Windows-Host-System mit installierter Python-Umgebung funktioniert, da es sich aber um einen Test handelt und vorab nicht bekannt war ob und wie die Dissect-Tools mit bestehenden Python-Installationen kompatibel sind, wurde die Installation in eine WSL-Umgebung bevorzugt.

Aufgrund der gesammelten Erfahrungen kann dieses Vorgehen vom Verfasser als empfehlenswert bezeichnet werden, da somit die Dissect-Umgebung isoliert getestet und verglichen werden kann, ohne ein komplettes Dual-Boot-System oder eine vollständige Virtualisierungslösung einzusetzen.

Die im Vergleich zur Voll-Virtualisierung ressourcenschonende WSL-Umgebung ermöglicht bei Bedarf sowohl eine saubere Replikation der Testumgebung als auch eine komplette, rückstandsfreie Entfernung dieser. Tatsächlich waren die Erfahrungen mit WSL so positiv, dass diese Vorgehensweise auch für eine Produktivumgebung mit Dissect empfohlen werden kann.

3.3. Dissect Bestandteile

Dissect ist modular aufgebaut, d.h. es besteht aus verschiedenen unabhängigen Modulen, die jedoch eng miteinander interagieren. Die Modularität ist der Grund dafür, weshalb zu Dissect auf einfache Weise spezifische Funktionen bei Bedarf hinzugefügt werden können.

Bei der Standardinstallation werden alle Module mitinstalliert und es muss nicht vorab entschieden werden, welche Bestandteile für den Einsatzzweck benötigt werden. Eine ausschließliche Installation der benötigten Komponenten ist jedoch ebenfalls möglich.

Zwei große Unterscheidungen müssen zwischen den beiden Dissect-Bestandteilen „Projects“ und „Tools“ gemacht werden, auf die nachfolgend eingegangen wird.

3.3.1. Dissect „Projects“

Die „Projects“ stellen die grundlegende Sammlung der Bibliotheken und Tools von Dissect dar und bilden die Grundlage für die Funktionsweise von Dissect. Zu den Projects gehören auch die Bestandteile, die für die Handhabung der unterschiedlichen Dateisysteme, Container-Formate oder auch Forensik-Artefakte zuständig sind. Ohne das richtige „Project“ kann Dissect nicht auf die benötigten Dateien zugreifen. Insgesamt sind zum Zeitpunkt der Fertigstellung dieser Arbeit insgesamt 26 unterschiedliche „Projects“ in der offiziellen Dokumentation gelistet.

Die für die Sammlung der Triage-Informationen wichtigsten Bestandteile werden nachfolgend kurz beschrieben.

dissect.volume

Dieses „Project“ sorgt dafür, dass Dissect die unterschiedlichen **Festplattenpartitionsschemas** LVM2, GPT und MBR interpretieren kann und stellt somit die Grundlage für weitere Module dar.

dissect.ntfs, dissect.extfs, dissect.fat

Diese drei „Projects“ stellen einen Auszug der verfügbaren Parser für die jeweils namensgebenden **Dateisysteme** dar. Sie sorgen dafür, dass Daten von den entsprechenden Volumes ausgelesen werden können. Diese drei Module sind zuständig für: NTFS-, ExtFS-, FAT- und exFAT-Dateisysteme.

dissect.evidence

Dieses Dissect „Project“ ermöglicht es, auf die **forensischen Containerformate** AD1, ASDF und EWF („E01“) direkt zugreifen zu können.

dissect.hypervisor

Dieses „Project“ implementiert die Fähigkeit auf diverse **Festplattenformate sowie Backup- und Konfigurationsdateien von VM-Hypervisoren** zugreifen zu können. Zu den unterstützten Formaten zählen die **Backup-Formate** VMA und VXA, Die **Metadaten-Deskriptoren** von Hyper-V VMCX (HyperVFile), OVF und VMX, die **virtuellen Disk-Formate** QCOW2, VDI, VHD, VHDX, VMDK sowie die **ESXi Formate** envelope, key store und visortar/vmtar.

Dissect.eventlog

Dieses „Project“ ist für das Auslesen von **Windows Logdateien** der Formate EVT, EVTX and WEVT zuständig.

Diese Übersicht stellt keine vollständige Aufzählung aller „Projects“ dar, die für die geplante Triage erforderlich sind, sondern gibt vielmehr einen Einblick in den modularen Aufbau von Dissect.

3.3.2. Dissect „Tools“

Als „Tools“ werden in Dissect spezifische Implementationen und Anwendungen bezeichnet, die auf den Libraries und Funktionen aufbauen, die von den verschiedenen „Projects“ bereitgestellt werden.

Für die Triage haben sich zwei der insgesamt neun verfügbaren „Tools“ hervorgehoben, die sich als wirksame Werkzeuge zur Extraktion der benötigten Informationen eignen: „**target-query**“ und „**target-shell**“. Die Terminologie „**target**“ bezeichnet im Kontext von Dissect gemäß der Dokumentation jede Art der unterstützten Quelldaten und umfasst alles, was verwendet werden kann, um einen bestimmten Zustand eines Systems zu beschreiben.[8] Die beiden in dieser Arbeit behandelten „Tools“ werden nachfolgend kurz beschrieben.

3.3.3. Dissect: target-query

Mit dem „Tool“ target-query können gezielt die benötigten Informationen bzw. Artefakte aus den „targets“ extrahiert, angezeigt oder exportiert werden.

Durch Eingabe des Kommandozeilenbefehls target-query, gefolgt vom Quelldatenpfad und der expliziten Angabe der gewünschten Funktion können eine oder mehrere Informationen gleichzeitig aus den gewünschten Quelldaten extrahiert werden. Die genaue Syntax wird in Punkt 3.4 beschrieben.

Dies dient beispielsweise der Analyse von Log-Dateien, wie beispielsweise den Kommandozeilen-Logs (z.B. Bash, PowerShell usw.) oder der simplen Rückgabe des Hostnamens und der Betriebssystemversion des auf dem Quelldatenträger installierten Betriebssystems.

Diese Funktionen werden in Dissect von so genannten Plugins zur Verfügung gestellt. Dabei können eigene Plugins programmiert oder eines von den vielen bereits enthaltenen verwendet werden. Alle in Dissect target-query enthaltenen Funktionen können mit dem Befehl target-query -L abegrufen werden. Die für die Triage benötigten Befehle werden in dieser Arbeit unter Punkt 5 aufgelistet.

3.3.4. Dissect: target-shell

Mithilfe von target-shell kann eine „Fixierung“ auf eine bestimmte Datenquelle (oder sogar mehrere) erfolgen. Durch Eingabe des Befehls target-shell gefolgt vom gewünschten Quellpfad erhält man Zugang zu einer virtuellen Shell-Umgebung. Sobald diese geöffnet ist, reicht die Eingabe bestimmter Befehle, um so schneller an die gewünschten Informationen zu gelangen. Eine erneute Angabe des Quellpfads entfällt dadurch. Innerhalb der Shell können mit dem Befehl „help“ die wichtigsten Befehle angezeigt werden.

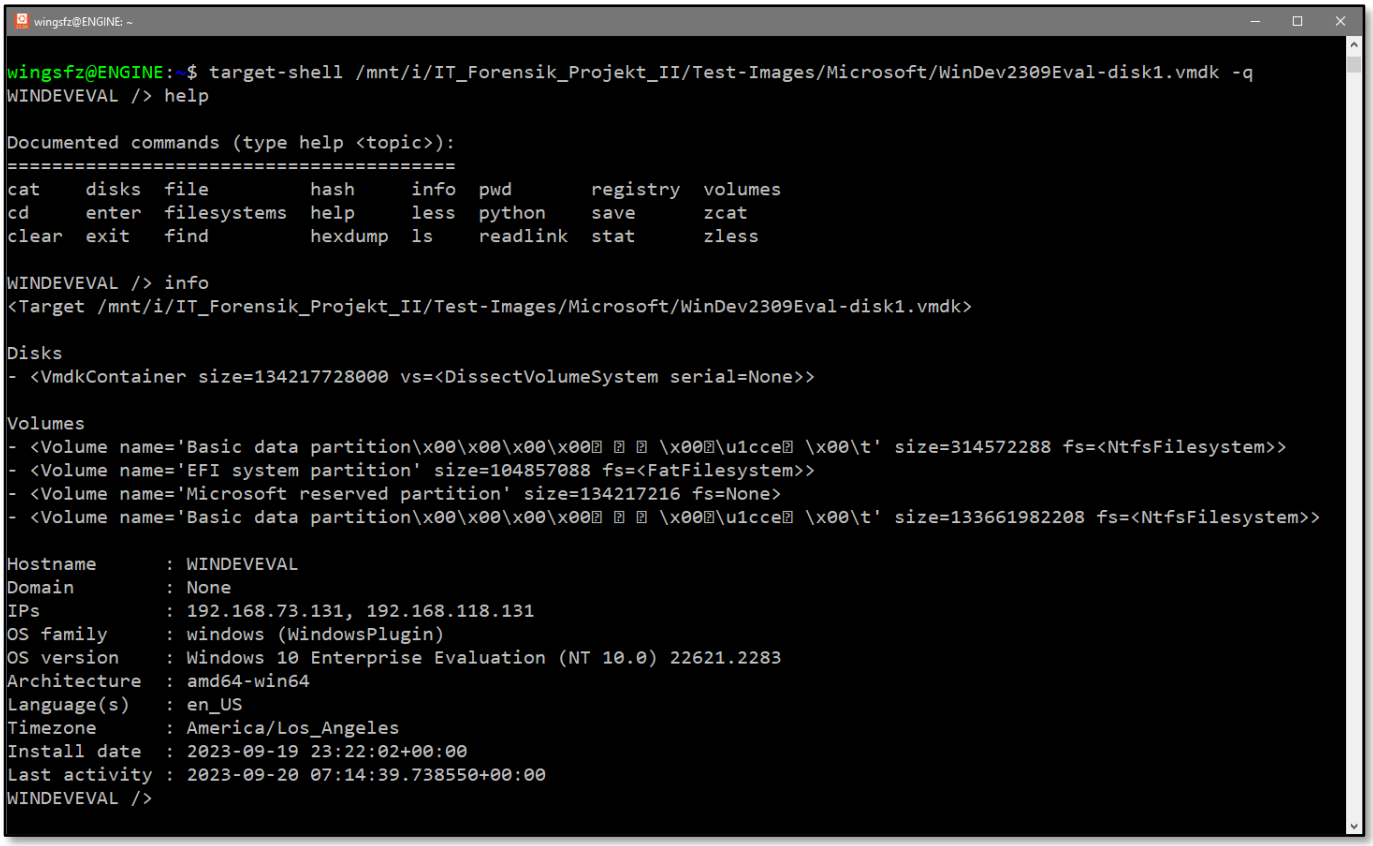


Bild 1: Ausführung und Anwendung der Befehle **help** und **info** innerhalb der **target-shell**

3.4. Dissect - Verwendung und Syntax

Mithilfe von Kommandozeilenbefehlen können in Dissect die Funktionen der mitgelieferten „Tools“ und „Projects“ abgerufen werden. Die Syntax der Befehle für die im Rahmen dieser Arbeit zu extrahierenden Triage-Informationen wird nachfolgend vorgestellt. Zur Ausführung muss lediglich das gewünschte „Tool“ gefolgt von den Parametern eingegeben werden.

3.4.1. Syntax: target-query

Die folgende Zeile stellt eine einfache Standardabfrage mit target-query dar:

```
>target-query /Pfad_zum_Image/Beispieldatei.vmdk -f <FUNKTIONSNAME> -q
```

target-query	Ausführung einer **target-query** Abfrage.
/Pfad_zum_Image/	Hier wird der **Pfad zur Quelldatei** angegeben.
Beispieldatei.vmdk	An dieser Stelle steht der **Dateiname des Quellimages**. Ein ***** als **Platzhalter** kann verwendet werden, um beispielsweise ALLE VMDK-Dateien eines Pfades auszuwerten (/*****.vmdk). Bei aufgesplitteten Images wird nur die erste Datei angegeben (z.B. E01 oder DD.01, etc.).
-f <FUNKTIONSNAME>	Die **gewünschte Funktion** (bzw. das gewünschte Plugin) wird durch **-f** gefolgt von dem Funktionsnamen ausgeführt.
-q	**Optional**: durch **-q** (steht für „quiet“) werden nicht kritische Warnmeldungen auf der Kommandozeile unterdrückt.

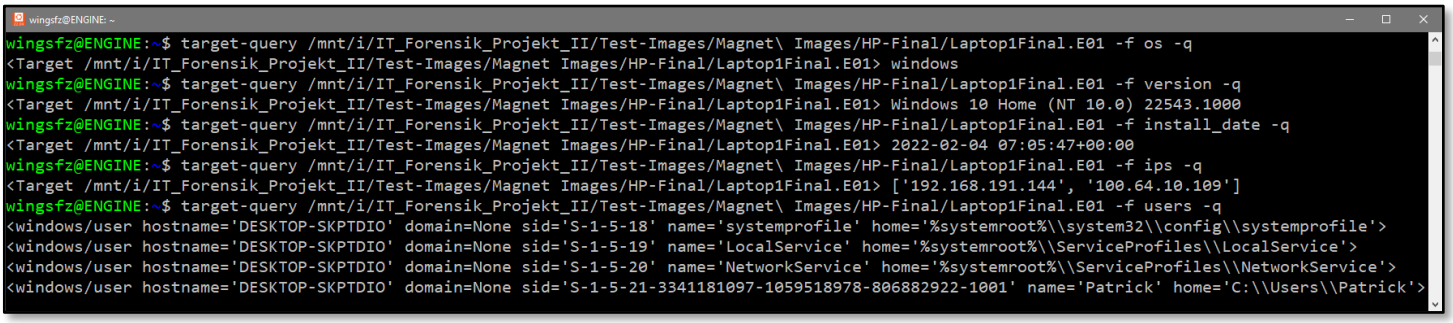


Bild 2: Beispielabfragen & -ausgaben mit **target-query** und den Funktionen: **os, version, install_date, ips und users**

Die im Bild 2 verwendeten Funktionen und deren Bedeutung:

os	Betriebssystemfamilie
version	Version des festgestellten Betriebssystems
install_date	Installationszeitpunkt des Betriebssystems
ips	IP-Adressen des Quell-Systems
users	eingerichteten Benutzerkonten

4. Testablauf und Durchführung

4.1. Bewertungskriterien - Definition des Mehrwertes

Um als sinnvolle Ergänzung für bestehende forensische Arbeitsabläufe in Frage zu kommen und unter Umständen sogar implementiert zu werden, müssen bestimmte Kriterien erfüllt werden. Diese Kriterien werden zusammen betrachtet als der „Mehrwert“ definiert, den das Triage-Werkzeug Dissect erfüllen muss, um den Test erfolgreich zu bestehen.

Dabei wurden bewusst wenige, in Summe nur zwei Kriterien festgelegt, die gleichermaßen gewichtet werden, jedoch beide vollumfänglich erfüllt werden müssen, um diesen Anforderungen gerecht zu werden. Diese beiden Kriterien wurden als „Korrektheit“ und „Effizienz“ betitelt und sollen im Folgenden ausführlicher dargestellt werden.

4.1.1. Korrektheit

In der digitalen Forensik sind die Genauigkeit und die Integrität der extrahierten Informationen essenziell. Jedes Werkzeug, das in diesem Bereich integriert wird, muss somit eine hohe Zuverlässigkeit aufweisen. Das bedeutet, dass während der Verarbeitungsvorgänge keine Modifikationen oder Verfälschungen an den Originaldaten stattfindet und die gelieferten Ergebnisse der Realität entsprechen.

Bei der Bewertung der Korrektheit müssen die Ergebnisse von Dissect daher mit den manuell erarbeiteten Referenzwahrheiten abgeglichen werden, um sicherzustellen, dass die Ausgaben tatsächlich korrekt und nachvollziehbar sind.

4.1.2. Effizienz

Dieses zunächst als weniger kritisch wirkende Kriterium wird neben der Korrektheit als entscheidender Faktor definiert, der über den Mehrwert bei der Einführung einer Triage-Lösung in Form von Dissect entscheidet. Dabei spielt nicht nur die Geschwindigkeit, mit der die Daten verarbeitet und analysiert werden eine Rolle, sondern auch die Integrationsfähigkeit des Tools in bereits bestehende Prozesse und natürlich deren Benutzerfreundlichkeit.

Die Beurteilung des Mehrwerts, der über die Einführung von Dissect entscheidet, wird sich somit insbesondere darauf konzentrieren, zu prüfen, ob die beiden essenziellen Kriterien Korrektheit und Effizienz in einem forensischen Umfeld erfüllt werden können.

4.2. Vorbereitungen

Zur Durchführung der Tests waren neben der Installation der benötigten Software (siehe Punkt 3.2) noch weitere Schritte nötig, die nachfolgend beschrieben werden.

4.2.1. Disk Images für den Test

Um die Funktionen von Dissect testen zu können, wurden diverse Datenträgersicherungen in Form von Disk-Images mit unterschiedlicher Herkunft und in verschiedenen Formaten organisiert. Um ein möglichst diverses Spektrum an Disk-Images abzudecken, wurden Festplattenimages von folgenden Quellen bezogen:

Microsoft (Windows 11 Entwicklungsumgebungen) [9]

Von Microsoft werden für Entwickler spezielle Entwicklungsumgebungen in diversen Formaten unterschiedlicher Virtualisierungslösungen bereitgestellt. Für die Tests wurden virtuelle Disks in den Formaten VHD und VMDK heruntergeladen.

NIST (CFReDS - Computer Forensic Reference Data Sets) [10]

Das „National Institute of Standards and Technology“ aus den USA stellt über CFReDS „simulierte digitale Beweismittel“ zur Untersuchung zur Verfügung.

Digital Corpora (Computer forensics education research) [11]

Auf der Website von Digital Corpora können Disk Images, Memory Dumps und weiteres für forensische Bildungs- bzw. Forschungszwecke heruntergeladen werden.

4n6.de (Herr Dipl.-Ing. Hans-Peter Merkel) [12]

Im Rahmen des Wings-Fernstudiums an der Hochschule Wismar wurden von Herrn Dipl.-Ing. Hans-Peter Merkel dankenswerterweise Test-Images auf der Plattform 4n6.de bereitgestellt, die insbesondere für die Dissect-Tests im Linux-Bereich eingesetzt werden konnten.

Konkret wurden insgesamt zehn Test-Images für die Versuche heruntergeladen. Die Dateinamen wurden dabei bewusst nicht verändert und entsprechen noch denen, wie sie von der zur Verfügung stellenden Quelle vergeben wurden. Die nachfolgende Tabelle enthält die Basisinformationen inklusive der Herkunft der Disk-Images.

Image-Name	Format	Größe	Quelle
nps-2008-jean.E01	EWF (E01)	2,83 GB	**Digital Corpora**
nps-2009-domexusers.E01	EWF (E01)	4,07 GB	
Test-Image-01.E01	EWF (E01)	XX GB	**www.4n6.de (Zugang nur für Berechtigte)**
Test-Image-02.E01	EWF (E01)	XX GB	
Test-Image-03.E01	EWF (E01)	XX MB	*Dateinamen und Größen für allgemeine Veröffentlichung auf generische Werte geändert bzw. unkenntlich gemacht.*
Test-Image-04.E01	EWF (E01)	XX MB	
Laptop1Final.E01	EWF (E01)	35,5 GB	**NIST (CFReDS)**
20348.169.amd64fre.fe_release_svc_refresh.210806-2348_server_serverdatacentereval_en-us.vhd	VHD (Virtual Hard Disk)	9,5 GB	**Microsoft** (Windows Dev Center)
WinDev2309Eval-disk1.vmdk	VMDK (VMware)	22,9 GB	
SCHARDT.001	RAW (DD)	4,53 GB	**NIST (CFReDS)**
Gesamtdatenmenge der Disk-Images		**91,86 GB**	

Tabelle 3: Basisinformationen der gesammelten Beispiel-Disk-Images

4.2.2. Festlegung der benötigten Triage-Informationen

Als nächsten Vorbereitungsschritt wurde konkret festgelegt, welche Triage-Informationen aus den Disk-Images gewonnen werden sollen. Folgende Informationen wurden dabei als relevant betrachtet:

Betriebssysteminfos	Installationsdatum, Zeitzone, Gerätename, OS-Version
Netzwerk	IPs der Geräte
Benutzer	Eingerichtete User-Accounts auf den Geräten
Aktivität	Informationen zum letzten Verwendungszeitpunkt

4.2.3. Gewinnung von Referenzwahrheiten

Um die das Kriterium der „Korrektheit“ von Dissect testen zu können, war es notwendig, dessen Ergebnisse mit Referenzwahrheiten zu vergleichen. Als Referenzwahrheiten werden die tatsächlich vorliegenden IST-Zustände innerhalb der getesteten Beispiel-Disk-Images bezeichnet. Um diese Referenzwahrheiten festzustellen, wurden die beiden Forensik-programme X-Ways Forensics und Magnet AXIOM verwendet. Dabei wurden alle für den Vergleichstest benötigten Informationen manuell ausgelesen bzw. extrahiert und miteinander abgeglichen. Dieser bedeutsame Teil der Vorbereitungen nahm einen entsprechend großen Zeitaufwand des gesamten Versuchsaufbaus in Anspruch. Am Ende der Gewinnung der Referenzwahrheiten konnte eine Referenztabelle mit allen benötigten Informationen erstellt werden. Die Ergebnisse sind der nachfolgenden Tabelle zu entnehmen.

Image-Name	Format	Betriebssystem	Hostname	Bit	Install.-Datum	Zeitzone	Letze Aktivität	IP-Adressen
nps-2008-jean.E01	EWF (E01)	Microsoft Windows XP SP 3	JEAN-13FBF038A3	32	13.05.2008 23:29	GMT	21.07.2008 03:31	192.168.117.129
nps-2009-domexusers.E01	EWF (E01)	Microsoft Windows XP SP 3	REALISTIC_XP	32	20.10.2008 23:43	PDT	30.10.2008 17:50	192.168.2.129
Test-Image-01.E01	EWF (E01)	Windows 11 Professional (Build 22000)	W11VBOX	64	31.10.2021 11:14	CET	25.11.2021 08:24	10.0.2.15
Test-Image-02.E01	EWF (E01)	Microsoft Windows XP SP 3	XPSP3	32	20.08.2004 00:48	CST	08.08.2014 16:42	192.168.1.12 192.168.2.119
Test-Image-03.E01	EWF (E01)	Ubuntu 16.04 LTS	stream	64	28.08.2017 10:19	CET	28.08.2017 11:16	192.168.2.23
Test-Image-04.E01	EWF (E01)	Ubuntu 14.04.1 LTS	logserver	32	09.07.2018 15:36	CET	09.07.2018 15:52	N/A
Laptop1Final.E01	EWF (E01)	Windows 11 Core (Build 22543)	DESKTOP-SKPTDIO	64	04.02.2022 08:05	EST	13.02.2022 00:37	192.168.191.144 100.64.10.109
20348.169.amd64fre.fe[...].vhd	VHD	Windows Server 2022 Datacenter	MINWINPC	64	--- *	PDT	07.08.2021 02:49	N/A
WinDev2309Eval-disk1.vmdk	VMDK	Windows 11 EnterpriseEval (Build 22621)	WINDEVEVAL	64	20.09.2023 01:22	PDT	20.09.2023 09:14	192.168.118.131 192.168.73.131
SCHARDT.001	RAW (DD)	Microsoft Windows XP (NT 5.1)	N-1A9ODN6ZXK4LQ	32	20.08.2004 00:48	CST	27.08.2004 17:46	192.168.1.111

Tabelle 4: Auflistung der manuell verifizierten Referenzwahrheiten. *Kein Installationsdatum vorhanden.

4.2.3.1. Vorgehen mit X-Ways Forensics

Die Extraktion der Referenzwerte mit X-Ways Forensics wurde mit der Version 20.9 SR-4 der 64-Bit-Variante des Programms durchgeführt.

Die benötigten Informationen der untersuchten Windows-Systeme konnten größtenteils aus der Windows Registry mit dem integrierten Registry Viewer gewonnen werden. Informationen zu Zeitstempeln mit dem Ziel der Feststellung der letzten Verwendung des Systems wurden aus dem Dateisystem gewonnen. Die Vorgehensweise zur Extraktion von Zeitstempeln basiert auf Erkenntnissen, die im Buch „File System Forensic Analysis“ von Brian Carrier beschrieben sind.[13]

Zur Gewinnung der benötigten Infos wurden u.a. folgende Programmkomponenten genutzt:

- Rekursiver Dateiüberblick
- Specialist / Datei-Überblick erweitern
- Schablonenmanager und -ansicht / Falleigenschaften und SID Ansicht
- Registry Viewer
- Hinweis: Linux-Systeminfos müssen mit X-Ways Forensics komplett manuell ausgelesen werden

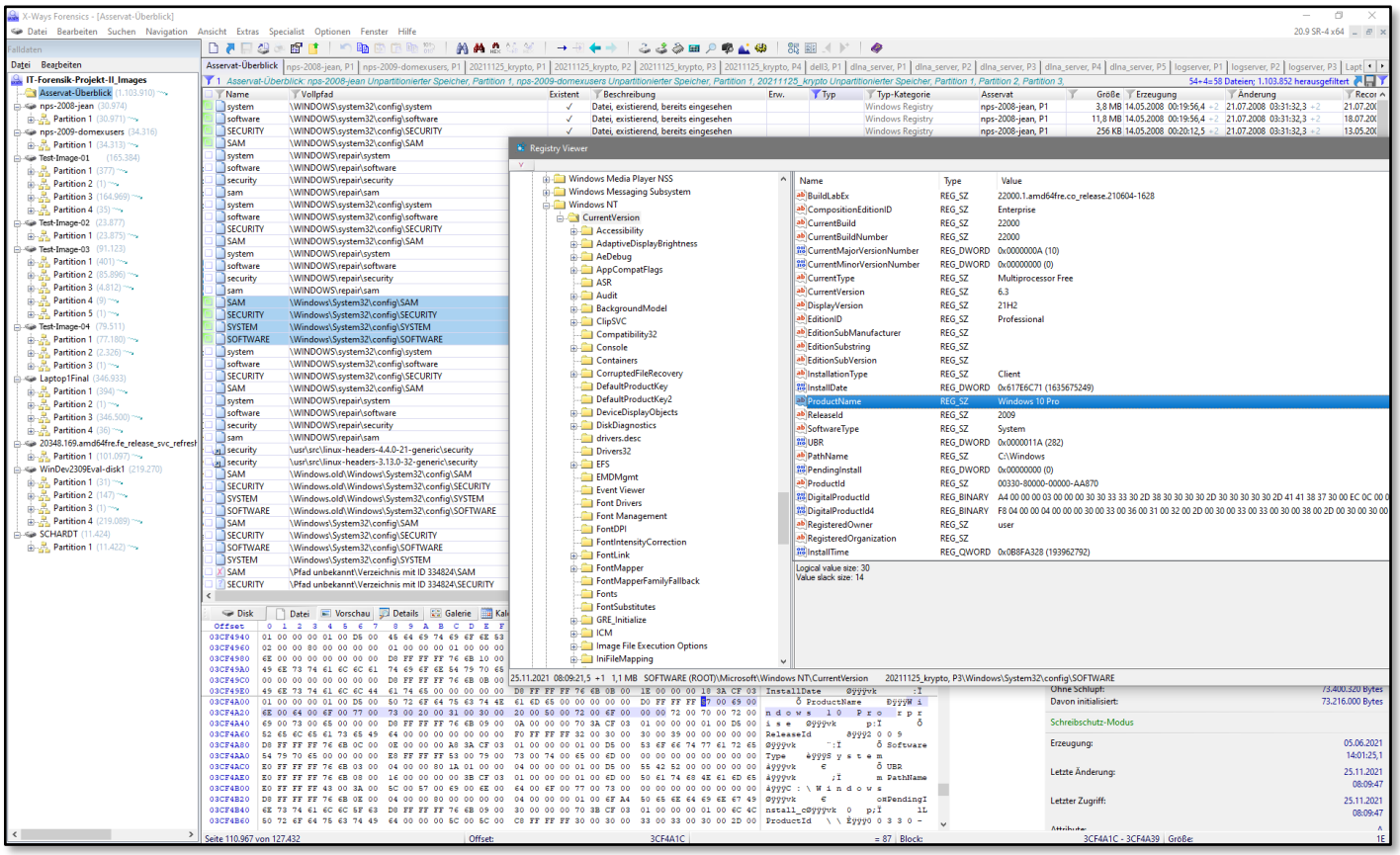


Bild 3: Screenshot von der Gewinnung der Referenzwerte mithilfe von X-Ways Forensics (Registry Viewer)

4.2.3.2. Vorgehen mit Magnet AXIOM

Zum Abgleich der Ergebnisse wurden alle Referenzinformationen ebenfalls mit Magnet AXIOM in der Version 7.5.0.37231 gewonnen. Dabei wurde neben der automatisierten Aufbereitung der Systeminformationen auch der integrierte Viewer für das „Dateisystem“ und die „Registrierung“ genutzt. Für die Gewinnung der benötigten Triage-Informationen waren vor allem die Artefakte der Kategorien Betriebssystem von Interesse.

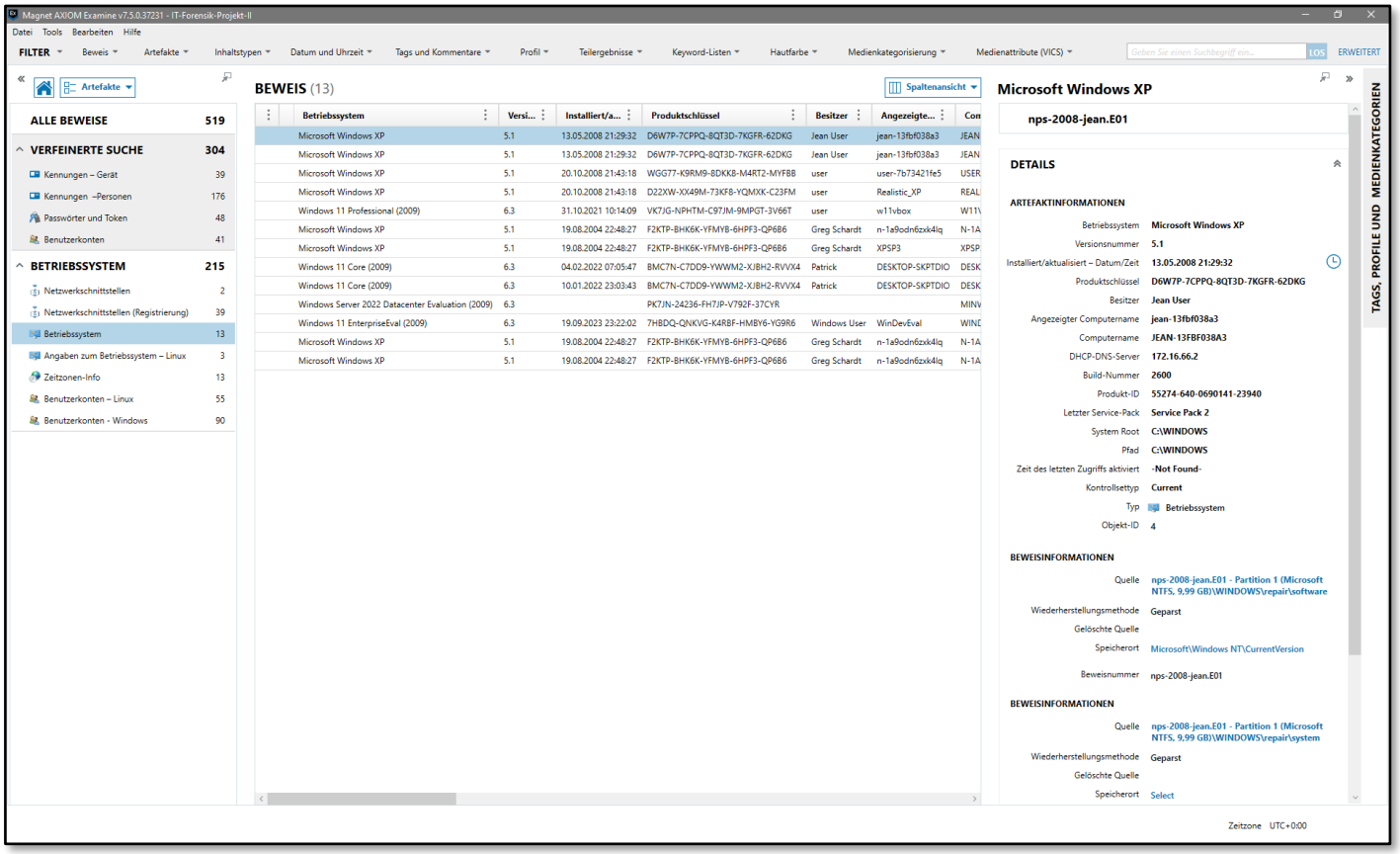


Bild 4: Screenshot aus Magnet AXIOM bei der Gewinnung der Referenzinformationen

4.2.3.3. Einsatz weiterer Tools und Anmerkung

Zur Verifizierung der Einträge der Windows Registry bei den Disk-Images mit installiertem Windows wurden die extrahierten Registry Dateien (SAM, SYSTEM, SOFTWARE, SECURITY) zusätzlich mit dem von Eric Zimmermann entwickeltem Registry Explorer in der Version 2.0.0.0[14] kontrolliert. Erfreulich war hierbei, dass die Ergebnisse aller drei Tools wie zu erwarten tatsächlich identisch waren.

5. Ergebnisse

Mithilfe der folgenden beiden Befehle konnten alle Triage-Informationen von allen zehn getesteten Disk-Images jeweils gleichzeitig extrahiert werden. Die Rückgabewerte von Dissect der OS-Informationen erfolgt als „Text“ und die Rückgabe der Benutzer als „Strings“. Aus diesem Grund wurden zur Bewahrung der Übersichtlichkeit die OS-Informationen und die Benutzer separat abgefragt.

Befehl 1 (Alle benötigten OS-informationen):

```
target-query -q /mnt/i/IT_Forensik_Projekt_II/Test-Images/**/*.{E01,
dd,vhd,vmdk,001} -f os,version,hostname,architecture,install_date,
timezone,activity,ips --cmdb -d ";" >> /mnt/i/IT_Forensik_Projekt_II/
Outputs/TRIAGE-Ergebnisse.csv
```

Die Ausgabe der der Informationen erfolgte mit dem oben genannten Befehl direkt in die Datei „TRIAGE-Ergebnisse.csv“. Durch Ergänzung von `--cmdb -d ";"` wurde der Befehl angewiesen die Formatierung in Form von kommaseparierten Werten zu speichern, genauer gesagt mit dem gewünschten Delimiter „;“ (Semikolon).

TRIAGE-Ergebnisse.csv

```
1  /mnt/i/IT_Forensik_Projekt_II/Test-Images/DigitalCorpora-nps-2008/nps-2008-jean.E01;windows;Microsoft Windows
2  /mnt/i/IT_Forensik_Projekt_II/Test-Images/DigitalCorpora-nps-2009/nps-2009-domexusers.E01;windows;Microsoft W
3  /mnt/i/IT_Forensik_Projekt_II/Test-Images/      Test-Image-01.E01      ;windows;Windows 10 Pro (NT 10.0) 220
4  /mnt/i/IT_Forensik_Projekt_II/Test-Images/  Test-Image-02.E01  ;windows;Microsoft Windows XP (NT 5.1) 2600 Ser
5  /mnt/i/IT_Forensik_Projekt_II/Test-Images/     Test-Image-03.E01     ;linux;Ubuntu 16.04 LTS;stream;x86_64-lin
6  /mnt/i/IT_Forensik_Projekt_II/Test-Images/    Test-Image-04.E01    ;linux;Ubuntu 14.04.1 LTS;logserver;x86_32-
7  /mnt/i/IT_Forensik_Projekt_II/Test-Images/Magnet Images/Laptop1Final.E01;windows;Windows 10 Home (NT 10.0) 22
8  /mnt/i/IT_Forensik_Projekt_II/Test-Images/Microsoft/20348.169.amd64fre.fe_release_svc_refresh.210806-2348_ser
9  /mnt/i/IT_Forensik_Projekt_II/Test-Images/Microsoft/WinDev2309Eval-disk1.vmdk;windows;Windows 10 Enterprise E
10 /mnt/i/IT_Forensik_Projekt_II/Test-Images/NIST-Hacking-Case/SCHARDT.001;windows;Microsoft Windows XP (NT 5.1)
```

Bild 5: Auszug aus der resultierenden CSV-Datei mit der gewünschten Separierung (Semikolon)
(Image-Namen in Zeilen 3-6 nachträglich durch generische Namen ersetzt)

Befehl 2 (Benutzerinformationen):

```
target-query -q /mnt/i/IT_Forensik_Projekt_II/Test-Images/**/*.{E01,
dd,vhd,vmdk,001} -f users
```

Im Fall der Benutzerinformationen werden die Ergebnisse mit dem oben abgebildeten Befehl direkt auf der Kommandozeile ausgegeben.

```
wingsfz@ENGINE:~$ target-query -q /mnt/i/IT_Forensik_Projekt_II/Test-Images/**/*.{E01,dd,vhd,vmdk,001} -f users
<windows/user hostname='JEAN-13FBF038A3' domain=None sid='S-1-5-18' name='systemprofile' home='%systemroot%\\system32\\config\\systemprofile'>
<windows/user hostname='JEAN-13FBF038A3' domain=None sid='S-1-5-19' name='LocalService' home='%SystemDrive%\\Documents and Settings\\LocalService'>
<windows/user hostname='JEAN-13FBF038A3' domain=None sid='S-1-5-20' name='NetworkService' home='%SystemDrive%\\Documents and Settings\\NetworkService'>
<windows/user hostname='JEAN-13FBF038A3' domain=None sid='S-1-5-21-484763869-796845957-839522115-1004' name='Jean' home='%SystemDrive%\\Documents and Settings\\Jean'>
```

Bild 6: Auszug aus der Linux-Shell mit den Benutzerinformationen und den dazugehörigen Infos

Die resultierenden Ergebnisse daraus werden nachfolgend bewertet.

5.1. Überprüfung auf Korrektheit

Die mit den unter Punkt 5 dargestellten Befehlen extrahierten Informationen wurden in dasselbe Tabellenmuster wie die Referenzwerte eingetragen, um einen Vergleich durchführen zu können. Das Resultat ist nachfolgender Abbildung zu entnehmen.

Image-Name	Format	Betriebssystem	Hostname	Bit	Install.-Datum	Zeitzone	Letze Aktivität	IP-Adressen
nps-2008-jean.E01	EWF (E01)	Microsoft Windows XP SP 3	JEAN-13FBF038A3	32	13.05.2008 23:29	GMT - Europe/London	21.07.2008 03:31	192.168.117.129
nps-2009-domexusers.E01	EWF (E01)	Microsoft Windows XP SP 3	REALISTIC_XP	32	20.10.2008 23:43	PDT -America/Los_Angeles	30.10.2008 17:50	192.168.2.129
Test-Image-01.E01	EWF (E01)	Windows 10 Pro (NT 10.0) 22000.282	W11VBOX	64	31.10.2021 11:14	CET - Europe/Berlin	25.11.2021 08:24	10.0.2.15
Test-Image-02.E01	EWF (E01)	Microsoft Windows XP SP 3	XPSP3	32	20.08.2004 00:48	CST - America/Chicago	08.08.2014 16:42	192.168.1.12 192.168.2.119
Test-Image-03.E01	EWF (E01)	Ubuntu 16.04 LTS	stream	64	28.08.2017 10:19	CET - Europe/Berlin	28.08.2017 11:16	192.168.2.23
Test-Image-04.E01	EWF (E01)	Ubuntu 14.04.1 LTS	logserver	32	09.07.2018 15:36	CET - Europe/Berlin	09.07.2018 15:52	N/A
Laptop1Final.E01	EWF (E01)	Windows 10 Home (NT 10.0) 22543.1000	DESKTOP-SKPTDIO	64	04.02.2022 08:05	EST - America/New_York	13.02.2022 00:37	192.168.191.144 100.64.10.109
20348.169.amd64fre.fe[...].vhd	VHD	Windows Server 2022 Datacenter	MINWINPC	64	01.01.1970 01:00	PDT -America/Los_Angeles	07.08.2021 02:49	N/A
WinDev2309Eval-disk1.vmdk	VMDK	Windows 10 Enterprise Evaluation (NT 10.0) 22621.2283	WINDEVEVAL	64	20.09.2023 01:22	PDT -America/Los_Angeles	20.09.2023 09:14	192.168.118.131 192.168.73.131
SCHARDT.001	RAW (DD)	Microsoft Windows XP (NT 5.1)	N-1A9ODN6ZXK4LQ	32	20.08.2004 00:48	CST - America/Chicago	27.08.2004 17:46	192.168.1.111

Tabelle 5: Auflistung der von Dissect festgestellten Werte. Abweichungen wurden farblich markiert

Die mit der Referenztabelle identischen Werte wurden mit grüner Farbe hervorgehoben. Diskrepanzen sind hellrot gekennzeichnet und ein Tabellenwert, der mittels manueller Analyse nicht ausgelesen werden konnte, aber von Dissect dennoch befüllt wurde, wurde gelb markiert.

Erfreulich ist zunächst, dass ein Großteil der Werte offensichtlich korrekt ausgelesen wurde. Da das Kriterium „Korrektheit“ von Dissect aber zu 100 Prozent erfüllt werden muss, um als Triage-Werkzeug eingesetzt werden zu können, werden die vermeintlich „falsch“ ausgelesenen Werte nachfolgend noch einmal näher betrachtet.

5.1.1. Sonderfall: Zeitstempel Installationsdatum

Mithilfe der manuellen Artefaktanalyse konnte vom Disk-Image des Hosts „MINWINPC“ (Windows Server 2022 Datacenter) kein Installationszeitpunkt ausgelesen werden. Der für diese Info zuständige Registry-Wert war im Disk-Image lediglich mit einer „0“ befüllt. Dieser Wert wurde in der Referenz daher logischerweise manuell als nicht gültig und somit nicht vorhanden bewertet. Der Wert befindet sich im SOFTWARE-Registry Schlüssel unter folgender Position:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\InstallDate
```

Dissect kommt beim Auslesen des Installationsdatums jedoch auf den 01.01.1970 um 01:00 Uhr.

Dies begründet sich daher, dass der entsprechende Registrierungseintrag im UNIX-Zeitstempel-Format abgespeichert wird. Das ist ein 32-Bit-Wert, der die Zeit als Sekunden seit dem 1.1.1970 speichert. [15] Folglich müsste der Wert eigentlich also 1.1.1970 00:00 Uhr lauten. In Kombination mit der auf dem Image abgelegten Zeitzone (PDT – Pacific Daylight Time) kommt dabei jedoch ein Wert vom 1.1.1970 01:00 Uhr heraus, wie er von Dissect entsprechend ausgefüllt wurde. Aus technischer Sicht muss somit festgestellt werden, dass Dissect den vorhandenen Wert (0) korrekt interpretiert und ausgegeben hat.

Für die Bewertung des Kriteriums „Korrektheit“ wird das Ergebnis somit auch als erfüllt betrachtet.

5.1.2. Sonderfall: Windows 11 Problematik

Ebenfalls aufgrund der Windows Registry stellen sich Diskrepanzen der festgestellten Betriebssysteme dar. In der Referenztabelle ist ersichtlich, dass sich unter den getesteten Disk-Images drei Windows 11 Installationen unterschiedlicher Versionen befinden. Dissect gab bei diesen drei Images jedoch stets Windows 10 aus – zum Teil mit Build-Nummern die es für Windows 10 nie gegeben hat. [16]

Bei der Gewinnung der Referenzinformationen musste bereits festgestellt werden, dass diese Systeme in deren Registrierungsdatenbank, die diese Information bereithält, bereits fälschlicherweise den Betriebssystemnamen („Product Name“) als Windows 10 darstellen.

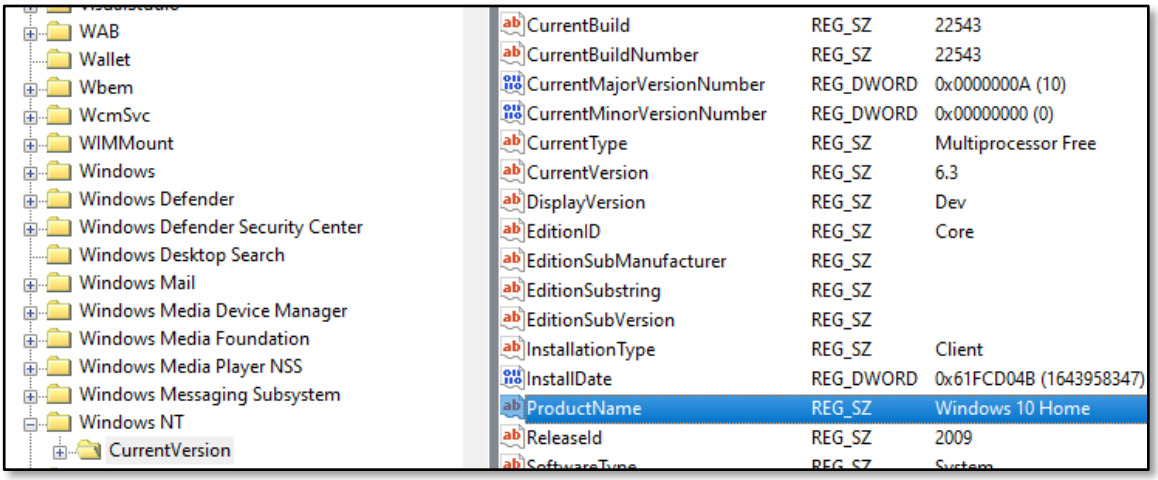


Bild 7: Falscher Produktname in der Registry bei allen getesteten Windows 11 Installationen

Einige Betriebssysteminformationen konnten, bis Windows 10, zuverlässig aus Folgendem SOFTWARE-Registry-Pfad entnommen werden:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion
```

Bei Internet-Recherchen, nach dem Grund weshalb dieser Wert nicht mehr korrekt eingetragen wird, konnte in Erfahrung gebracht werden, dass dies ein „zu erwartendes Verhalten“ von allen Windows 11 Varianten ist, um die Kompatibilität zu älterer Software nicht zu gefährden.[17]

Ein offizieller Artikel von Microsoft konnte diesbezüglich im Rahmen der Recherchen nicht gefunden werden. Im Microsoft eigenen Support-Forum (learn.microsoft.com) meldeten sich jedoch einige als Microsoft Mitarbeiter gekennzeichnete Benutzer zu Wort, die dies bestätigen.

Stellvertretend hierfür wird eine der gefundenen Antworten von Jason Sandys, Senior Architect Product Manager bei Microsoft abgebildet, der auf den Vorschlag eines Forenteilnehmers ablehnend reagiert, dass Microsoft die Build-Nummer von Windows 11 entsprechend anpasst.

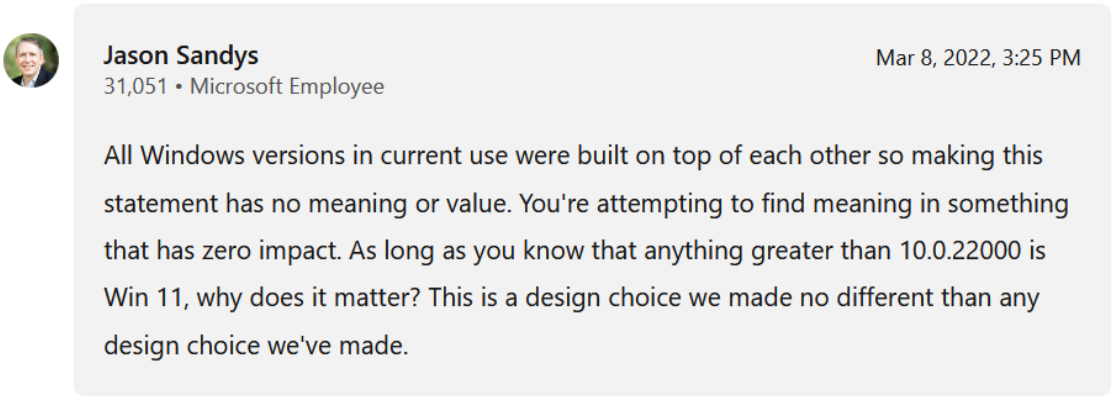


Bild 8: Microsoft Mitarbeiter zum Thema Windows 11 in der Registry in Antwort auf eine Nutzerfrage [18]

Der Konsens ist somit lediglich, dass wenn die Windows Build-Nummer 10.0.22000 oder höher ist, es sich um Windows 11 handelt. Aktuelle Windows 10 Build-Nummern der Version 22H2 starten alle mit „10.0.19...“.

Der Ausgabewert, den Dissect bei der Abfrage nach der Betriebssystem-Version liefert beschränkt sich jedoch erfreulicherweise nicht nur auf den Registry-Wert „Product Name“, sondern liefert ebenfalls den Wert „CurrentBuild“ mit, der die korrekte Build-Nummer darstellt.

Somit kann durch aufmerksame Betrachtung des Rückgabewerts der OS-Version korrekterweise entnommen werden, dass bei den entsprechenden Disk-Images Windows 11 installiert ist, da alle ausgegebenen Build-Nummern über 10.0.220000 liegen.

Dies bedeutet, dass Dissect den Wert nicht falsch ausliest, sondern den von Microsoft Windows selbst in die Registry eingetragenen falschen Wert korrekt zurückgibt.

Zum Vergleich gibt X-Ways Forensics beim Aufruf der Beweismitteleigenschaften ebenfalls „korrekt“ den „falschen Namen“ für das Betriebssystem an. Auf dem Disk-Image befindet sich in Wirklichkeit eine Installation von „Windows 11 Core“.

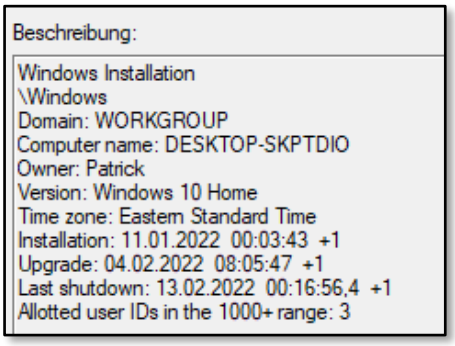


Bild 9: Beweismittelbeschreibung in X-Ways Forensics

Nur das Forensikprogramm Magnet AXIOM nennt das Betriebssystem beim richtigen Namen. Auf welches Artefakt das Programm für diese Information zugreift ist jedoch nicht dokumentiert. Vermutlich wird hierbei ebenfalls die korrekt eingetragene Build-Nummer berücksichtigt und dann mit einer hinterlegten Namensreferenz abgeglichen.

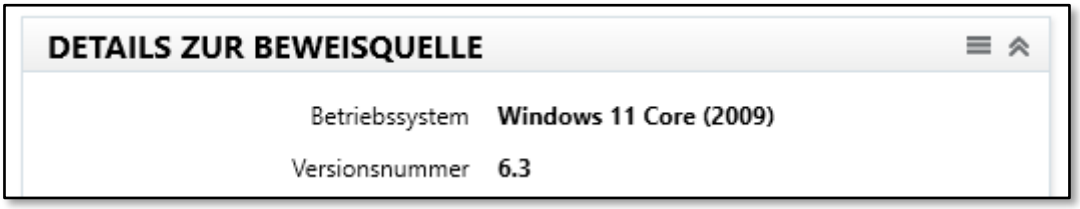


Bild 10: Beweisquelleninformation in Magnet AXIOM (OS-Name korrekt)

5.1.3. Kriterium „Korrektheit“ erfüllt

Obgleich es zwei Sonderfälle hinsichtlich des Installationsdatums und der Betriebssystembezeichnung gibt, wurden die gesuchten Werte von Dissect durchweg korrekt extrahiert.

Das Kriterium der „Korrektheit“ wird somit als erfüllt angesehen.

In Anbetracht dieser Grundvoraussetzung wurde im Weiteren auch der Mehrwert in Bezug auf die Effizienz evaluiert.

5.2. Überprüfung auf Effizienz

Um die Effizienz von Dissect gegenüber den bestehenden Forensikprogrammen X-Ways Forensics und Magnet AXIOM zu vergleichen, wurden mehrere Zeitmessungen durchgeführt, um festzustellen, wie lange die Gewinnung der benötigten Triage-Informationen dauert.

Nur wenn die Zeit der Informationsgewinnung durch Dissect signifikant geringer ist als die Vorgehensweise mit den beiden Forensikprogrammen kann hier von einem Erfolg ausgegangen werden. Als signifikant wird ein Geschwindigkeitsvorteil festgelegt, der nicht mehr als die Hälfte der benötigten Zeit im Vergleich zu den vorhandenen Programmen in Anspruch nimmt. Beim Testen wurde darauf geachtet, lediglich die minimalst benötigten Schritte auszuführen, die für die Feststellung der benötigten Triage-Informationen notwendig sind.

Für die Informationsgewinnung wurden immer alle zehn Disk-Images in einem Verarbeitungsschritt gleichzeitig verarbeitet, um das Szenario eines größeren Falls mit zehn unterschiedlichen Asservaten nachzubilden.

Pro Werkzeug wurden die Tests jeweils fünfmal durchgeführt, um evtl. Messungenauigkeiten zu minimieren. Zum Vergleich wird der daraus berechnete Durchschnittswert herangezogen. Vor jedem Test wurde die Testumgebung neu gestartet und alle Cache-Verzeichnisse gelöscht, um die Ergebnisse nicht zu verfälschen.

Die Zeitmessung von X-Ways Forensics sowie Magnet AXIOM fanden durch Zuhilfenahme der Online-Stoppuhr des Anbieters „Time and Date AS“ aus Norwegen[19] statt. Die Nutzung der Online-Stoppuhr wurde stichprobenartig auch mit einer physischen Stoppuhr kontrolliert. Die Ergebnisse wiesen keine Diskrepanzen auf.

Die Ausführungszeiten der Dissect-Befehle wurden über das Linux-Kommando „time“ direkt gemessen.

Sollten die Zeitmessungen keine eindeutigen Ergebnisse liefern wird die Effizienz mithilfe weiterer Bewertungskriterien geprüft. Mögliche Kriterien wären hierbei die Bedienbarkeit, die Ressourcennutzung, usw.

5.2.1. X-Ways Forensics - Geschwindigkeitstest

Die Geschwindigkeitstests wurden mit X-Ways Forensics Version 20.9 SR-4 (64 Bit) durchgeführt.

X-Ways Forensics	Test 1	Test 2	Test 3	Test 4	Test 5	Mittel
Fallprojekt erstellen	00 min 00 s	00 min 00 s	00 min 00 s	00 min 00 s	00 min 00 s	**00 min 00 s**
Images hinzufügen	01 min 16 s	01 min 01 s	01 min 19 s	01 min 18 s	00 min 56 s	**01 min 10 s**
Rekursiven Dateiüberblick erzeugen	01 min 12 s	01 min 16 s	01 min 11 s	01 min 18 s	01 min 23 s	**01 min 16 s**
OS-Informationen aufrufen	03 min 16 s	03 min 09 s	03 min 03 s	02 min 56 s	03 min 01 s	**03 min 05 s**
Benutzerkonten (SIDs) anzeigen	00 min 10 s	00 min 09 s	00 min 08 s	00 min 07 s	00 min 08 s	**00 min 08 s**
Letzte Aktivitäten prüfen (Erzeugungszeitstempel)	01 min 38 s	01 min 31 s	01 min 28 s	01 min 29 s	01 min 24 s	**01 min 30 s**
Gesamt	07 min 32 s	07 min 06 s	07 min 09 s	07 min 08 s	06 min 52 s	**07 min 09 s**
Speicherplatz für X-Ways Forensics Projektdatei:					**248 MB**	

Tabelle 6: Ergebnisse der Geschwindigkeitstests von X-Ways Forensics

Die Durchschnittliche Geschwindigkeit zur Extraktion der benötigten Triage-Informationen betrug 7 Minuten und 9 Sekunden. Die Zeit für die Erstellung der Projektdatei wurde dabei nicht berücksichtigt, da diese in der Praxis keine Rolle spielt.

Es wird darauf hingewiesen, dass die in der Tabelle aufgezeigten Werte nur die Zeit umfassen, wie lange es dauert, die Informationen abzurufen. Eine etwaige Dokumentation ist dabei noch nicht mit eingerechnet.

Darüber hinaus muss noch erwähnt werden, dass folgende Informationen von X-Ways Forensics nicht automatisch erfasst werden und manuell aufbereitet werden müssen:

- IP-Adressen der Systeme
- Sämtliche Informationen bzgl. Linux

Die manuelle Erfassung der o.g. Informationen sowie die Dokumentation würden die gemessene Zeit nochmals erhöhen.

5.2.2. Magnet AXIOM - Geschwindigkeitstest

Die Geschwindigkeitstests wurden mit Magnet AXIOM Version 7.5.0.37231 durchgeführt.

Magnet AXIOM	Test 1	Test 2	Test 3	Test 4	Test 5	Mittel
Fallkonfiguration inklusive Hinzufügen der Disk-Images	08 min 51 s	08 min 44 s	08 min 23 s	08 min 16 s	08 min 18 s	**08 min 30 s**
Beweisanalyseprozess (Minimalkonfiguration)	01 h 08 min 38 s	01 h 16 min 10 s	01 h 14 min 18 s	01 h 02 min 19 s	01 h 15 min 45 s	**01 h 11 min 26 s**
Gesamt	01 h 17 min 29 s	01 h 24 min 54 s	01 h 22 min 41 s	01 h 10 min 35 s	01 h 24 min 03 s	**01 h 19 min 56 s**
Speicherplatz für Magnet AXIOM Projektdatei:					**3,13 GB**	

Tabelle 7: Ergebnisse der Geschwindigkeitstests von Magnet AXIOM

Die Verarbeitungszeit aller zehn Disk-Images betrug insgesamt 1 Stunde 19 Minuten und 56 Sekunden. Bevor mit den Artefakten in Magnet AXIOM interagiert werden kann, werden diese vom Programmbestandteil „Process“ aufbereitet.

Die Aufbereitung richtet sich dabei nach der gewählten Konfiguration. Hierbei wurde die minimal benötigte Konfiguration gewählt, um keine ungewollte Benachteiligung zu provozieren. Ein Carving von Informationen aus gelöschten Bereichen wurde ebenfalls deaktiviert.

Um die Referenzwerte für Abschnitt 4.2.3.2 zu ermitteln, wurden die Images mittels Magnet AXIOM Process bereits umfassend und unter Aktivierung aller Artefaktoptionen verarbeitet. Daraus wurden die entsprechenden Artefaktquellen der Ergebnisse notiert. Für die Geschwindigkeitstests in Magnet AXIOM waren somit lediglich die vier benötigten Artefaktquellen aktiv.

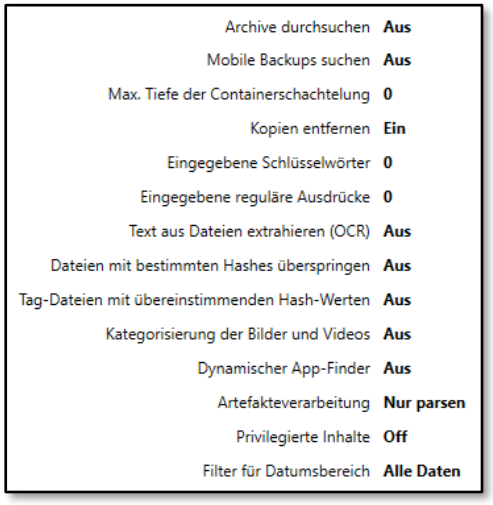


Bild 11: Screenshot der Konfigurationszusammenfassung aus „Magnet AXIOM Process“

Die Verarbeitungszeit von Magnet AXIOM ist im Vergleich zu X-Ways Forensics deutlich länger. Es muss jedoch erwähnt werden, dass dabei alle benötigten Triage-Informationen vollumfänglich erfasst wurden, sowohl von den Windows- als auch den Linux-Systemen. Das abschließende Zusammenfassen der Informationen war nicht Bestandteil der Betrachtung.

5.2.3. Dissect - Geschwindigkeitstest

Die Geschwindigkeitstests wurden mit Dissect in der Version 3.9 ausgeführt.

Dissect	Test 1	Test 2	Test 3	Test 4	Test 5	Mittel
Dissect-Befehl eingeben	00 min 34 s	00 min 31 s	00 min 26 s	00 min 30 s	00 min 29 s	**00 min 30 s**
Betriebssysteminfos in CSV-Datei speichern	00 min 27 s	00 min 26 s	00 min 24 s	00 min 30 s	00 min 23 s	**00 min 26 s**
Benutzerinformationen ausgeben	00 min 05 s	00 min 06 s	00 min 06 s	00 min 05 s	00 min 05 s	**00 min 05 s**
Gesamt	01 min 06 s	01 min 03 s	00 min 56 s	01 min 05 s	00 min 57 s	**01 min 01 s**
Größe der resultierenden CSV-Datei:					**2,45 KB**	

Tabelle 8: Ergebnisse der Geschwindigkeitstests von Dissect

Dissect liefert die gewünschten Ergebnisse von allen zehn Disk-Images durchschnittlich in einer Minute zurück. Die Betriebssysteminformationen werden dabei sogar in einer CSV-Datei abgelegt, die direkt weiterverarbeitet werden kann.

Hinweis: Die Benutzerinformationen können nicht mit in die CSV-Datei geschrieben werden und müssen von der Kommandozeile aus herauskopiert werden.

5.3. Erfüllung des Mehrwertes

Die Tests konnten aufzeigen, dass Dissect die Ergebnisse nicht nur korrekt, sondern auch effizient, in einer deutlich schnelleren Zeit als die bestehenden Forensikprogramme extrahieren kann.

Durch die ausgesprochen deutlich höhere Effizienz von Dissect mussten mögliche weitere Bewertungsaspekte wie die „Einfachheit der Bedienung“ und der „geringe Ressourcenverbrauch“ für die Erkennung des Mehrwertes nicht mehr berücksichtigt werden.

Folgende Tabelle zeigt nochmal die durchschnittlichen Testzeiten im Vergleich:

X-Ways Forensics	Magnet Axiom	Dissect
00 h 07 min 09 s	01 h 19 min 56 s	00 h 01 min 01 s

Tabelle 9: Vergleich der Gesamtverarbeitungszeiten

Der als Zeitersparnis festgelegte Mehrwert mit einer Differenz von „der Hälfte der Zeit“ (folglich also 50% Zeitersparnis) wurde im Vergleich zu beiden Forensikprogrammen deutlich erreicht.

In Zahlen ist Dissect um den Faktor 7,03 schneller als X-Ways Forensics, benötigt also lediglich 14,22 % der Zeit. Im Vergleich zu Magnet AXIOM ist Dissect um den Faktor 78,62 schneller und benötigt für die Extraktion der Informationen lediglich 1,27 % der Zeit.

Die Kriterien „Korrektheit“ sowie „Effizienz“ konnten durch die ausgeführten Tests eindeutig festgestellt werden. Somit kann Dissect als geeignetes Triage-Werkzeug mit signifikantem Mehrwert für den Bereich der IT-Forensik betrachtet und als solches in bestehende Workflows integriert werden.

6. Zusammenfassung und Ausblick

Diese Projektarbeit zielte darauf ab, Möglichkeiten einer effektiven Triage für Disk-Images im forensischen Umfeld zu beleuchten, um eine effiziente Lösung zur Bewältigung der wachsenden Anzahl von Asservaten zu finden.

Der Fokus lag hierbei auf der Identifizierung eines geeigneten Programms, das den spezifisch definierten Anforderungen entspricht und als nützliche Ergänzung in bestehende forensische Workflows integriert werden kann.

Nach einer umfassenden Marktschau und Prüfung der verfügbaren Optionen, stellte sich das Tool "Dissect" als vielversprechende Option heraus. Als kostenfreies Open Source Programm bietet es die Möglichkeit, nahtlos in bestehende Arbeitsabläufe integriert zu werden.

Durch die Evaluation von Dissect anhand festgelegter Testkriterien konnte das Tool seine Korrektheit und Effizienz unter Beweis stellen und dabei sämtliche festgelegten Anforderungen erfüllen, wodurch es als wertvolles Werkzeug für IT-Forensik-Labore eingestuft werden kann.

Das Tool bietet das Potential, zeitnah in den forensischen Arbeitsalltag integriert zu werden. Die Open Source – Lizenz des Tools spielt dabei eine wichtige Rolle bei der Implementierung und der Möglichkeit das Tool auf eigene Bedürfnisse anzupassen.

Durch die Modularität von Dissect besteht zudem das Potential die forensische Gemeinschaft global zu fördern und lädt diese dazu ein, eigene Ideen, Plugins und Module dafür beizutragen.

Durch die intensive manuelle Beschäftigung mit den forensischen Artefakten, die für eine Disk-Triage notwendig sind, wurde zudem wieder bewusst vor Augen geführt, dass es nicht sinnvoll ist, sich uneingeschränkt auf die Ergebnisse von etablierten Forensikprogrammen zu verlassen. Es lässt sich abschließend betonen, dass im Bereich der IT-Forensik ständige Achtsamkeit gegeben sein sollte, um feststellen zu können, ob die Ergebnisse der eingesetzten Werkzeuge wirklich den Tatsachen entsprechen. Dies gilt insbesondere bei Versionsupdates von Programmen und Betriebssystemen wie die beschriebene Windows 11 Problematik deutlich veranschaulicht hat.

Literaturverzeichnis

[1] *BSI Leitfaden IT-Forensik Version 1.0.1*. (o. D.). Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf; Letzter Abruf am 18.09.2023

[2] *Open Source Software und Vorabversionen von Betriebssystemen*. (o. D.). Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cyber-Sicherheitsempfehlungen/Updates-Browser-Open-Source-Software/Open-Source-Vorabversionen-von-Betriebssystemen/open-source-vorabversionen-von-betriebssystemen_node.html; Letzter Abruf am 18.09.2023

[3] Bencherchali, N. (2019, 15. September). Windows Forensics Analysis — Windows Artifacts (Part I). *Medium*. https://nasbench.medium.com/windows-forensics-analysis-windows-artifacts-part-i-c7ad81ada16c; Letzter Abruf am 20.09.2023

[4] SECUINFRA Falcon Team (2021, 22. Juli). Digitale Forensik - relevante Artefakte für eine forensische Analyse. *SECUINFRA*. https://www.secuinfra.com/de/techtalk/digitale-forensik-triage/; Letzter Abruf am 20.09.2023

[5] Fux, C. & Tiefenböck, F. (2022, 4. Januar). *Triage*. NetDoktor. https://www.netdoktor.de/diagnostik/triage/; Letzter Abruf am 20.09.2023

[6] Lennon, M. (2015, 25. November). NCC Group pays $142 million to acquire Fox-IT. *SecurityWeek*. https://www.securityweek.com/ncc-group-pays-142-million-acquire-fox-it/; Letzter Abruf am 21.09.2023

[7] Fox-It. (o. D.). *GitHub - Fox-IT/Dissect.* GitHub. https://github.com/fox-it/dissect; Letzter Abruf am 08.10.2023

[8] https://docs.dissect.tools/en/latest/overview/index.html; Letzter Abruf am 08.10.2023

[9] *Microsoft (o. D.). Herunterladen eines virtuellen Windows-Computers – Entwicklung von Windows-Apps*. Microsoft Developer. https://developer.microsoft.com/de-de/windows/downloads/virtual-machines/; Letzter Abruf am 06.10.2023

[10] CFREDS Portal. https://cfreds.nist.gov/; Letzter Abruf am 01.10.2023

[11] Digital Corpora. https://digitalcorpora.org/; Letzter Abruf am 02.10.2023

[12] Merkel, H.-P. Forensik Portal, https://www.4n6.de/; Letzter Abruf am 04.10.2023

[13] Carrier, B. (2005). *File System Forensic analysis*. Addison-Wesley Professional.

[14] *Eric Zimmerman's Tools*. (o. D.). https://ericzimmerman.github.io/#!index.md; Letzer Abrufm am 07.10.2023

[15] Klein, H. (2009, 24. September). *How to determine the Windows installation date with and without PowerShell*. sepago. https://www.sepago.de/blog/how-to-determine-the-windows-installation-date-with-and-without-powershell/; Letzer Abruf am 05.10.2023

[16] Microsoft. (2023, 11. Juli). *Windows 10 - Release Information*. Microsoft Learn. https://learn.microsoft.com/en-us/windows/release-health/release-information; Letzter Abruf am 06.10.2023

[17] Windows 11 Product Name in Registry - Microsoft Q&A. (o. D.). https://learn.microsoft.com/en-us/answers/questions/555857/windows-11-product-name-in-registry?page=2; Letzter Abruf am 07.10.2023

[18] *Windows 11 Build Ver is still 10.0.22000.194 - Microsoft Q&A*. (o. D.). https://learn.microsoft.com/en-us/answers/questions/586619/windows-11-build-ver-is-still-10-0-22000-194; Letzter Abruf am 07.10.2023

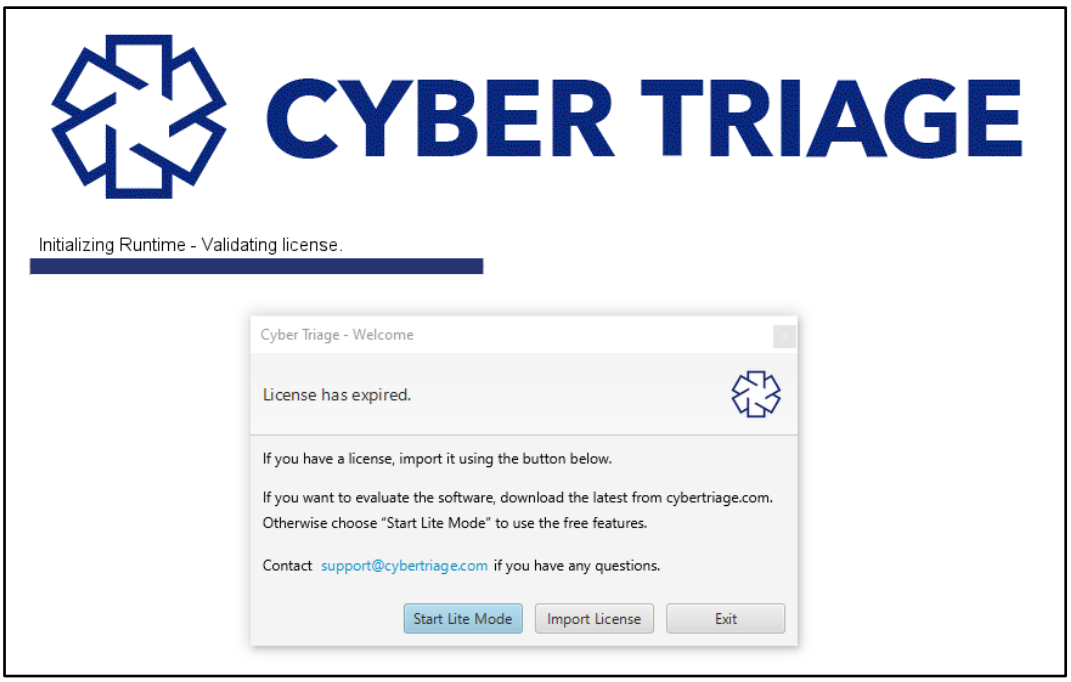
[19] *Online Stopwatch - easy to use*. (o. D.). https://www.timeanddate.com/stopwatch; Letzter Abruf am 06.10.2023

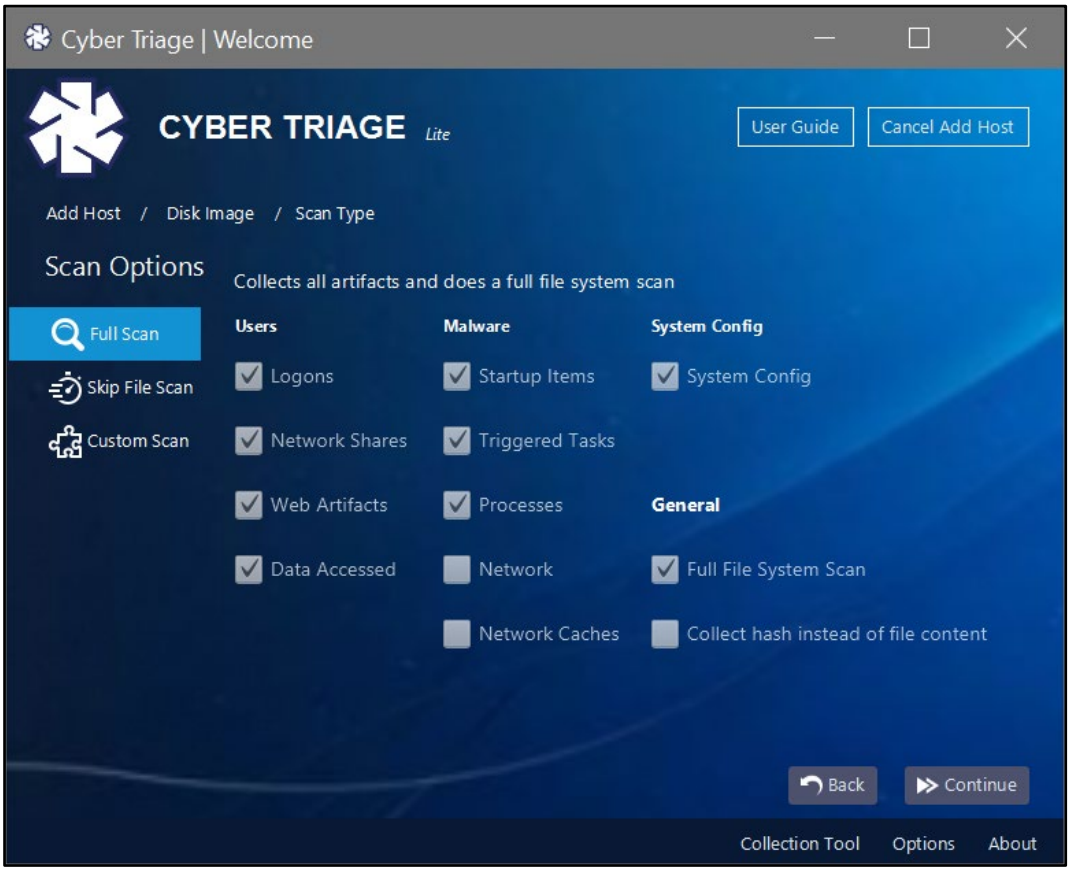
Bilderverzeichnis

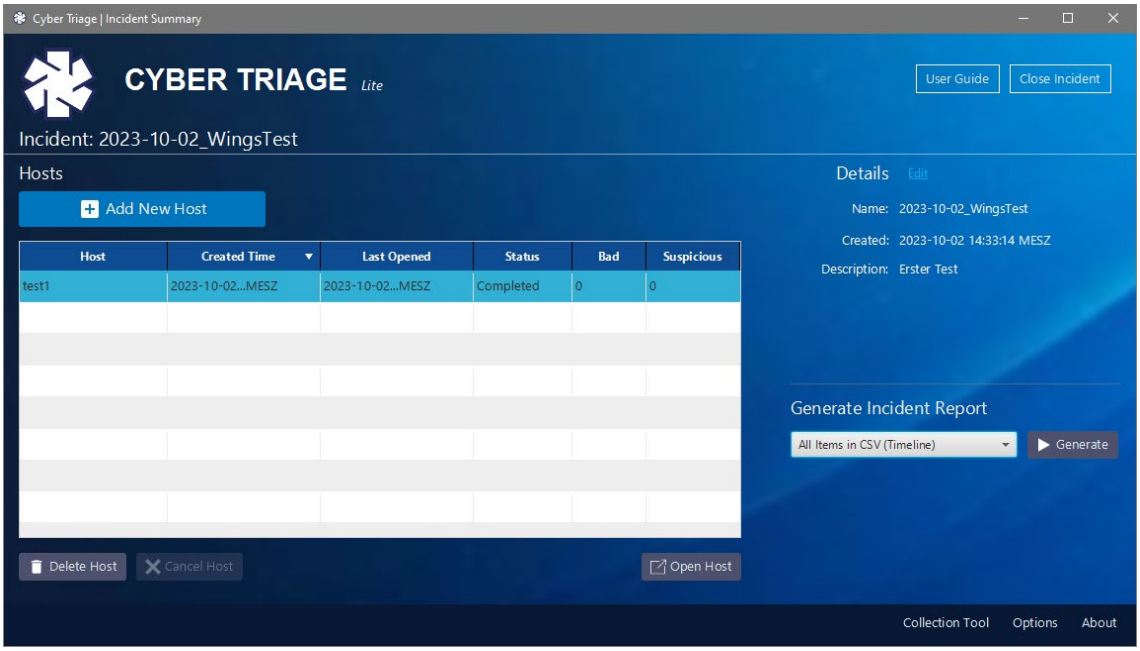
Tabellenverzeichnis

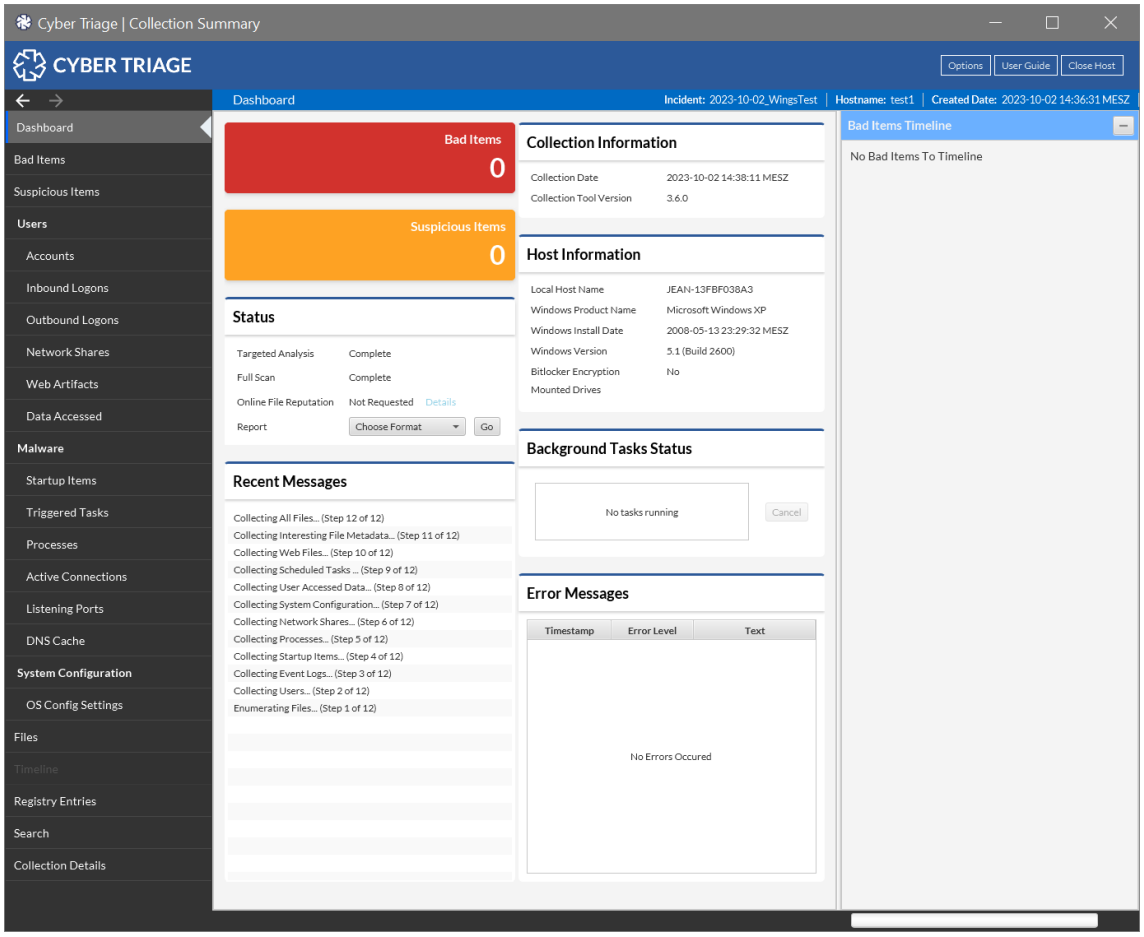
Anhänge

A.1 Screenshots zu Cyber Triage

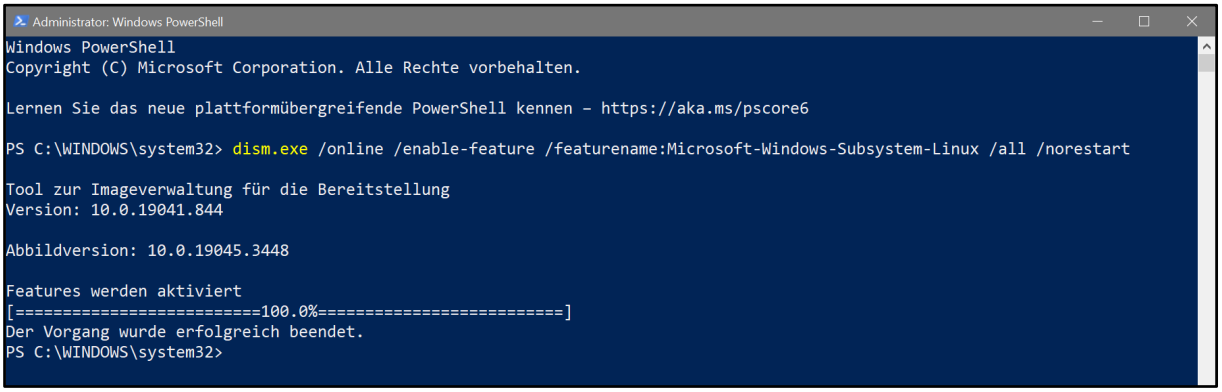


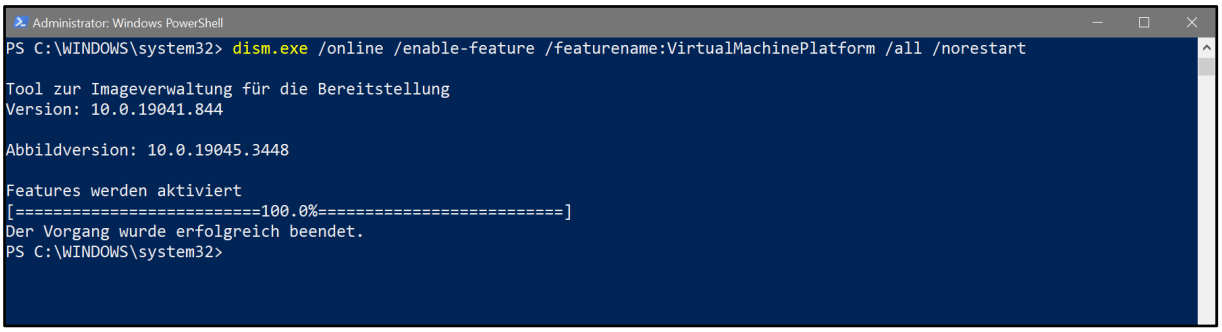


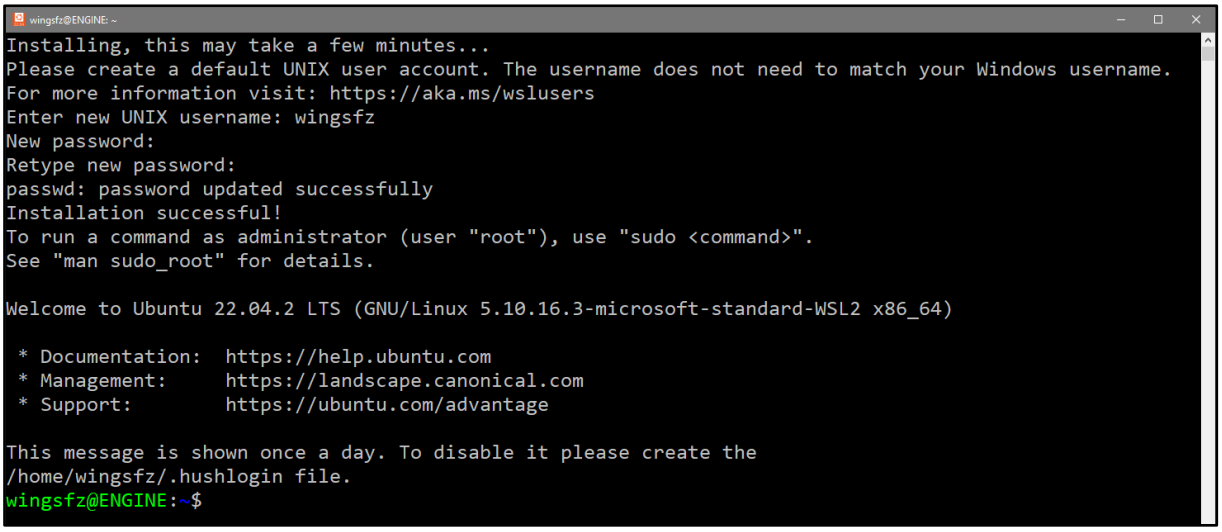

Cyber Triage | Incident Summary
CYBER TRIAGE Lite
User Guide
Close Incident
Incident: 2023-10-02_WingsTest
Hosts
Add New Host
Host
Created Time
Last Opened
Status
Bad
Suspicious
test1
2023-10-02...MESZ
2023-10-02...MESZ
Completed
0
0
Details
Edit
Name: 2023-10-02_WingsTest
Created: 2023-10-02 14:33:14 MESZ
Description: Erster Test
Generate Incident Report
All Items in CSV (Timeline)
Generate
Delete Host
Cancel Host
Open Host
Collection Tool
Options
About

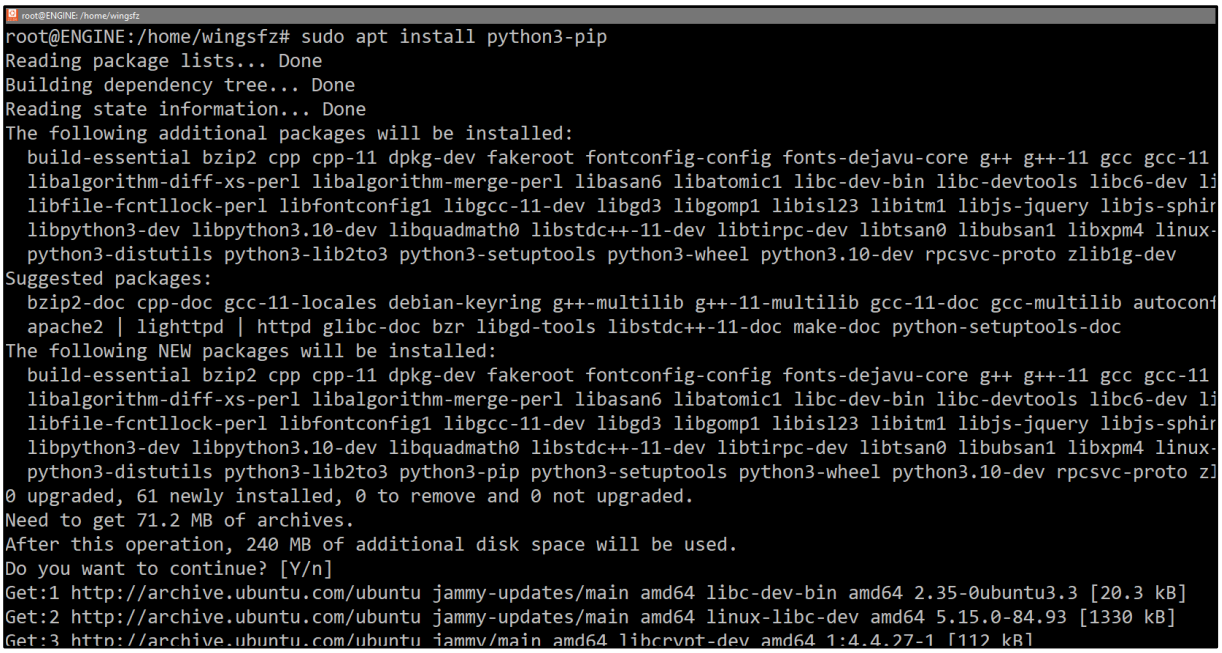

Cyber Triage | Collection Summary
CYBER TRIAGE
Options
User Guide
Close Host
Dashboard
Incident: 2023-10-02_WingsTest
Hostname: test1
Created Date: 2023-10-02 14:36:31 MESZ
Dashboard
Bad Items
Suspicious Items
Users
Accounts
Inbound Logons
Outbound Logons
Network Shares
Web Artifacts
Data Accessed
Malware
Startup Items
Triggered Tasks
Processes
Active Connections
Listening Ports
DNS Cache
System Configuration
OS Config Settings
Files
Timeline
Registry Entries
Search
Collection Details
Bad Items
0
Suspicious Items
0
Status
Targeted Analysis Complete
Full Scan Complete
Online File Reputation Not Requested Details
Report Choose Format Go
Recent Messages
Collecting All Files... (Step 12 of 12)
Collecting Interesting File Metadata... (Step 11 of 12)
Collecting Web Files... (Step 10 of 12)
Collecting Scheduled Tasks ... (Step 9 of 12)
Collecting User Accessed Data... (Step 8 of 12)
Collecting System Configuration... (Step 7 of 12)
Collecting Network Shares... (Step 6 of 12)
Collecting Processes... (Step 5 of 12)
Collecting Startup Items... (Step 4 of 12)
Collecting Event Logs... (Step 3 of 12)
Collecting Users... (Step 2 of 12)
Enumerating Files... (Step 1 of 12)
Collection Information
Collection Date 2023-10-02 14:38:11 MESZ
Collection Tool Version 3.6.0
Host Information
Local Host Name JEAN-13FBF038A3
Windows Product Name Microsoft Windows XP
Windows Install Date 2008-05-13 23:29:32 MESZ
Windows Version 5.1 (Build 2600)
Bitlocker Encryption No
Mounted Drives
Background Tasks Status
No tasks running
Cancel
Error Messages
Timestamp
Error Level
Text
No Errors Occured
Bad Items Timeline
No Bad Items To Timeline

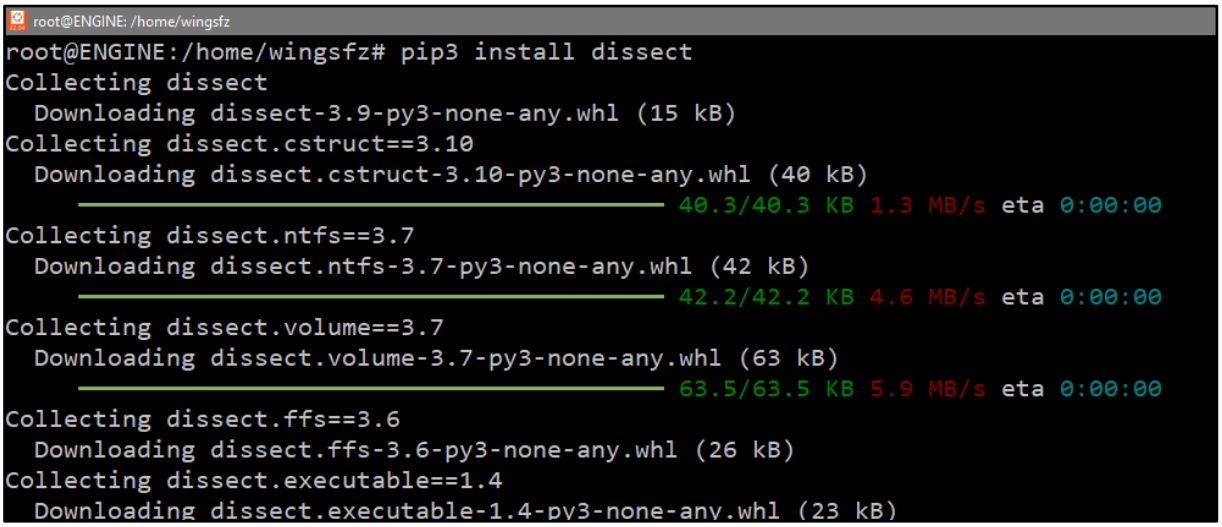
A.2 Screenshots zur Installation von WSL und Dissect








root@ENGINE: /home/wingsfz
root@ENGINE:/home/wingsfz# sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev li
libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjs-jquery libjs-sphir
libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 libxpm4 linux-
python3-distutils python3-lib2to3 python3-setuptools python3-wheel python3.10-dev rpcsvc-proto zlib1g-dev
Suggested packages:
bzip2-doc cpp-doc gcc-11-locales debian-keyring g++-multilib g++-11-multilib gcc-11-doc gcc-multilib autoconf
apache2 | lighttpd | httpd glibc-doc bzr libgd-tools libstdc++-11-doc make-doc python-setuptools-doc
The following NEW packages will be installed:
build-essential bzip2 cpp cpp-11 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-11 gcc gcc-11
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1 libc-dev-bin libc-devtools libc6-dev li
libfile-fcntllock-perl libfontconfig1 libgcc-11-dev libgd3 libgomp1 libisl23 libitm1 libjs-jquery libjs-sphir
libpython3-dev libpython3.10-dev libquadmath0 libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 libxpm4 linux-
python3-distutils python3-lib2to3 python3-pip python3-setuptools python3-wheel python3.10-dev rpcsvc-proto zl
0 upgraded, 61 newly installed, 0 to remove and 0 not upgraded.
Need to get 71.2 MB of archives.
After this operation, 240 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc-dev-bin amd64 2.35-0ubuntu3.3 [20.3 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 linux-libc-dev amd64 5.15.0-84.93 [1330 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libcrypt-dev amd64 1:4.4.27-1 [112 kB]


root@ENGINE: /home/wingsfz
root@ENGINE:/home/wingsfz# pip3 install dissect
Collecting dissect
Downloading dissect-3.9-py3-none-any.whl (15 kB)
Collecting dissect.cstruct==3.10
Downloading dissect.cstruct-3.10-py3-none-any.whl (40 kB)
40.3/40.3 KB 1.3 MB/s eta 0:00:00
Collecting dissect.ntfs==3.7
Downloading dissect.ntfs-3.7-py3-none-any.whl (42 kB)
42.2/42.2 KB 4.6 MB/s eta 0:00:00
Collecting dissect.volume==3.7
Downloading dissect.volume-3.7-py3-none-any.whl (63 kB)
63.5/63.5 KB 5.9 MB/s eta 0:00:00
Collecting dissect.ffs==3.6
Downloading dissect.ffs-3.6-py3-none-any.whl (26 kB)
Collecting dissect.executable==1.4
Downloading dissect.executable-1.4-py3-none-any.whl (23 kB)

A.3 Links zu den Beispiel-Disk-Images

Image-Name	Format	Quelle
nps-2008-jean.E01	EWF (E01)	https://digitalcorpora.org/corpora/scenarios/m57-jean/
nps-2009-domexusers.E01		https://downloads.digitalcorpora.org/corpora/drives/nps-2009-domexusers
Test-Image-01.E01		www.4n6.de (Zugang nur für Berechtigte)
Test-Image-02.E01		
Test-Image-03.E01		
Test-Image-04.E01		
Laptop1Final.E01		https://cfreds.nist.gov/all/MagnetForensics/2022WindowsMagnetCTF
20348.169.amd64fre.fe_release_svc_refresh.210806-2348_server_serverdatacentereval_en-us.vhd	VHD (Virtual Hard Disk)	https://developer.microsoft.com/de-de/windows/downloads/virtual-machines/
WinDev2309Eval-disk1.vmdk	VMDK (VMware)	https://developer.microsoft.com/de-de/windows/downloads/virtual-machines/
SCHARDT.001	RAW (DD)	https://cfreds-archive.nist.gov/Hacking_Case.html