Hochschule Wismar
University of Applied Sciences
Technology, Business and Design
Fakultät für Ingenieurwissenschaften, Fachbereich Informatik

Möglichkeiten und Grenzen des Windows System Resource Usage Monitor (SRUM) als forensisches Artefakt unter Windows 11

Abschlussarbeit

zur Erlangung des akademischen Grades

Bachelor of Engineering (B. Eng.)

im Fernstudiengang IT-Forensik

Erstgutachter:
Zweitgutachter:
Eingereicht von: Simon Schneider
Matrikelnummer:
Datum der Abgabe:

Aufgabenstellung

Titel: Möglichkeiten und Grenzen des Windows System Resource Usage Monitor (SRUM) als forensisches Artefakt unter Windows 11

Title: Capabilities and Limitations of the Windows System Resource Usage Monitor (SRUM) as a Forensic Artifact on Windows 11

Das Ziel der Bachelorthesis liegt in der Untersuchung des Verhaltens des Windows *System Resource Usage Monito*r (SRUM) unter Windows 11. Hierbei wird das Artefakt, im Sinne des Informationsgehaltes und auf deren forensische Relevanz, bewertet. Mithilfe von Testfällen sollen die zugrunde liegenden Eigenschaften und Funktionsweisen identifiziert und nachvollzogen werden. Es werden dabei forensische Werkzeuge zur Extraktion, Aufbereitung und Analyse genutzt. Zudem soll eine mögliche Darstellung von Grenzen des Artefakts, sowie deren Kompensation, in Verbindung mit anderen Windows Artefakten, stattfinden. Die in dieser Thesis verwendete Forschungsmethode basiert auf den Beobachtungen von durchgeführten Experimenten. Dabei soll die Thesis Antworten auf die folgenden Fragen liefern:

1. Können durch SRUM zuverlässige Ergebnisse zur Aufklärung forensischer W-Fragen [1] geliefert werden?
2. Welche Nutzeraktivitäten können durch SRUM nachgewiesen werden?
3. Ist SRUM vor einem möglichen Spurenverwischen (*Artefact Wiping*) geschützt?
4. Welche Grenzen weist SRUM auf und wie können diese kompensiert werden?

Kurzreferat

Seit Windows 8 verfügen die Betriebssysteme von Microsoft über eine neue Technologie, welche das Anwendungsverhalten überwacht und speichert. Diese Daten werden vom SRUM erfasst.

In der Bachelorthesis soll untersucht werden, ob Windows SRUM als forensisches Artefakt dienlich ist. Dazu findet ein Versuchsaufbau statt, in dem Nutzeraktionen simuliert werden. Anschließend wird eine forensische Auswertung der Daten durchgeführt, um Möglichkeiten und Grenzen von SRUM zu identifizieren.

Die Analyse von SRUM unter Windows 11 ergab, dass es als forensisches Artefakt einen Mehrwert bei der Aufklärung von Vorfällen bringt. Es kann Ergebnisse zur Aufklärung von forensischen W-Fragen [1] liefern, jedoch wurden auch Grenzen bei der Erfassung von Anwendungen ermittelt.

Abstract

Since Windows 8 the operating systems of Microsoft contain a new technology, which monitors and stores the behavior of applications. This kind of data is recorded by the System Resource Usage Monitor (SRUM).

This Bachelor's thesis examines whether Windows SRUM is a useful forensic artifact. Thus, a test environment was set up to simulate the user activities. Subsequently, a forensics analysis of the data was carried out to identify the possibilities and limitations of SRUM.

The analysis of SRUM running in Windows 11 yielded that SRUM does provide additional value as a forensic artifact in the resolution of incidents. It provides results which lead to the answering of central forensic questions [1]. However, there were also limitations in the detection of some applications.

Inhaltsverzeichnis

Abkürzungsverzeichnis

1 Einleitung

Digitale Informationen stellen für Unternehmen einen zunehmenden Mehrwert dar. Beeinträchtigung, Manipulation, Zerstörung oder Diebstahl können nicht absehbaren Folgen haben. Aufgrund unzureichender Daten kann die Aufklärung des Tathergangs häufig nicht oder nur mit hohem Aufwand erfolgen. Forensische Analysten sind daher oft mit der Frage konfrontiert, ob ein Ausführungsnachweis der Anwendung vorliegt und ob diese zur Datenexfiltration genutzt wurde.

Im Zuge der Veröffentlichung von Windows 8 wurde eine neue Funktion mit dem Namen *System Resource Usage Monitor* (SRUM) eingeführt. Diese Funktion ermöglicht unter anderem einen Einblick in die historische Anwendungsnutzung auf einem Computer.

Das Ziel der Arbeit ist die Aufdeckung des Verhaltens des Windows SRUM unter Windows 11. Dieses Artefakt wird im Sinne des Informationsgehaltes und auf deren forensische Relevanz bewertet. Zudem soll eine mögliche Darstellung von Grenzen des Artefakts, sowie deren Kompensation, in Verbindung mit anderen Windows Artefakten, stattfinden. Hierzu soll ein Testfall geschaffen werden, welcher mögliche Antworten auf die folgenden Forschungsfragen liefert:

1. Können durch SRUM zuverlässige Ergebnisse zur Aufklärung forensischer W-Fragen [1] geliefert werden?
2. Welche Nutzeraktivitäten können durch SRUM nachgewiesen werden?
3. Ist SRUM vor einem möglichen Spurenverwischen (*Artefact Wiping*) geschützt?
4. Welche Grenzen weist SRUM auf und wie können diese kompensiert werden?

Die Erkenntnisse in der Thesis beziehen sich auf die Auswertung der geparsten SRUM-Datenbanken, sowie auf die Beobachtungen der durchgeführten Experimente. Es findet keine vollumfängliche Analyse der Funktionsweise von SRUM, der zugrunde liegenden Datenbankstruktur oder den genutzten

Telemetriediensten statt. Zudem beruhen die Erkenntnisse auf einer Testumgebung, welche in der Version ‚Microsoft Windows 11 Pro 21H2 22000.778‘ betrieben wurde. Aussagen innerhalb dieser Thesis haben somit keine Allgemeingültigkeit, sondern konnten nur in der genannten Version verifiziert werden.

2 Grundlagen

Im nachfolgenden Kapitel soll ein grundlegendes Verständnis für IT-Forensik, Windows-Diagnose- und -Telemetriedaten, Extensible Storage Engine (ESE) sowie Windows-Artefakte geschaffen werden.

Innerhalb der Thesis wird nach Methoden der IT-Forensik gearbeitet. Ein grundlegendes Verständnis über Anforderungen an einen Ermittlungsprozess, gängige Leitfragen einer Untersuchung sowie anerkannte Vorgehensmodelle zur Beweiserhebung und -auswertung sind nötig, um Vorgehensweisen und Interpretationen innerhalb der Thesis nachvollziehen zu können.

Anschließend werden Erkenntnisse über die Funktionsweise der Erhebung von Windows-Diagnose- und -Telemetriedaten zusammengetragen sowie das ESE-Datenbankformat vorgestellt. Beide Abschnitte dienen der Einordnung von SRUM im Kontext des Betriebssystem Microsoft Windows. Zum Abschluss werden Windows-Artefakte betrachtet, die zur späteren Kompensation von SRUM genutzt werden.

2.1 IT-Forensik

In diesem Abschnitt wird zunächst der Begriff der IT-Forensik erläutert. Zudem werden Anforderungen an einen forensischen Prozess, forensische Fragen sowie gängige Vorgehensmodelle, zur Untersuchung eines Vorfalls, angeführt. Abschließend wird auf die Antiforensik eingegangen.

2.1.1 Definition

Der Begriff ‚forensisch' leitet sich vom lateinischen Wort *forensis* ab und wird folgendermaßen definiert: „zum Forum, zum Markt gehörig, Markt" [2]. In der Antike wurden auf dem Marktplatz/Forum öffentliche Gerichtsverhandlungen abgehalten. Der Begriff ‚forensisch' kann demnach als „für den Gebrauch vor

Gericht“ definiert werden [3]. Aus diesem Verwendungszweck ergeben sich Anforderungen, die im nachfolgenden Kapitel 2.1.2 erläutert werden.

Die IT-Forensik stellt ein Teilgebiet der allgemeinen Forensik da. Eine einheitliche Bestimmung des Begriffs existiert jedoch nicht, da diese an den Eigenschaften der jeweiligen Technik ausgerichtet ist [3]. Eine mögliche Bestimmung wurde durch das Bundesamt für Sicherheit in der Informationstechnik (BSI) vorgenommen und lautet wie folgt:

„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.“ [1].

2.1.2 Anforderungen an den Ermittlungsprozess

Forensisch erhobene Daten sind voranging für die Nutzung als Beweise in Gerichtsverfahren vorgesehen. Aus diesem Zweck ergeben sich spezielle Anforderungen an den Ermittlungsprozess, u. a. an die Beweissicherung, -verwahrung und -auswertung. Durch ein streng methodisches Vorgehen soll die Gerichtsverwertbarkeit von Beweisen bewahrt werden.

Nach Geschonneck [4] sind die allgemeinen Anforderungen, die Ermittler im Vorfeld an die zum Einsatz kommenden Methoden und Hilfsmittel stellen sollten, die folgenden:

- Akzeptanz
- Glaubwürdigkeit
- Wiederholbarkeit
- Integrität
- Ursache und Auswirkungen
- Dokumentation

Innerhalb der Bachelorthesis erfolgt die Umsetzungen der genannten

Anforderungen durch den Einsatz anerkannter forensischer Programme. Weitgehend unbekannte Programme werden vor der Verwendung geprüft. Hierdurch soll sichergestellt werden, dass Programme, mit denen Spuren gesichert und analysiert werden, auch dafür geeignet sind.

2.1.3 Leitfragen im Ermittlungsprozess

Neben den Anforderungen an die genutzten Methoden und Instrumente sollten Ermittler sogenannte W-Fragen stellen [1]. Diese werden nachfolgend als ‚Leitfragen' bezeichnet und können als Orientierung dienen, um ein Teillagebild eines Vorfalls zu erarbeiten. Mit vier dieser abgewandelten Fragen soll im Laufe der vorliegenden Thesis der forensische Informationsgehalt von SRUM bestimmt werden. Die Leitfragen lauten wie folgt [1]:

- Was ist geschehen?
- Wo ist es passiert?
- Wann ist es passiert?
- Wie ist es passiert?
- Wer hat es getan?
- Was kann gegen eine Wiederholung getan werden?

2.1.4 Vorgehensmodelle in der IT-Forensik

Aufgrund der Anforderungen an den forensischen Prozess wurden in den letzten 20 Jahren verschiedene Vorgehensweisen zur IT-forensischen Untersuchung und Beweissicherung entwickelt. Jedoch existiert kein einheitliches Modell, das für alle digitalen forensischen Analysen übernommen werden kann.

International anerkannt sind u. a. die Verfahren nach Casey (2004), Cohen (2009) und Kent et al (2006) sowie die Vorgehensweise des BSI [1, 5–9]. Der IT-Forensiker Sachowski [7] veröffentlichte im Jahr 2016 ein High-Level Modell (s. Bild **1**), das in vier Phasen und sieben Schritten unterteilt ist.

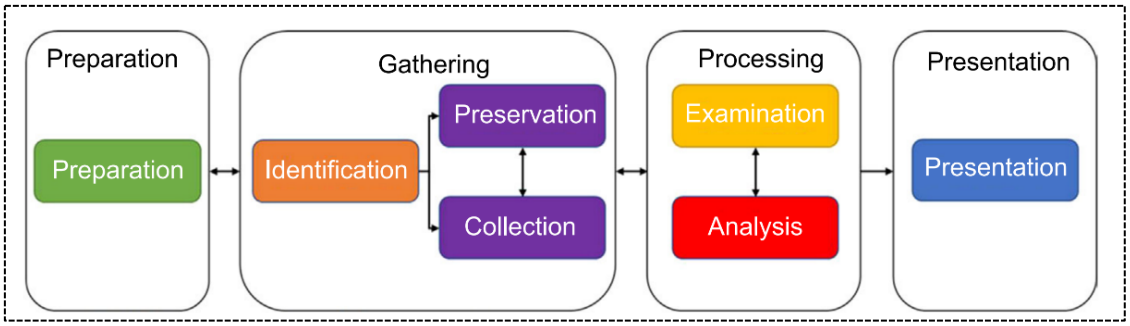

Bild 1: High-Level-Vorgehensmodell aus der IT-Forensik [7]

Dieses Modell wird im Verlauf dieser Bachelorthesis für Untersuchungen angewandt, jedoch nicht weiter explizit benannt. Aus diesem Grund findet nachfolgend eine zusammenfassende Erläuterung der einzelnen Phasen statt.

In der ersten Phase werden strategische und organisatorische Entscheidungen getroffen sowie grundlegende Fragen zum Untersuchungsgegenstand beantwortet. Die Umsetzung findet in Kapitel 4 statt. In der zweiten Phase werden fallrelevante Daten identifiziert, extrahiert und anschließend dupliziert. Diese Maßnahmen werden in Kapitel 5.5 näher erläutert. Die dritte Phase wird in Kapitel 6 dargelegt. Hierbei werden die gesicherten Daten analysiert und aufbereitet. In der letzten Phase wird die Auswertung der resultierenden Daten verständlich zusammengefasst und präsentiert. Die Umsetzung erfolgt im Rahmen des Kolloquiums.

2.1.5 Antiforensik

Forensischen Auswertungen können aufgrund gegensätzlicher Handlungen erschwert werden. Unter Anti- oder Counter-Forensik werden Maßnahmen verstanden, durch die die Analysen und deren Ergebnisse verfälscht, unterdrückt oder sogar zerstört werden können [10–12]. Gemäß Conlan (2016) [13] können Antiforensik-Handlungen in folgende Kategorien eingeordnet werden [11]:

- Verstecken von Daten (*Data Hiding*)
- Löschen von Artefakten (*Artefact Wiping*)
- Verschleiern der Herkunft/Quelle (*Trail Obfuscation*)
- Angriffe auf Forensik-Tools (Attacks against Forensics Process or Tool)

Im Rahmen dieser Thesis werden zwei Antiforensik-Maßnahmen gegen Windows SRUM durchgeführt und anschließend ausgewertet.

2.2 Windows-Diagnose- und -Telemetriedaten

In diesem Abschnitt werden Erkenntnisse über die Funktionsweise der Erhebung von Windows-Diagnose- und -Telemetriedaten zusammengetragen. Zudem finden Begriffsdefinitionen und eine Einordnung von SRUM, innerhalb dieser Begrifflichkeiten, statt.

2.2.1 Definition

Unter dem Begriff Windows-Diagnose- und -Telemetriedaten wird ein automatischer Prozess zur Sammlung und Überwachung von Daten auf einem Computersystem verstanden. Ziel der Erhebung ist eine Personalisierung und Produktverbesserung sowie eine Integritäts-, Qualitäts-, Sicherheits- und Leistungsanalyse zur Optimierung der bereitgestellten Programme und Dienste. Hierzu finden Messungen, Überwachungen und Analysen von Anwendungen statt. Die erhobenen Daten werden anschließend aggregiert und an eine von Microsoft betriebene Backend-Infrastruktur gesendet [14, 15].

Je nach Ausmaß kann die Datenerhebung unter Windows 11 (*Enterprise*, *Education* und *Professional*) in drei, vormals vier [16], Kategorien unterteilt werden. Eine Übersicht ist in Tabelle **1** dargestellt.

Tabelle 1: Stufen der Diagnosedateneinstellungen unter Windows 11

Einstellungen alt	Einstellungen neu
0 – Sicherheit	Diagnosedaten aus (nicht empfohlen)
1 – Standard	erforderliche Diagnosedaten senden
2 – erweitert	erforderliche Diagnosedaten senden
3 – vollständig	optionale Diagnosedaten senden

Im Einstellungsmodus ‚Diagnosedaten aus' werden keine Diagnosedaten an Microsoft gesendet. Eine lokale Datenerhebung findet dennoch statt. Dieser Modus ist nur in den Versionen Windows Server, Windows Education und Windows Enterprise verfügbar.

Im Einstellungsmodus ‚Standard/Erforderliche Diagnosedaten senden' werden im begrenzten Umfang Daten gesammelt und an Microsoft gesendet. Diese sind gemäß Microsoft [15] erforderlich, um die Sicherheit, Aktualität und erwartungsgemäße Funktion des Geräts sicherzustellen.

Im optionalen Einstellungsmodus werden zusätzlichen Daten zu aufgerufenen Webseiten u. a. Informationen zur Nutzung und Leistung von Apps sowie Absturzabbilder (vollständige und Triage-Speicherabbilder) gesendet [15].

2.2.2 Funktionsweise der Datenerhebung

Zur Erhebung der Diagnose- und -Telemetriedaten wird die Protokollierungsfunktion ‚Ereignisablaufverfolgung für Windows-Protokollierung' (*Event Tracing for Windows*, ETW) genutzt. Hierbei handelt es sich um ein im Kernel implementiertes Diagnose- und Logging-Framework von Windows. Es ermöglicht eine dynamische und hochperformante Protokollierung von Events (Ereignissen). Diese umfasst Anwendungen, Dynamic-Link-Library(DLL)-Dateien, Kerneldaten und Treiber [17, 18]. Mithilfe von Providern werden Ereignisse aus bestimmten Kategorien des Kernel- und User-Modes erhoben und an die ETW-Sitzungen übermittelt (s. Bild **2**).

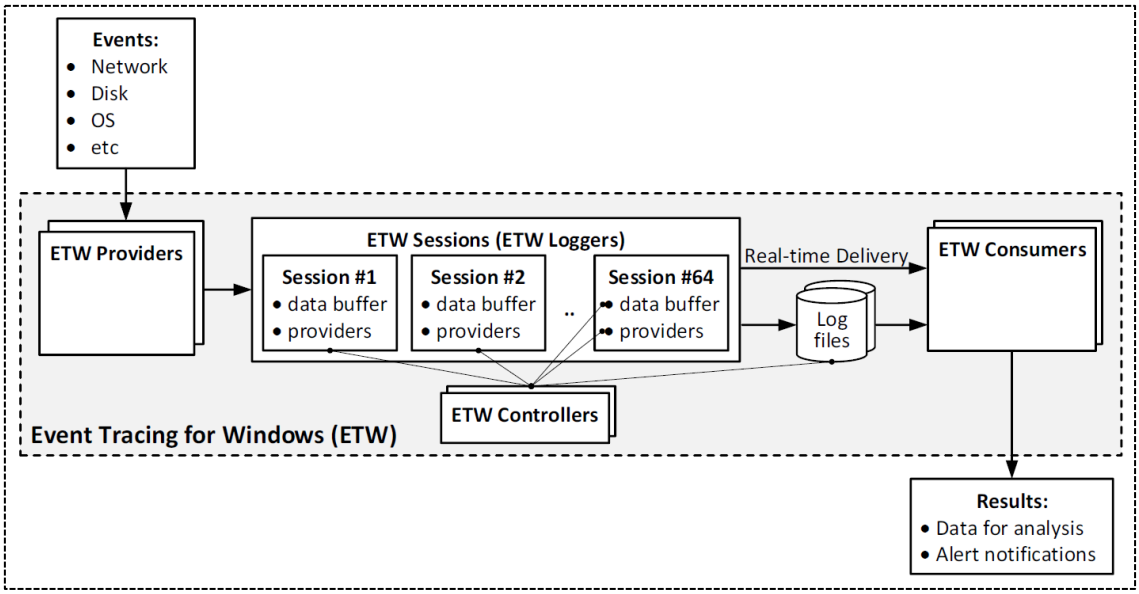

Bild 2: Funktionsweise von ETW [19]

Unter Windows 11 kann ETW mehr als 50 000 Events von über 1000 Providern erfassen [19]. Hierzu gehören u. a. die Provider App Timeline, Windows Push Notifications (WPN) SRUM und Energy Usage. Diese werden von SRUM zur Datenerhebung genutzt [20] und im Verlauf der Thesis genauer betrachtet.

2.2.3 Diagnostic Policy Service

Der Diagnostic Policy Service (DPS) stellt einen Teil der Windows-Diagnose-Infrastruktur (WDI) dar [21, 22], welcher zur Diagnose- und -Telemetriedatenerhebung genutzt wird. Der Dienst DPS ermöglicht die Problemerkennung, Fehlerbehebung und -lösung für Windows-Komponenten. Er ist standardmäßig aktiviert und so konfiguriert, dass er beim Hochfahren des Systems automatisch gestartet wird. Windows SRUM ist Teil des DPS [20] und nutzt ETW-Provider zur Datenerhebung.

2.3 ESE

Bei Windows SRUM handelt es sich um eine ESE-Datenbank, welche im Verlauf

der Thesis analysiert wird. In diesem Abschnitt werden bisherige Erkenntnisse über die Windows ESE aus der Literatur zusammengetragen. Zu Beginn wird eine Begriffsdefinition vorgenommen. Anschließend werden die Verwendung, Identifizierung, Überprüfung und Rekonstruktion einer ESE-Datenbank dargelegt.

2.3.1 Definition

Bei der ESE handelt es sich um ein proprietäres relationales Datenbankmanagementsystem (RDBMS), das vom Hersteller Microsoft seit der Version Windows NT 3.51 eingesetzt wird. Die Engine nutzt die Microsoft Joint Engine Technology (JET) Blue. Das Äquivalent der Engine ist die Microsoft JET Red, die in Microsoft-Access-Datenbanken zum Einsatz kommt [23–26].

Die JET Blue verwendet die von Fa. IBM Ende der 1960er Jahre entwickelte Index Sequential Access Method (ISAM) [27, 28], die einen wahlfreien oder sequenziellen indexbasierten Zugriff ermöglicht. Des Weiteren werden diskrete Transaktionen und Protokolldateien zur Integritätswahrung genutzt. Hierzu wird jede Erstellung, Löschung oder Änderung von Objekten als Zusammenfassung einer Folge von Operationen behandelt. Da eine Transaktion als unteilbare Einheit gilt, werden entweder alle darin erfolgenden Vorgänge erfolgreich abgeschlossen und dauerhaft gespeichert oder es wird keine Operation durchgeführt [25].

2.3.2 Verwendung der ESE

Die ESE wird in den nachfolgenden drei Versionen eingesetzt:

- Exchange 5.5 in der Version ESE97
- Exchange 2000 und höher in der Version ESE98
- Windows NT und höher in Version ESENT

In aktuellen Betriebssystemen kommt die Variante ESENT zum Einsatz. Der Verwendungszweck der ESE umfasst u. a. Windows SRUM sowie weitere

Dienste, die auszugsweise in Tabelle **2** dargestellt sind [24].

Tabelle 2: Verwendungszwecke der Windows ESENT

Verwendungszweck	Dateiname
Exchange Server	priv1.edb
Active Directory	ntds.dit
Windows Desktop Search	Windows.edb
Windows Mail	WindowsMail.MSMessageStore
Windows SRUM	SRUDB.dat

2.3.3 Identifizierung einer ESE-Datenbank

Die Identifizierung einer ESE-Datenbank (EDB) kann mithilfe der eindeutigen Signatur erfolgen. Diese befindet sich im Datenbank-Header und wird in hexadezimaler Schreibweise als Wert ‚\xef \xcd \xab \x89' auf Höhe des Offset 4 (0-basiert) dargestellt [23]. In Bild **3** ist diese Signatur zu sehen.

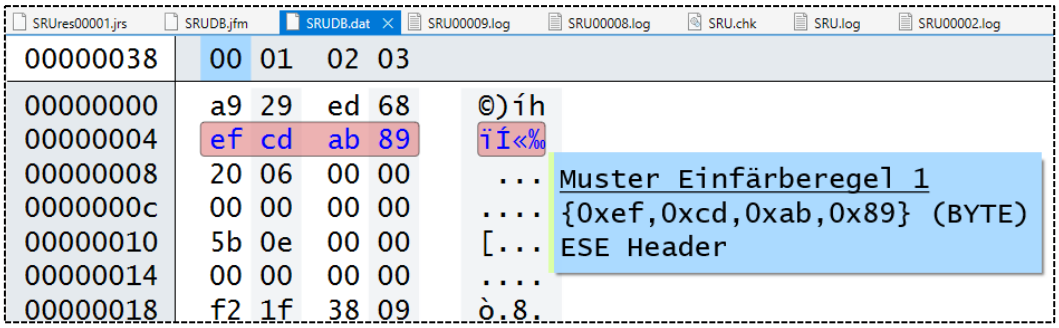

Bild 3: EDB-Signatur im Hex-Editor

2.3.4 Überprüfung einer ESE-Datenbank

Als Datenbankdienstprogramm für ESE stellt Microsoft das Programm Esentutl bereit. Hierbei handelt es sich um ein Befehlszeilentool, das u. a. zur Betrachtung, Integritätsüberprüfung oder Reparatur von EDB genutzt werden kann [23, 29]. Unter der Angabe des Befehls ‚/mh' (m = mode-*modifier*, h = *dump database header*) kann der Zustand einer EDB kontrolliert werden. In Bild **4** ist

der Zustand einer sauberen (clean) Datenbank dargestellt.

```
C:\Users\srum\Desktop\sru>esentutl /mh SRUDB.dat

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

Initiating FILE DUMP mode...
         Database: SRUDB.dat


DATABASE HEADER:
Checksum Information:
Expected Checksum: 0x8f1d8d91
  Actual Checksum: 0x8f1d8d91

Fields:
        File Type: Database
         Checksum: 0x8f1d8d91
   Format ulMagic: 0x89abcdef
   Engine ulMagic: 0x89abcdef
 Format ulVersion: 0x620,200,440  (attached by 9400)
 Engine ulVersion: 0x620,200,440  (efvCurrent = 9400)
Created ulVersion: 0x620,20
     DB Signature: Create time:07/04/2022 11:14:35.046 Rand:3586673415 Computer:
         cbDbPage: 4096
            State: Clean Shutdown
     Log Required: 0-0 (0x0-0x0) [Min. Req. Pre-Redo: 0 (0x0)]
Gen. Max. Req. Creation Time: 00/00/1900 00:00:00.000 LOC
    Log Committed: 0-0 (0x0-0x0)
```

Bild 4: Metadaten eines EDB-Headers

2.3.5 Rekonstruktion und Betrachtung einer ESE-Datenbank

Es kann vorkommen, dass eine EDB nicht ordnungsgemäß geschlossen wurde und sich somit in einem unsauberen (*dirty*) Zustand befindet. Dieser kann mit Esentutl (s. Bild **5**) oder im Hex-Editor überprüft werden.

```
Initiating FILE DUMP mode...
         Database: SRUDB (3).dat


DATABASE HEADER:
Checksum Information:
Expected Checksum: 0x86e49714
  Actual Checksum: 0x86e49714

Fields:
        File Type: Database
         Checksum: 0x86e49714
   Format ulMagic: 0x89abcdef
   Engine ulMagic: 0x89abcdef
 Format ulVersion: 0x620,200,440  (attached by 9400)
 Engine ulVersion: 0x620,110,240  (efvCurrent = 9180)
Created ulVersion: 0x620,20
     DB Signature: Create time:03/05/2022 16:20:02.661 Rand:118928735 Computer:
         cbDbPage: 4096
            State: Dirty Shutdown
     Log Required: 92-92 (0x5c-0x5c)
    Log Committed: 0-92 (0x0-0x5c)
```

Bild 5: EDB im Status *Dirty Shutdown*

In Bild **6** ist auf der Höhe von Offset 52 (0-basiert) eine unsaubere EDB [30] (Wert = 2) dargestellt.

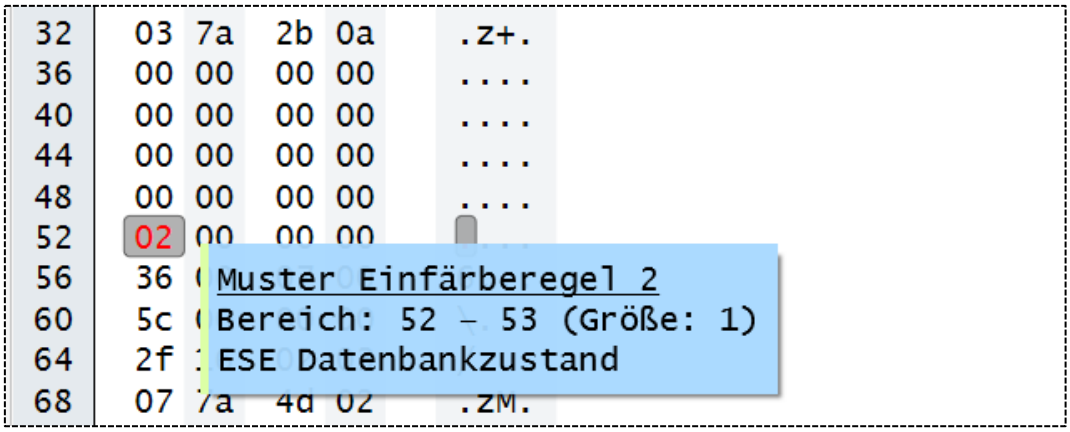

Bild 6: *Dirty Shutdown* EDB im Hex-Editor

Zur Rekonstruktion der EDB kann eine Wiederherstellung (*Recovery*) durchgeführt werden, sofern die erforderlichen Transaktionslogs vorhanden sind [23]. Hierfür kann ebenfalls das Tool Esentutl verwendet werden.

Nach einer erfolgreichen Wiederherstellung und einem anschließenden Integritätscheck kann die EDB mit Programmen betrachtet werden, die die Bibliothek libesedb [31] nutzen. Hierunter fällt das Programm NirSoft ESEDatabaseView, welches ein *Graphical User Interface* (GUI) für EDB-Tabellen bietet.

2.4 Windows-Artefakte

In diesem Abschnitt wird zunächst der Begriff ‚Artefakt' erläutert. Anschließend werden zwei Windows-Artefakte vorgestellt, die zur späteren Beantwortung von unbeantworteten forensischen Leitfragen genutzt werden. Eine Kompensation mithilfe der beiden Windows-Artefakte ist nötig, da sich im Verlauf der Thesis herausstellt, dass SRUM unzuverlässige Ergebnisse in der Aufklärung forensischer Leitfragen liefert.

2.4.1 Definition

Unter Artefakten werden Objekte verstanden, die durch Verwendung einer Soft- oder Hardware erstellt wurden. Sie können u. a. den Nutzungsverlauf eines

Anwenders zeigen und somit potenzielle digitale Beweise beinhalten [32].

2.4.2 ActivitiesCache.db

ActivitiesCache ist eine SQLite-Datenbank (Version 3), die mit Windows 10 Version 1803+ [33] eingeführt wurde. Sie ist Teil der Connected Devices Platform (CDP), die zur Interaktion zwischen PCs und Smartphones genutzt wird [34, 35]. Mithilfe der ActivitiesCache-Datenbank können Ausführungszeitpunkt und -dauer sowie der Pfad einer Anwendung nachvollzogen werden.

2.4.3 Prefetch

Windows-Prefetch(PF)-Dateien werden verwendet, um den Start einer Anwendung zu beschleunigen. Hierzu beobachtet der Microsoft-Cache-Manager, der Teil des Memory-Manager ist, für ca. 10 Sekunden die Anwendung sowie alle Dateien und Verzeichnisse, die von dieser Anwendung benötigt werden. Diese Informationen werden in einer PF-Datei gespeichert. Beim nächsten Ausführen wird die Anwendung schneller gestartet, da Codepages bereits vor der Benutzung in den Arbeitsspeicher geladen wurden [36, 37].

Seit Windows 8 können bis zu 1024 PF-Dateien vom System gespeichert werden. Diese werden nicht automatisch gelöscht, wenn ein Programm deinstalliert oder entfernt wird [33, 38–40]. Daher ist eine vorhandene PF-Datei ein Beweis dafür, dass ein Programm auf einem System ausgeführt wurde [41].

3 Windows SRUM

In der vorliegenden Thesis wird Windows SRUM als forensisches Artefakt unter Windows 11 analysiert. Zu den späteren Beantwortungen der Forschungsfragen ist ein grundlegendes Verständnis über SRUM nötig. In diesem Kapitel werden Grundlagen zum Aufbau und die bisher bekannten Funktionsweisen von SRUM dargelegt. Die angeführten Erkenntnisse entstammen wissenschaftlichen Fachpublikationen und Blogeinträgen. Hierbei beinhaltet die gesichtete Literatur Informationen zu SRUM unter Windows 8, 8.1 und 10 sowie Windows Server 2019.

3.1 Allgemeines

Khatri (2015) [20] beschäftigte sich erstmals im wissenschaftlichen Kontext mit Windows SRUM. Er stellte fest, dass es sich bei Windows SRUM um eine EDB mit dem Dateinamen SRUDB.dat handelt. SRUM wurde unter Windows 8 als neue Funktion eingeführt und ist seitdem in den nachfolgenden Versionen standardgemäß integriert. Gemäß der Arbeit von Khatri (2015) [20], liefert Windows SRUM einen teilweise historischen Einblick (30 bis 60 Tage) in die Nutzung von Anwendungen und könnte daher als forensische Beweisquelle für eine Anwendungsausführung dienen.

3.2 SRUM aus Benutzersicht

Windows-Benutzer haben einen beschränkten Einblick in die SRUM-EDB. So kann lediglich eine gekürzte Zusammenfassung im Taskmanager unter ‚App-Verlauf' betrachtet werden. Diese zeigt jedoch nur kumulierte Statistiken der vergangenen 30 Tage und keinen detaillierten Anwendungsverlauf [42]. Die Zusammenfassung enthält des Weiteren Informationen zur CPU-Zeit, zum Netzwerk, zum getakteten Netzwerk sowie zu Kachelupdates.

3.3 Eigenschaften von Windows SRUM

Windows SRUM ist ein Teil des Windows-Dienstes DPS. Dieser Dienst erhebt Statistiken zur Systemressourcennutzung von Programmen und Services. Hierbei werden prozessbezogene Informationen, wie der Anwendungspfad, die Security Identifiers (SID) des Benutzerkontos sowie Netzwerkdatenmengen aufgezeichnet [20, 40]. Diese Informationen werden anschließend in die SRUDB.dat gespeichert. Während SRUM kontinuierlich Daten über das laufende System sammelt, aktualisiert sich die SRUM-EDB standardmäßig nur in regelmäßigen Abständen von einer Stunde [20, 40, 43]. Wird das System ordnungsgemäß heruntergefahren, so erfolgt ein sofortiges Schreiben der Datenbank [20].

3.4 Tabellenübersicht der SRUDB.dat

In diesem Abschnitt werden die SRUM-Providertabellen und deren Funktion erläutert. Zunächst werden allgemeine Informationen zum Anwendungspfad und zur Bezeichnung der Tabellenstruktur dargelegt. Danach werden die einzelnen Provider und deren bisher bekannte Funktion erläutert.

3.4.1 Allgemeines

Mithilfe der Programme Esentutl und NirSoft ESEDatabaseView lässt sich die SRUM-EDB betrachten. Die SRUDB.dat hat den folgenden Speicherort:

%SystemRoot%\System32\sru\SRUDB.dat

Windows SRUM wird kontinuierlich durch Microsoft weiterentwickelt. So verfügt die SRUM-EDB mittlerweile über 14 Providertabellen [44], die im Anhang (s. Tabelle **14**) dargestellt sind [45–47].

Tabellen mit der Bezeichnung ‚MSys*‘ und ‚SruDb*‘ beinhalten Metadaten. Die restlichen nutzen einen Globally Unique Identifier (GUID) [48] als Bezeichner und umfassen erhobene Daten. Eine Namensauflösung der GUIDs ist mittels

Registry-Editors möglich. Diese können im Registry-Hive ‚Software' unter folgendem Speicherort betrachtet werden [46]:

\Microsoft\Windows NT\CurrentVersion\SRUM\Extensions

Unter jeder Extension ist ein Eintrag zur verwendeten Dynamic Link Library (DLL) und zum Providernamen zu finden, wie in Bild **7** dargestellt ist.

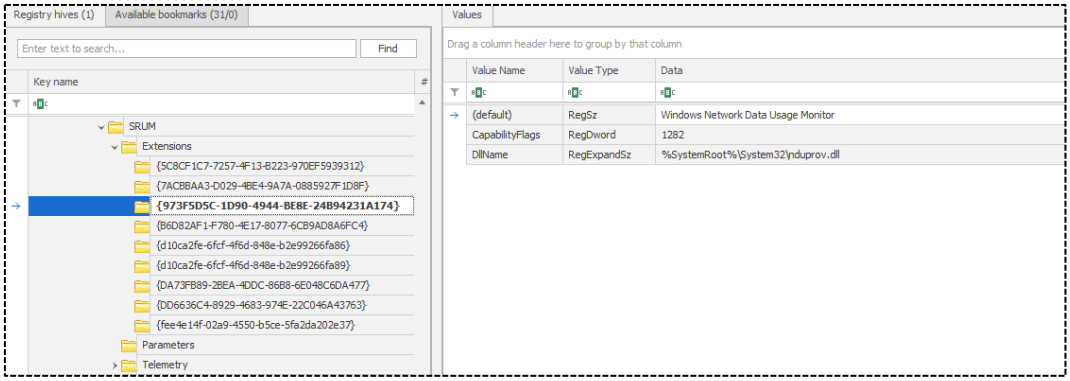

Bild 7: GUID-Darstellung im Tool Registry-Explorer

3.4.2 SRUM Provider

Der App Timeline Provider (GUID 5C8CF1C7-7257-4F13-B223-970EF5939312) liefert in 44 Spalten Auskunft über den Zeitpunkt des Schreibens in die SRUM-EDB, den Anwendungsnamen, die verwendete SIDs sowie Ein- und Ausgabezeiten von Tastatur, Maus und Ton [20, 49].

Der SRUM-Provider Vfuprov (GUID 7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F) besitzt acht Spalten. Der Verwendungszweck ist unklar, da keine Literatur dazu gefunden wurde.

Die Tabelle des Windows Network Data Usage Monitor Provider (GUID 973F5D5C-1D90-4944-BE8E-24B94231A174) speichert in neun Spalten Daten über den Netzwerkverkehr [43, 50, 51] und den verwendeten Interfacetyp in Form eines Local Unique Identifier (LUID) [48, 52, 53].

Der WPN SRUM Provider (GUID D10CA2FE-6FCF-4F6D-848E-B2E99266FA86) stellt, in sieben Spalten, Informationen über die Ausführung von

WPNs [54] bereit [20, 50].

Der Application Resource Usage Provider (GUID D10CA2FE-6FCF-4F6D-848E-B2E99266FA89) beinhaltet Daten über Lese- und Schreiboperationen von ausgeführten Anwendungen [20, 43, 50, 55].

Der Energy Estimation Provider (GUID DA73FB89-2BEA-4DDC-86B8-6E048C6DA477) verfügt über fünf Spalten, in denen u. a. Binärdaten gespeichert werden. Der Verwendungszweck dieser Tabelle ist unbekannt.

In neun Spalten liefert der Windows Network Connectivity Usage Monitor Provider (GUID DD6636C4-8929-4683-974E-22C046A43763) Informationen über verbundene Netzwerke, verwendete Netzwerkinterfaces und deren Verbindungsdauer [20, 43, 50].

Der Energy Usage Provider (GUID FEE4E14F-02A9-4550-B5CE-5FA2DA202E37) und der Energy Usage Provider Long Term (GUID {FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT) liefern in elf bzw. 16 Spalten Daten über den Energieverbrauch und Akkuzustand [46].

Der Tagged Energy Provider (GUID B6D82AF1-F780-4E17-8077-6CB9AD8A6FC4) nutzt sieben Spalten, in denen u. a. Binärdaten gespeichert werden. Der Verwendungszweck dieser Tabelle ist unbekannt.

Provider mit der Bezeichnung ‚SDP‘ wurden bisher nicht unter Windows 10, jedoch unter Windows Server 2019, nachgewiesen [56]. Die Bedeutung der Abkürzung SDP und deren Funktionen als Provider sind unbekannt.

4 Methodik

In dieser Thesis erfolgt eine Analyse der Möglichkeiten und Grenzen von SRUM unter Windows 11. Hierzu soll das Artefakt im Hinblick auf seinen Informationsgehalt und seine forensische Relevanz bewertet werden. In diesem Kapitel werden Vorüberlegungen und die Festlegung der Herangehensweise beschrieben, mit der die Möglichkeiten und Grenzen von SRUM unter Windows 11 analysiert werden sollen.

Zu Beginn des Abschnitts werden die vier Forschungsfragen aus Kapitel 1 genauer erläutert. Anschließend werden Anforderungen an den Aufbau und den Kontext des Versuchs erläutert. Im Abschnitt 4.3 wird auf die Forschungsumgebung sowie die verwendeten Programme zur Analyse eingegangen.

4.1 Erläuterung der Forschungsfragen

In diesem Abschnitt werden folgende vier Forschungsfragen im Zusammenhang mit der Thesis genauer betrachtet:

1. Können durch SRUM zuverlässige Ergebnisse zur Aufklärung forensischer W-Fragen [1] geliefert werden?
2. Welche Nutzeraktivitäten können durch SRUM nachgewiesen werden?
3. Ist SRUM vor einem möglichen Spurenverwischen (*Artefact Wiping*) geschützt?
4. Welche Grenzen weist SRUM auf und wie können diese kompensiert werden?

4.1.1 Frage 1

Das Artefakt SRUM soll im Hinblick auf seinen Informationsgehalt und seine forensische Relevanz bewertet werden. Um dies zu realisieren, soll beleuchtet

werden, ob durch SRUM Antworten auf die forensischen Leitfragen gegeben werden können. Da Letztere allgemein gehalten sind, wurden die passenden Fragen ausgewählt und an den Kontext angepasst. Die Darstellung der ausformulierten Fragen erfolgt in Tabelle **3**.

Tabelle 3: Vier Leitfragen in Bezug auf Windows SRUM

W-Fragen	Angepasste W-Fragen
Wer?	Welches Benutzerkonto hat eine Aktion ausgeführt?
Wo?	Wie lautet der Dateipfad der Aktion?
Wann?	Wann wurde die Aktion ausgeführt?
Wie?	Wie oft wurde eine Aktion ausgeführt?

Wie bereits in Kapitel 3 erläutert, kann SRUM Informationen zu Anwendungspfad, Ausführungszeitpunkt und Benutzerprofil liefern. In der späteren Auswertung (Kapitel 6) liegt der Fokus auf der Beantwortung der vier Fragen aus Tabelle **3**. Dabei sind besonders die Vollständigkeit (Erfassung aller Ergebnisse) und eine hohe Verlässlichkeit (Reproduzierbarkeit der Ergebnisse unter gleichen Bedienungen) relevant.

4.1.2 Frage 2

Die zweite Frage ist ergänzend zur ersten zu sehen. Sie dient der Erkennung weiterer Funktionsweisen, die Informationen über eine Anwendungsaktivität liefern.

4.1.3 Frage 3

Anhand dieser Frage sollte erforscht werden, ob Windows SRUM vor Antiforensik-Maßnahmen wie *Artefact Wiping* geschützt ist. Hierzu soll ein Cleaner-Tool genutzt werden, mit dem mögliche Spuren verwischt werden könnten. Zudem soll die Löschung des Verlaufs im Taskmanager durchgeführt werden, um eventuelle Auswirkung auf Windows SRUM zu untersuchen. Die

Beantwortung der Frage soll Informationen zur Robustheit des Artefakts liefern. Sollte sich SRUM z. B. durch das Löschen der Taskmanager-Historie manipulieren lassen, so ist der forensische Mehrwert gering, da eine solche Operation mit geringem Aufwand und eingeschränkten Systemrechten vorgenommen werden kann.

4.1.4 Frage 4

Mögliche Grenzen von SRUM sollen identifiziert und mit anderen bekannten Windows-Artefakten kompensiert werden. Hierzu werden forensische Leitfragen während der Auswertung beantwortet. Sollten durch SRUM unzuverlässige Ergebnisse zur Aufklärung der Leitfragen geliefert werden, wird mit anderen bekannten Windows-Artefakten eine Beantwortung der noch offenen Fragen angestrebt.

4.2 Vorüberlegungen zum Testfall

Zur Umsetzung des Testfalls sind Vorüberlegungen nötig. Hierzu wurden sich Anforderungen an die Durchführung des Versuchs und die Testumgebung überlegt. Diese werden nachfolgend aufgelistet. Zudem wird beschrieben, wie die Anforderungen erfüllt werden sollen.

4.2.1 Software für den Test

Die Auswahl der einzusetzenden Software erfolgt aufgrund mehrerer Faktoren. So werden Programme verwendet, die größtenteils kostenlos zur Verfügung stehen und deren Bedienung bekannt ist.

Um Spuren zu erzeugen, werden folgende Aktionen durchgeführt:

- Installation von Anwendungen
- Öffnen von Dateien, z. B. PDF- oder Textdokumenten
- Download von Dateien aus dem Internet via Browser und wie Cloud-

Synchronisations-Software
- Nutzung von Anwendungen aus dem Microsoft-App-Store
- Nutzung unterschiedlicher Netzwerke (Kabel, WLAN)
- Langes Öffnen von Anwendungen (mehr als eine Stunde)

Um ein mögliches normales Benutzerverhalten zu simulieren, wird Software aus den folgenden Kategorien verwendet:

- Textverarbeitung (OpenOffice, Notepad++, Notepad)
- Betrachtung von Dokumenten (PDF)
- Hören von Musik (VLC-Player)
- Betrachten von Videos (Amazon Prime App)
- Dateidownload aus dem Internet (Microsoft-Edge-Browser, Mega.io-Dateisynchronisierung)
- Nutzung von Passwortspeichern (KeePass)
- Löschen von Spuren (CCleaner)

4.2.2 Software zum parsen der SRUM-EDB

Durch die Betrachtung der SRUM-EDB mit Programmen wie NirSoft ESEDatabaseView wird eine forensische Analyse aufgrund der Rohdatendarstellung übermäßig komplex. Daher wurde eine Internetrecherche betrieben, um geeignete SRUM-Parser zu finden. Hierbei wurde entschieden, das Programm SrumMonkey von Devgc [57] nicht zu nutzen, da eine Python-Umgebung zur Nutzung nötig ist.

Die folgenden Instrumente wurden ausgewählt:

- SrumECmd von Eric Zimmerman [58]
- srum_dump2 von Mark Bagget [59]

Um die Eignungsfähigkeit der eingesetzten Tools sicherzustellen, muss eine Voranalyse durchgeführt werden. Hierbei erfolgt ein Abgleich der geparsten Tabellen mit der Rohdatendarstellung der Datenbank.

4.2.3 Sicherung der Testumgebung

Um die Windows-11-Umgebung zu installieren, wird eine virtuelle Maschine (VM) eingesetzt. Der Vorteil liegt hierbei vor allem darin, dass die Zustände des Systems durch Sicherungspunkte gespeichert und zurückgesetzt werden können.

4.2.4 Verhinderung von Messfehlern

Um die Nachvollziehbarkeit der Ereignisse und eine spätere Auswertung zu ermöglichen, werden die Aktionen protokolliert. Während des Testfalls wird händisch ein Protokoll geführt, mit dem die durchgeführten Aktionen und deren Dauer erfasst werden. Um hierbei Lücken oder Abweichungen zu verhindern, musste eine Lösung gefunden werden, um eine systemseitige Erfassung der Anwendungsausführung zu ermöglichen. Hierfür fiel die Wahl auf den kostenlosen Microsoft-Service System Monitor (SysMon) [60]. Dabei handelt es sich um einen Gerätetreiber, der detaillierte Systemaktivitäten aufzeichnet und im Windows-Event-Log abspeichert [61, 62].

4.2.5 Erfassung von Netzwerkverkehr

Um den Netzwerkverkehr von Anwendungen zu erfassen, wurden die folgenden drei Möglichkeiten identifiziert:

- händische Erfassung der Dateigröße beim Up- oder Download
- Einsatz des Tools NirSoft AppNetworkCounter [63], das eine grafische Oberfläche mit Up- und Downloadwerten pro Anwendung bietet
- Einsatz des Tools Wireshark [64], das eine detaillierte Ansicht der übertragenen Pakete, inkl. möglicher Auswertung auf jedem Open-System-Interconnection(OSI)-Layer ermöglicht

Die Entscheidung fiel auf das kostenlose Tool AppNetworkCounter, da dieses den Anforderungen genügt, zuverlässiger als eine händische Erfassung ist und keine übermäßige Komplexität in den Versuchsaufbau bringt.

4.2.6 Erhebung von Windows-Artefakten

Während der Beweissicherung wird zunächst eine Eins-zu-eins-Kopie auf Bit-Ebene des Datenträgers (virtuelle Festplatte) mittels forensischer Software erstellt. Anschließend erfolgt eine automatisierte Extraktion der relevantesten Artefakte (Triage) aus dem Image durch das Programm *Kroll Artifact Parser and Extractor* (KAPE) [65]. Hierbei wird eine große Zahl forensischer Artefakte aus dem Datenträgerabbild kopiert und für eine spätere Analyse gesichert.

4.3 Forschungsumgebung

In diesem Abschnitt werden die Konfigurationen des Testsystems und das forensische Auswertungssystem (Workstation) beschrieben. Zudem werden die eingesetzten forensischen Tools und die vorbereiteten Testdaten aufgelistet.

4.3.1 Forensische Workstation

Auf der Workstation erfolgt die Verwaltung der Testumgebung und die forensischen Untersuchungen der Artefakte. Es handelt sich hierbei nicht um ein virtuelles, sondern um ein physisches System. Auf diesem ist bereits eine große Zahl an forensischen Tools installiert. Die technischen Daten der Workstation sind im Anhang (Tabelle **15**) dargestellt.

4.3.2 Testsystem

Das virtuelle Testsystem dient als Versuchsumgebung, in der die Nutzeraktionen zur Analyse der SRUM-EDB simuliert werden soll.

Die Installation des Systems erfolgt auf der forensischen Workstation mit dem Programm VMware Workstation 16 Pro. Zunächst wird ein Windows-11-Datenträgerimage von der Microsoft-Webseite [66] heruntergeladen. Anschließend wird die VM mit VMware erstellt, jedoch nicht installiert. Hierbei muss darauf geachtet werden, dass die virtuelle Festplatte als einzelne Datei

(*single File*) und nicht in Multi-Dateien angelegt wird, um eine spätere Beweissicherung zu ermöglichen. Der Grund hierfür ist, dass das forensische Programm AccessData FTK Imager nur mit einer virtuellen Festplatte umgehen kann.

Zusätzlich muss vor dem Installieren eine Einstellungsdatei des virtuellen Systems (.VMX) editiert werden. Hierzu wird händisch ein virtuelles Trusted Platform Module (TPM) hinzugefügt, da dieses eine Voraussetzung für die Windows-11-Installation ist [67, 68].

Während der Installation wird ein Microsoft-Konto erstellt, das für den Emailversand benötigt wird. Die detaillierten Installationsparameter der Umgebung und die technischen Daten sind im Anhang (Tabelle **16** und Tabelle **13**), dargestellt.

Nach dem Abschluss der Installation werden Microsoft-Updates installiert, um das System auf den aktuellen Stand zu bringen (s. Tabelle **13**) [69]. In weiterer Folge werden Analyseprogramme von einem USB-Stick sowie aus dem Microsoft-App-Store heruntergeladen und installiert (s. Bild **8**). Hierbei handelt es sich um die Tools Microsoft SysMon und NirSoft AppNetworkCounter. Abschließend wird ein Sicherungspunkt (nachfolgend als ‚Snapshot' bezeichnet) des Betriebssystems erstellt.

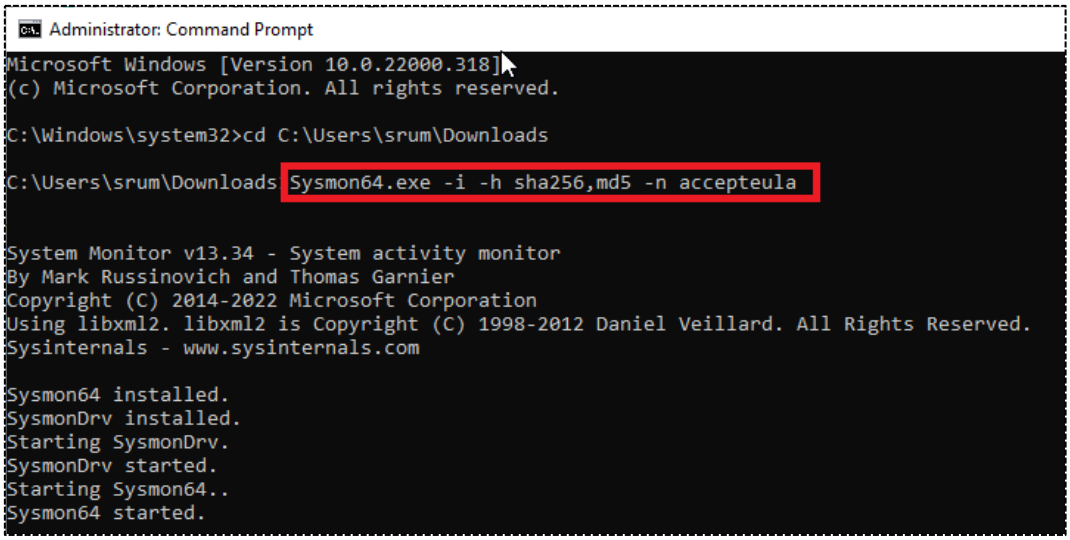

Bild 8: Microsoft-SysMon-Installation

4.3.3 Vorbereitete Testdaten

Für die Simulation der Nutzeraktionen sind Vorbereitungen nötig, welche nachfolgend erläutert werden.

Zunächst wird ein Konto bei Mega.io erstellt. Hierbei handelt es sich um einen Cloud-Speicher-Anbieter, der für den Testfall genutzt werden soll. Zudem wird ein USB-Stick mit Software vorbereitet. Auf diesem werden Musikdateien, eine KeePass-Datenbank, PDF-Dateien und Programme wie Wireshark gespeichert.

4.3.4 Verwendete Forensik-Tools

In diesen Abschnitt werden die Programme aufgelistet, die zur forensischen Erhebung und Aufbereitung genutzt werden. Sie sind in Tabelle **4** dargestellt.

Tabelle 4: Verwendete Programme zur forensischen Untersuchung

Software	Version	Zweck
(Exterro) AccessData FTK Imager [70, 71]	4.7.1.2	Erzeugung eines Datenträgerabbilds
Arsenal Image Mounter Professional [72]	3.6.188	Mounten von Datenträgerabbildern
Eric Zimmerman (EZ) EvtxECmd [73]	1.5.0.0	Event-Log-Parser
EZ PECmd [74]	1.5.0.0	PF-Parser
EZ Registry Explorer [75]	2.0.0.0	Registry-Hive-Betrachtung
EZ SrumECmd [58]	0.5.1.0	SRUM-Parser
EZ TimeApp [76]	1.0	Anzeige der aktuellen Uhrzeit
EZ WxTCmd [77]	1.0.0.0	ActivitiesCache-Parser
HHD Software Hex Editor Neo Professional Edition [78]	7.01.00.7839	HEX-Editor
Kroll Artifact Parser and Extractor [65]	1.2.0.0	Artefaktextraktion
Mark Baggett srum_dump2 [59]	2.4 Kayak	SRUM-Parser
Mark Baggett Srum Template [79]	2.0	Template für srum_dump2

Microsoft Esentutl	10.0	Betrachtung und Reparatur von EDB
Microsoft Excel 365 MSO	16.0.15330.20260	Betrachtung von Dateien
Microsoft SysMon [60]	13.34	Anwendungslogging
NirSoft AppNetworkCounter [63]	1.51	Erfassung von Netzwerkdatenverkehr
NirSoft ESEDatabaseView [80]	1.70	Betrachtung von EDB

5 Durchführung

In diesem Kapitel werden die Dokumentation der Voranalyse, die Durchführung der Nutzersimulation, das Erzeugen von Systemzuständen und die anschließende Beweissicherung dargestellt.

5.1 Vorgehensweise bei der Durchführung der Voranalyse

Das Ziel der Voranalyse ist die Identifikation möglicher Fehler und Probleme, die die Durchführung der Testung oder die anschließende Beweissicherung be- oder verhindern könnten.

Bevor eine Voranalyse stattfindet, muss ein möglicher Ablauf der Testdurchführung und der anschließenden Beweissicherung erarbeitet werden. Gemäß dem Plan sollen nach der Erstellung des Snapshots 1 Nutzeraktionen mittels diverser Software simuliert werden. Anschließend wird das System im eingeschalteten Zustand gesichert, wie in Snapshot 2 dargestellt. Diese Vorgänge sind in Bild **9** dargestellt.

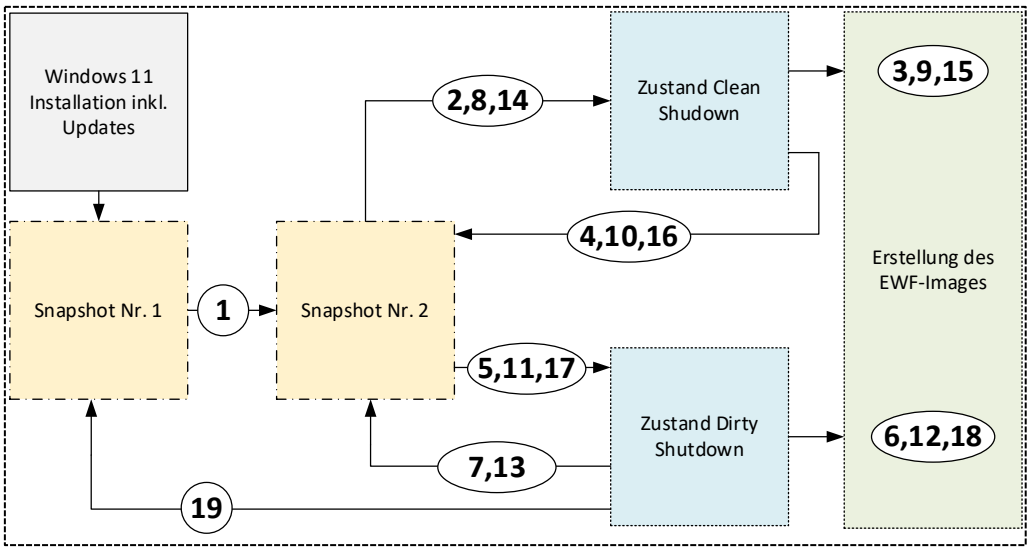

Bild 9: Geplanter Testablauf

Danach werden unterschiedliche Zustände der Testumgebung erzeugt. Diese

sind in Tabelle **5** beschrieben. Dadurch soll zusätzliches Ausgangsmaterial für die Untersuchung der Möglichkeiten und Grenzen von SRUM geliefert werden.

Tabelle 5: Gekürzter Ablaufplan der Probe

Schritt	Handlung
1	Durchführung von Nutzerinteraktionen
2	Sauberes Herunterfahren des Systems via GUI
3, 6, 9, 12, 15, 18	Datenträgerimage erstellen
4, 7, 10, 13, 16	Zurücksetzen der VM zum Snapshot 2
…	…

Als Ergebnis werden sechs Datenträgerimages im Expert Witness Format (EWF) erwartet, die in Kapitel 5.4.2 dargestellt sind. Aus diesen werden anschließend mithilfe des Tools KAPE Artefakte extrahiert. Abschließend wird die SRUDB.dat durch die Programme SrumECMD und srum_dump2 geparst. Die geparsten Ergebnisse werden danach stichprobenartig mit den EDB-Rohdaten der SRUDB.dat verglichen. Anhand der Resultate wird eines der beiden Tools ausgewählt, das anschließend für die weitere Auswertung verwendet wird.

5.2 Erkenntnisse aus der Durchführung

Am 04.07.2022 wurden die Voranalyse der Testumgebung und eine Überprüfung der einzusetzenden Tools durchgeführt. Hierbei wurde nicht der Vorgang aus Tabelle **5** simuliert, sondern lediglich ein EWF-Image erstellt. Jedoch kam es zu einem Problem während der Beweissicherung.

5.2.1 Problem mit mehreren Snapshots

Die virtuelle Festplatte (*Virtual Machine Disk*, VMDK) wurde zwar als einzelne Datei mit dem Namen ‚ssd.vmdk' erstellt, jedoch wurden durch das Anlegen der Snapshots weitere virtuelle Festplatten erzeugt. Diese sind in Bild **10**

veranschaulicht.

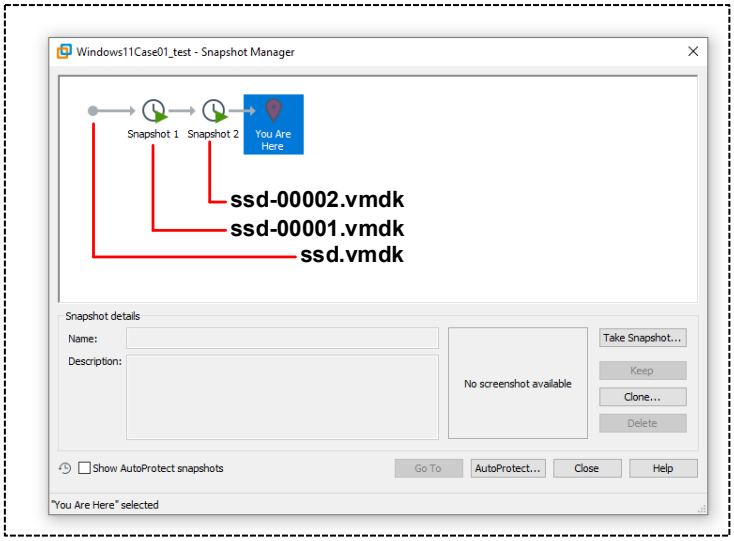

Bild 10: Darstellung der Snapshots in VMware

Um ein Datenträgerimage zu erstellen, mussten zunächst die drei virtuellen Festplatten zu einer einzigen Datei zusammenkopiert werden. Hierzu liefert der Hersteller VMware das Tool vmware-vdiskmanager [81]. Das Zusammenkopieren funktioniert jedoch unter Windows 11 nicht, da die VMDKs verschlüsselt sind [82].

5.2.2 Integritätsproblem der VM

Ein weiterer Lösungsversuch bestand darin, eine Kopie der gesamten VM zu erzeugen. Hierzu wurde die interne VMware-Funktion zum Duplizieren genutzt. Anschließend wurden die Snapshots aus dem kopierten System gelöscht. Hierdurch wurde eine einzelne VMDK erzeugt. Nach der Duplizierung des Systems kam es zu einem Problem mit der Integrität der VM. Das TPM der VM hat den Kopiervorgang bemerkt und beide Systeme – das originale und das duplizierte – verschlüsselt. Ein Starten des Systems oder Verwenden der Festplatte war somit nicht mehr möglich. Aus diesem Grund musste eine komplette Neuinstallation der Testumgebung mit anschließender Installation der Windows-Updates und des Tools SysMon stattfinden.

5.2.3 Lösung zur Beweissicherung

Nach der erfolgreichen Neuinstallation der Testumgebung, wurde der komplette Ordner auf der Workstation der VM kopiert und auf einem externen Datenträger als Backup gesichert (Zustand 0, s. Bild **11**). Danach wurde eine weitere Kopie der Testumgebung angefertigt (Zustand 1, s. Bild **11**). Zustand 1 dient für die nachfolgenden Untersuchungen als Ausgangspunkt und muss in jedem Testfall auf der Workstation erneut erstellt werden. Hierzu wird der Zustand 0 dupliziert. Es werden durch diese Vorgehensweise kein Snapshot erzeugt, daher ist das Erstellen des Datenträgerabbildes fehlerfrei möglich.

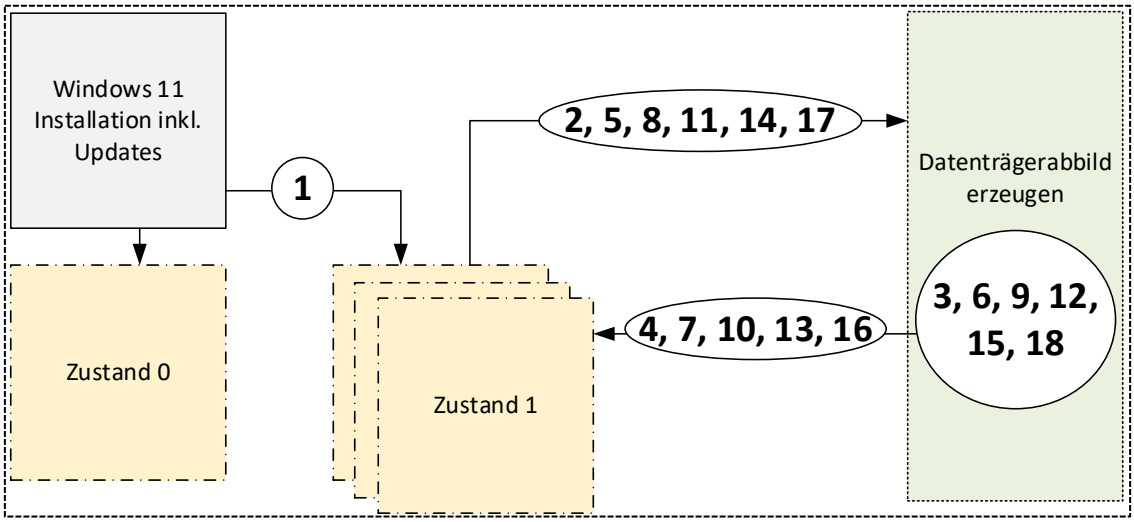

Bild 11: Ablaufplan der Testdurchführung

Der überarbeitete Ablaufplan ist gekürzt in Tabelle **6** sowie vollständig im Anhang (Tabelle **17**) dargestellt.

Tabelle 6: Gekürzte Version des finalen Ablaufplans

Schritt	Handlung
1	Durchführung von Nutzerinteraktionen/Duplizierung des Ordners auf der Workstation /Duplikat als Ausgangspunkt (Zustand 1)
2	Duplizierung von Zustand 1/sauberes Herunterfahren des Systems via GUI
3, 6, 9, 12, 15, 18	Erstellung des EWF-Datenträgerimage
4, 7, 10, 13, 16	Duplizierung von Zustand 1

5.2.4 Verifikation der SRUM-Parser

Nachdem ein Test-Datenträgerabbild erzeugt wurde, konnte mithilfe des KAPE eine Extraktion der SRUDB.dat und weiterer Artefakte erfolgen. Die Artefakte lagen in einer gepackten Datei vor. Diese wurden entpackt und anschließend mit dem SRUM-Parser SrumECmd von Eric Zimmerman überprüft. Hierbei wurde festgestellt, dass der Parser keine Windows-11-EDBs unter Windows 10 parsen kann, da eine andere JET-Version genutzt wird. Somit ist das Tool für die spätere Auswertung unbrauchbar. Dieser Fehler ist in Bild **12** illustriert.

```
PS C:\KAPE\Modules\bin> .\SrumECmd.exe -d "E:\WorkCopy\SmokeTest\" --csv .\Test1
SrumECmd version 0.5.1.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/Srum

Command line: -d E:\WorkCopy\SmokeTest\ --csv .\Test1

Found SRUM database file 'E:\WorkCopy\SmokeTest\SRU\SRUDB.dat'!
Found SOFTWARE hive 'E:\WorkCopy\SmokeTest\config\SOFTWARE'!

Processing 'E:\WorkCopy\SmokeTest\SRU\SRUDB.dat'...
Error processing file! Message: Version of log file is not compatible with Jet version.

This almost always means the database is dirty and must be repaired. This can be verified by running 'esentutl.exe /mh S
RUDB.dat' and examining the 'State' property
Microsoft.Isam.Esent.Interop.EsentBadLogVersionException: Version of log file is not compatible with Jet version
   at Microsoft.Isam.Esent.Interop.Api.Check(Int32 err) in D:\ManagedEsent\EsentInterop\Api.cs:line 3003
   at Microsoft.Isam.Esent.Interop.Api.JetInit2(JET_INSTANCE& instance, InitGrbit grbit) in D:\ManagedEsent\EsentInterop
\Api.cs:line 150
   at Microsoft.Isam.Esent.Interop.Instance.Init(InitGrbit grbit) in D:\ManagedEsent\EsentInterop\Instance.cs:line 222
   at SrumData.Srum..ctor(String fileName, String softwareHive) in D:\Code\Srum\SrumData\Srum.cs:line 1159
   at SrumECmd.Program.DoWork(String f, String r, String d, String csv, String dt, Boolean debug, Boolean trace)
```

Bild 12: SRUM-Parser SrumECmd (Zimmerman)

Als zweites Programm wurde Mark Baggetts srum_dump2 verifiziert. Es verfügt über eine GUI und benötigt neben dem Pfad der SRUM-EDB jenen zum Registry-Hive Software und zu einem leeren SRUM-Comma-separated-Values(CSV)-Template.

Nach dem Parsen wurden die Spaltenüberschriften und Tabellenblätter der CSV-Datei mit den Rohdaten der EDB verglichen. Hierbei wurden die Tools NirSoft ESEDatabaseView, Microsoft Excel 365 und EZ Registry Explorer verwendet. Der Abgleich ist im Anhang (Bild **22**) dargestellt.

Durch den Vergleich wurde festgestellt, dass keine neuen GUIDs unter Windows 11 hinzugekommen sind, jedoch die Tagged-Energy-Provider-Tabelle fehlt. Aufgefallen ist zudem, dass die geparste CSV-Datei ein zusätzliches leeres Tabellenblatt mit der Bezeichnung ‚ruDbCheckpoint‘ enthält. Anhand einer

Überprüfung durch das Parsen weiterer SRUM-EDBs aus dem Internet [83] wurde gezeigt, dass der ruDbCheckpoint-Tabellenreiter in Windows 10 und 11 durch das Programm srum_dump2 angelegt wird. Daher wird dieses Tabellenblatt in der späteren Auswertung nicht weiter beachtet. Eine ausführliche Darstellung der Tabellenbezeichnungen befindet sich im Anhang (Tabelle **18**).

Nachdem die Tabellenbezeichnungen und die Spaltenüberschriften geprüft wurden, fand eine stichprobenartige Kontrolle der Tabellenwerte statt. Ein Beispiel ist in Bild **13** dargestellt.

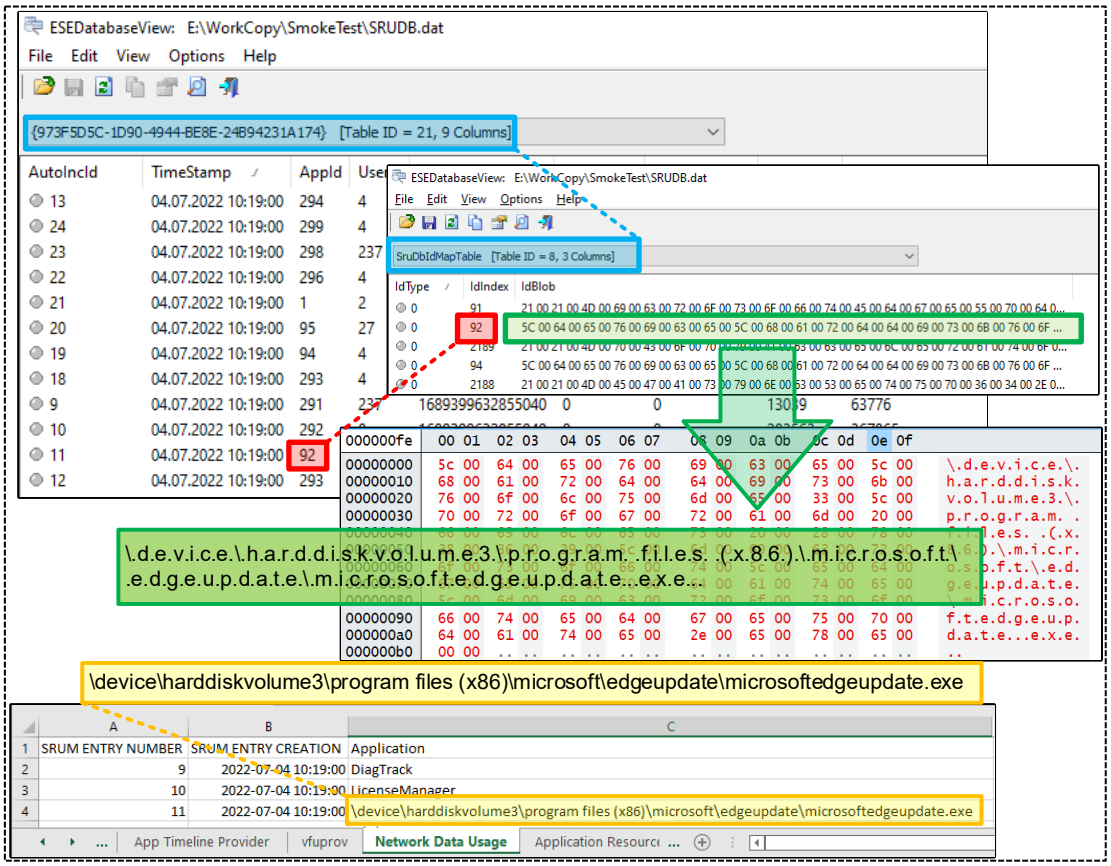

Bild 13: Überprüfung der geparsten SRUM-Werte

Hierbei wurde eine zufällige Zeile im Tabellenblatt ‚Windows Network Data Usage Monitor' (GUID 973F5D5C-1D90-4944-BE8E-24B94231A174) ausgewählt. Diese beinhaltet den AppID-Wert 92, der ebenfalls im Tabellenblatt ‚SruDbIdMapTable' zu finden ist. In der Tabelle ‚SruDbIdMapTable' liegt der Anwendungspfad und -name als Binary Large Object (BLOB) [84] vor. Eine Konvertierung des BLOB mithilfe des Hex-Editors ergab den lesbaren

Anwendungspfad und -namen, der mit dem geparsten Ergebnis übereinstimmt.

Bei einer Überprüfung des Tools srum_dump2 wurden keine Auffälligkeiten oder falsche Werte festgestellt. Daher wurde es für die spätere Auswertung als geeignet betrachtet.

5.3 Fazit zur Voranalyse

Durch die Voranalyse wurden Fehler in der Verwaltung der Testumgebung aufgezeigt, die behoben werden konnten. Zunächst konnte kein Datenträgerabbild erstellt werden. Es wurde jedoch eine Lösung für das Problem gefunden. Zudem wurde das Programm SrumECmd (Zimmerman) als ungeeignet, das Programm srum_dump2 für geeignet, befunden.

5.4 Durchführung der Nutzersimulation

Nach der Durchführung der Voranalyse wurden Testdaten mithilfe einer Nutzersimulation erzeugt. Hierzu erfolgte am 04.07.2022 in der Zeit von 14:09 bis 20:07 Uhr (MESZ) ein sechsstündiger Betrieb der Windows-11-VM mit mehreren Neustarts.

5.4.1 Ablauf der Aktionen

Der Ablauf der Aktionen wurde zunächst manuell erfasst und während der späteren Auswertung mit Daten aus dem Windows-Event-Log abgeglichen. Er ist vollständig im Anhang (Tabelle **19**) dargestellt.

Während der Nutzersimulation wurden folgende Aktionen durchgeführt:

- Installation von Anwendungen
- Öffnen von Anwendungen
- Speichern einer Textdatei
- Starten des Microsoft-Edge-Browsers und Herunterladen einer

Anwendung

- Hören von Musik
- Versuch, eine Anwendung mehrfach parallel zu starten
- Aufruf der Kommandozeile (CMD) und Absetzen von Befehlen
- Aufruf der CMD und Starten einer Software
- Verbinden mit einem WLAN-Hotspot
- Öffnen von PDF-Dokumenten
- Anschließen eines USB-Sticks
- Installation portabler Software auf dem USB-Stick
- Kopieren portabler Software auf eine Windows Partition mit anschließendem Ausführen
- Herunterladen von Dateien via Cloudsoftware
- Herunterladen einer Anwendung aus dem Microsoft Store
- Nutzung von Microsoft-Store-Software, um Videos zu betrachten

Der Status des Systems nach der durchgeführten Nutzersimulation gilt als Ausgangspunkt, für weitere Untersuchungen. Es wurde in diesem Zustand, wie in Kapitel 5.2.3 beschrieben, dupliziert und auf einem externen Datenträger gesichert.

5.4.2 Erzeugung von Systemzuständen

Ausgehend vom Systemstatus 1 (s. Bild **14**) wurden sechs unterschiedliche Zustände erzeugt, aus denen jeweils ein Datenträgerimage erstellt werden konnte.

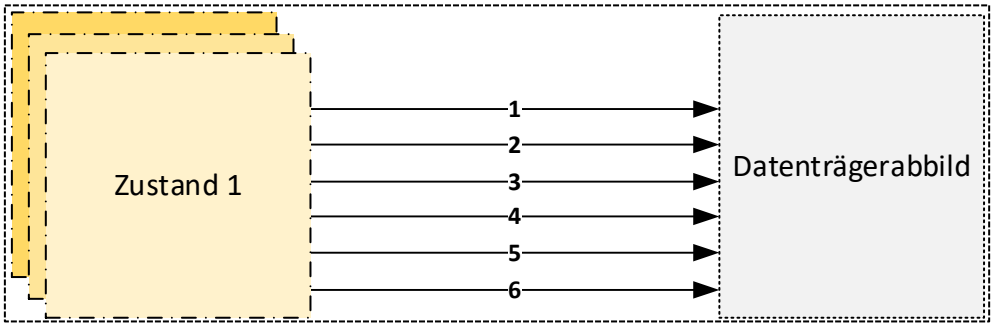

Bild 14: Generierung unterschiedlicher SRUM-Zustände

Die sechs Zustände unterscheiden sich hinsichtlich der Aktionen, die nach der Simulation der Nutzerinteraktionen durchgeführt wurden. Diese Handlungen sind in Tabelle **7** erläutert.

Tabelle 7: Umgebungszustände und deren Bezeichnung im Testfall

Nr.	Handlung	Bezeichnung
1	Herunterfahren des Systems via VMware (unsauberer Shutdown)	Case01_01
2	Sauberes Herunterfahren des Systems via GUI	Case01_02
3	Löschen der SRUM-Historie im Taskmanager/Sauberes Herunterfahren des Systems via GUI	Case01_03
4	Löschen der SRUM-Historie im Taskmanager/Herunterfahren des Systems via VMware (unsauberer Shutdown)	Case01_04
5	Ausführung des CCleaner/Sauberes Herunterfahren des Systems via GUI	Case01_05
6	Ausführung des CCleaner/Herunterfahren des Systems via VMware (unsauberer Shutdown)	Case01_06

5.5 Beweissicherung

Die Beweissicherung erfolgte in allen sechs Fällen nach dem in Bild **15** dargestellten Prinzip. Als Ausgangspunkt diente das jeweils erstellte Datenträgerabbild. Dieses wurde mittels der Hashes im FTK-Log und des Programms Microsoft Certutil abgeglichen. Anschließend wurde eine Arbeitskopie des Datenträgerabbilds angefertigt. Diese wurde wiederum anhand eines Hash-Vergleichs überprüft. Die Arbeitskopie wurde in weiterer Folge mit dem Tool Arsenal Image Mounter Professional in der Workstation gemountet. Danach wurde die grafische Version von KAPE (gKape) genutzt, um Artefakte aus dem Datenträgerimage zu extrahieren. Diese wurden aus der erstellten KAPE-ZIP-Datei entpackt und mithilfe von srum_dump2 geparst. Abschließend wurde aus der erstellten geparsten CSV-Datei eine Arbeitskopie erstellt.

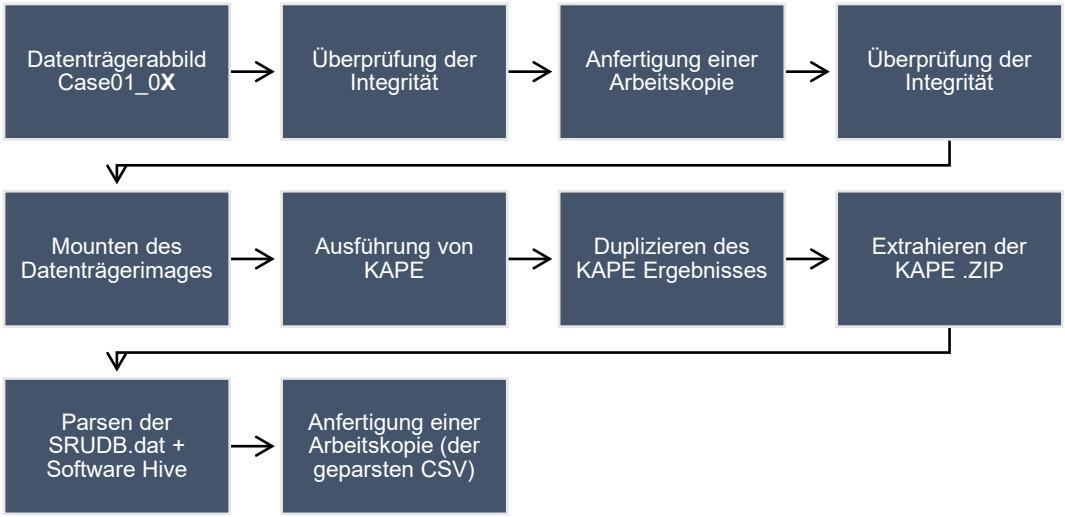

Bild 15: Ablauf der Beweissicherung

Neben der beschriebenen Beweissicherung wurden einmalig aus dem Image Case01_02 zwei Windows-Event-Logs extrahiert. Hierbei handelt es sich um das SysMon- und das System-Event-Log, die für den Zeitenabgleich der Anwendungsausführung nötig sind.

6 Auswertung der Ergebnisse

In diesem Kapitel wird die Auswertung der geparsten Windows SRUM-Daten dargestellt. Zunächst erfolgt eine grundlegende Analyse der Systemzeiten. Danach finden drei beispielhafte Auswertungen für die Anwendungen KeePass, Wireshark und Mega.io statt. Abschließend wird eine Bewertung von SRUM als forensisches Artefakt durchgeführt.

6.1 Zeitauswertung

In diesem Abschnitt wird der Abgleich der Protokollzeiten dargelegt. Zudem wird der Prozess zur Ermittlung der Start- und Stoppzeiten des Systems erläutert.

6.1.1 Abgleich der Protokollzeiten

Die Ausführungszeitpunkte der jeweiligen Anwendungen wurden händisch festgehalten. Aufgrund der menschlichen Beobachtung kann, die im manuellen Protokoll erfassten Zeiten von der tatsächlichen Dauer abweichen. Um dies zu verhindern, wurde während der Installation der Versuchsumgebung der Microsoft-Service SysMon installiert. Zu Beginn der Auswertungsphase wurde das Protokoll mit den Daten aus dem Windows-SysMon-Event-Log zusammengeführt (s. Tabelle **8**).

Tabelle 8: Windows-Event-Log mit Pfadangabe und FilterID

SysMon-Event-Log – Start- und Stoppzeiten von Anwendungen (ID 1, 5)
KAPE\Windows\system32\winevt\logs\Microsoft-Windows-Sysmon%4Operational.evtx
System-Event-Log – Start- und Stoppzeiten des Betriebssystems (ID 6005, 6006)
KAPE\Windows\system32\winevt\logs\System.evtx

Dabei wurde das Event-Log mit dem Tool EvtxECmd in eine CSV-Datei geparst und mit den händischen Zeiten abgeglichen. Die detaillierte Darstellung befindet

sich im Anhang (Tabelle **19**). Während des Abgleichs mussten die Zeitunterschiede der jeweiligen Logdaten beachtet werden. So wurden die Zeiten im Event-Log in UTC, im händischen Protokoll hingegen in MESZ erfasst.

6.1.2 Start- und Stoppzeiten des Systems

Um eine Übersicht über den Zeitpunkt des Schreibens der SRUM-EDB zu erhalten, wurden das Event-Log und der Network Connectivity Usage Provider ausgewertet. Zunächst wurde das System-Event-Log geparst und analysiert. Hierbei wurde eine detaillierte Übersicht der Start- und Stoppzeiten des Systems erstellt, die im Anhang (Tabelle **20**) zu sehen ist. Anschließend wurde der Network Connectivity Usage Provider betrachtet. Dieser gibt Auskunft über verbundene Netzwerke und kann für die Ermittlung der Start- und Stoppzeiten genutzt werden. In Bild **16** ist ein Auszug aus dem Network Connectivity Usage Provider dargestellt. Die Bedeutung der Spalten ist im Folgenden erläutert:

- SRUM Entry Creation: Erstellung des SRUM-Eintrags (UTC)
- InterfaceLuid: Netzwerktyp [48, 52]
- L2ProfileID: Netzwerkname/Name verbundener SSIDs
- ConnectedTime: Verbindungsdauer
- ConnectStartTime: Start der Netzwerkverbindung (UTC)

SRUM Entry Creation	InterfaceLuid	L2ProfileId	ConnectedTime	ConnectStartTime
2022-07-04 13:10:00	IF_TYPE_ETHERNET_CSMACD		00 01:00:08	2022-07-04 12:09:51
2022-07-04 14:02:00	IF_TYPE_ETHERNET_CSMACD		00 01:52:08	2022-07-04 12:09:51
2022-07-04 15:23:00	IF_TYPE_IEEE80211		00 00:00:00	2022-07-04 15:00:29
2022-07-04 15:23:00	IF_TYPE_IEEE80211	BCH	00 00:01:52	2022-07-04 15:00:30
2022-07-04 15:23:00	IF_TYPE_IEEE80211		00 00:00:00	2022-07-04 15:03:30
2022-07-04 15:23:00	IF_TYPE_IEEE80211	BCH	00 00:01:18	2022-07-04 15:03:30
2022-07-04 15:23:00	IF_TYPE_IEEE80211		00 00:00:00	2022-07-04 15:06:56
2022-07-04 15:23:00	IF_TYPE_IEEE80211	Galaxy A52F0C1	00 00:16:03	2022-07-04 15:06:56
2022-07-04 15:24:00	IF_TYPE_IEEE80211		00 00:00:00	2022-07-04 15:23:50
2022-07-04 15:24:00	IF_TYPE_IEEE80211	Galaxy A52F0C1	00 00:00:09	2022-07-04 15:23:50
2022-07-04 15:26:00	IF_TYPE_IEEE80211		00 00:00:00	2022-07-04 15:24:59
2022-07-04 15:26:00	IF_TYPE_IEEE80211	Galaxy A52F0C1	00 00:01:00	2022-07-04 15:24:59
2022-07-04 15:28:00	IF_TYPE_IEEE80211	Galaxy A52F0C1	00 00:01:07	2022-07-04 15:26:52
2022-07-04 15:39:00	IF_TYPE_ETHERNET_CSMACD		00 00:07:30	2022-07-04 15:31:29
2022-07-04 16:41:00	IF_TYPE_ETHERNET_CSMACD		00 01:00:28	2022-07-04 15:40:31
2022-07-04 17:42:00	IF_TYPE_ETHERNET_CSMACD		00 02:01:28	2022-07-04 15:40:31
2022-07-04 18:06:00	IF_TYPE_ETHERNET_CSMACD		00 02:25:28	2022-07-04 15:40:31

Bild 16: Network Connectivity Usage Provider

Die gekürzte Auswertung des Providers ist in Tabelle **9** dargestellt. Hierbei wurde ermittelt, dass im gesamten Testlauf zuverlässig Ergebnisse durch SRUM geschrieben wurden. Die durchschnittliche Abweichung beträgt ca. 1 min. Jedoch besteht eine Ausnahme, bei der eine maximale Differenz von 04:46 min vorliegt.

Tabelle 9: Abgleich der Ein- und Ausschaltzeiten

Bezeichner	Uhrzeit UTC	Event-Log-Zeit UTC	Differenz	Zustand
ConnectStartTime	12:09:51	12:09:46	+00:00:05	Einschalten
ConnectedTime	01:00:08			
errechneter Zeitpunkt	13:09:59			
SRUM Entry Creation	13:10:00	-	+00:00:14	1h Betrieb
ConnectStartTime	12:09:51			
ConnectedTime	01:52:08			
errechneter Zeitpunkt	14:01:59			
SRUM Entry Creation	14:02:00	14:02:49	-00:00:49	Ausschalten
ConnectStartTime	15:00:29	14:55:43	+00:04:46	Einschalten
ConnectedTime	00:00:00			
errechneter Zeitpunkt	15:00:29			
SRUM Entry Creation	15:23:00	15:23:20	+00:00:20	Ausschalten
…	…	…		…

6.2 Detailauswertung am Beispiel KeePass

In diesem Abschnitt wird die Auswertung der Anwendung KeePass schrittweise erläutert. Hierbei sollen das Vorgehen während der Analyse sowie die daraus resultierenden Erkenntnisse dargestellt werden. Zunächst wird die Auswahl der Datenquellen mit der anschließenden Auswertung dargelegt. Danach wird eine Annahme über den Anwendungsverlauf getroffen, welche mit dem tatsächlichen

Verlauf abgeglichen wurde. Abschließend findet eine Erkenntnisdarlegung statt, welche auf Basis der Leitfragen den forensischen Informationsgehalt bestimmt.

6.2.1 Datensammlung

Um einen unvoreingenommenen Blick zu haben, wurde entschieden, die SysMon-Anwendungszeiten vor der Auswertung nicht zu betrachten. Bevor eine Annahme über den Anwendungsverlauf von KeePass getroffen werden konnte, mussten die benötigten Datenquellen (SRUM-Provider) identifiziert und zusammengetragen werden. Zunächst wurden sämtliche geparsten Tabellen (s. Tabelle **18**) des Case01_02 (s. Tabelle **7**) betrachtet. Hierbei fiel auf, dass alle Tabellen über eine Spalte mit der Bezeichnung ‚Anwendung' verfügen.

In jeder Tabelle wurde nach dem Anwendungsnamen ‚KeePass' gefiltert. Dieser Vorgang ist in Bild **17** am Beispiel des App Timeline Providers dargestellt.

A	B	C	D	E	F
Srum	Srum Entry Creation	Application	User SID	BinaryDat	EndTime
2922	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.tmp!2022/04/14:16:10:23!0!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17563906	2022-07-04 15:43:00
2974	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.tmp!2022/04/14:16:10:23!0!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17563650	2022-07-04 15:43:00
3029	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.exe!2022/04/14:16:10:23!43434e!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17563650	2022-07-04 15:43:00
3032	2022-07-04 16:41:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17825794	2022-07-04 16:41:00
3033	2022-07-04 16:41:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17563906	2022-07-04 15:46:00
3209	2022-07-04 17:42:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17825794	2022-07-04 17:42:00
3466	2022-07-04 18:06:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	S-1-5-21-2451268110-1258480682-3566770918-1001 (srum)	17760258	2022-07-04 18:06:00

Bild 17: App Timeline Provider gefiltert nach KeePass

Die identifizierten Ergebnisse wurden anschließend zur deutlicheren Übersicht in einer gesonderten Tabelle zusammengeführt.

Ein Nachweis über die Anwendung KeePass befindet sich in den folgenden Providertabellen:

- App Timeline Provider
- Application Resource Usage Provider
- Energy Estimation Provider

6.2.2 Annahme über den Anwendungsverlauf von KeePass

Nach der Identifikation der Provider wurden die einzelnen Spalten betrachtet.

Diese wurden anhand der Vermutung ausgewählt, dass mit ihnen der Anwendungsverlauf rekonstruiert werden kann.

Bei der Betrachtung der Spaltenüberschriften zeigte sich, dass redundante Einträge vorliegen. Hierbei handelt es sich um die folgenden:

- SRUM Entry Creation
- Application
- User SID

Um eine zeitliche Ordnung vorzunehmen, wurden alle drei Providertabellen zusammengeführt (s. Bild **18**).

A	B	C	D	E	F	G
Herkunft	Srum Entry Creation	Application	EndTime	DurationMS	KeyboardInputS	MouseInputS
ATP	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.tmp!2022/04/14:16:10:23!0!	2022-07-04 15:43:00	179997	707406378	14
ATP	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.tmp!2022/04/14:16:10:23!0!	2022-07-04 15:43:00	239988	707406378	707406378
ATP	2022-07-04 16:41:00	!!KeePass-2.51.1-Setup.exe!2022/04/14:16:10:23!43434e!	2022-07-04 15:43:00	119994	707406378	707406378
ATP	2022-07-04 16:41:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	2022-07-04 16:41:00	3540008	707406378	707406378
ATP	2022-07-04 16:41:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	2022-07-04 15:46:00	180101	8	24
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Users\srum\Downloads\KeePass-2.51.1-Setup.exe				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\ShInstUtil.exe				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-1V7TE.tmp\KeePass-2.51.1-Setup.tmp				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-0BUQ4.tmp\KeePass-2.51.1-Setup.tmp				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-1V7TE.tmp\KeePass-2.51.1-Setup.tmp				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-0BUQ4.tmp\KeePass-2.51.1-Setup.tmp				
EEP	2022-07-04 16:41:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
ARU	2022-07-04 16:42:00	\Device\HarddiskVolume3\Users\srum\Downloads\KeePass-2.51.1-Setup.exe				
ARU	2022-07-04 16:42:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-0BUQ4.tmp\KeePass-2.51.1-Setup.tmp				
ARU	2022-07-04 16:42:00	\Device\HarddiskVolume3\Users\srum\AppData\Local\Temp\is-1V7TE.tmp\KeePass-2.51.1-Setup.tmp				
ARU	2022-07-04 16:42:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\ShInstUtil.exe				
ARU	2022-07-04 16:42:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
ATP	2022-07-04 17:42:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	2022-07-04 17:42:00	3660001	707406378	707406378
EEP	2022-07-04 17:42:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
ARU	2022-07-04 17:43:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
ATP	2022-07-04 18:06:00	!!KeePass.exe!2022/05/09:07:49:29!313cc9!	2022-07-04 18:06:00	1440005	707406378	707406378
EEP	2022-07-04 18:06:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				
ARU	2022-07-04 18:07:00	\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe				

Bild 18: Tabelle über die Anwendungsausführung von Keepass

Anschließend fand eine Sortierung der Spalte *Srum Entry Creation*, sowie die Auswertung der Spalteninhalte statt. Es zeigte sich, dass der SID des Benutzerkontos bei allen Einträgen übereinstimmt. Diese konnte dem Benutzerkonto *srum* zugeordnet werden.

Die Spalte ‚Application' beinhaltet in allen Zeilen das Stichwort ‚KeePass', jedoch in unterschiedlichen Ausprägungen (s. Tabelle **10**). So zeigt der App Timeline Provider den Dateinamen und die -endung sowie weitere Daten, die nicht zugeordnet werden konnten. Der Application Resource Usage Provider und der Energy Estimation Provider beinhalten den korrekten Dateipfad der ausgeführten

Anwendung inklusive der Dateiendung.

Tabelle 10: Vergleich des Dateinamens und -pfades von KeePass

Anwendungsname im App Timeline Provider
!!KeePass.exe!2022/05/09:07:49:29!313cc9!
Anwendungsname im Application Resource Usage Provider und Energy Estimation Provider
\Device\HarddiskVolume3\Program Files\KeePass Password Safe 2\KeePass.exe

Die Auswertung weiterer Tabellenspalten ergab, dass die Spalte ‚EndTime' aus dem App Timeline Provider vor dem SRUM-Entry-Creation-Zeitpunkt liegt. Daher wurde der jeweilige Wert der Spalte ‚EndTime' als Startzeit für die Rekonstruktion des Anwendungsverlaufs verwendet. In den Spalten ‚KeyboardInputS' und ‚MouseInputS' ist zudem der Default-Wert 707406378 angegeben. Der Default-Wert weicht in zwei SRUM-Einträgen auffällig ab. Im ersten Fall handelt es sich um die Installation von KeePass, wie anhand der Angabe ‚KeePass-2.51.1-Setup.tmp' zu erkennen ist. Der zweite Eintrag lautet ‚KeePass.exe' und deutet auf das Ausführen von KeePass hin.

Nachdem alle Tabelleneinträge gesichtet, redundante Daten entfernt sowie die Start- und Stoppzeit des Systems ermittelt wurden, fand die Formulierung einer Annahme über den Anwendungsverlauf von KeePass statt. Gemäß dieser wurde die Installation von KeePass um 15:43 Uhr (UTC) fertiggestellt. Ausschlaggebend hierfür ist die Spalte ‚EndTime' des App Timeline Providers mit dem Eintrag der KeePass-Installationsdatei. Anschließend wurde KeePass um 15:46 Uhr (UTC) ausgeführt, wie anhand des Eintrags ‚KeePass.exe' sowie der Keyboard- und MouseInputS-Angaben vermutet wird. Aufgrund der stündlichen Einträge von SRUM besteht der Verdacht, dass KeePass bis zum Herunterfahren des Systems um 18:06:39 Uhr (UTC) aktiv war.

6.2.3 Tatsächlicher Anwendungsverlauf

Um die zuvor getroffenen Annahmen zum Anwendungsverlauf zu überprüfen,

musste ein Abgleich mit den tatsächlichen Zeiten der Nutzung erfolgen. Hierzu wurde das SysMon-Event-Log nach Einträgen zu KeePass gefiltert und durchsucht. Die Überprüfung ergab, dass in der Rekonstruktion die Installationszeit um < 1 min und die Ausführungszeit um < 4 min von der tatsächlichen SysMon-Zeit abwich. Die zuvor getroffene Annahme der Aktionen (Installation und Ausführung), des Dateipfads und des SID des Benutzerkontos war korrekt und ist in Tabelle **11** sowie in Bild **19** dargestellt. Auffällig ist, dass der letzte SRUM-Entry-Creation-Eintrag nach dem Ausschaltzeitpunkt des Systems liegt. Hier liegt die Vermutung nahe, dass SRUM die Minute des Herunterfahrens aufrundet.

Tabelle 11: Anwendungsverlauf von KeePass

(B: Beschreibung, S: SysMon-Uhrzeit, A: Annahmezeit, D: Dateiname u. -pfad)	
B	Installation von KeePass 2.51.1
S	15:41:42–15:42:27 Uhr UTC
A	15:43 Uhr UTC
D	C:\Users\srum\Downloads\KeePass-2.51.1-Setup.exe
B	Ausführen von KeePass
S	15:42:28–18:06:34 Uhr UTC
A	15:46 Uhr UTC
D	C:\Program Files\KeePass Password Safe 2\KeePass.exe

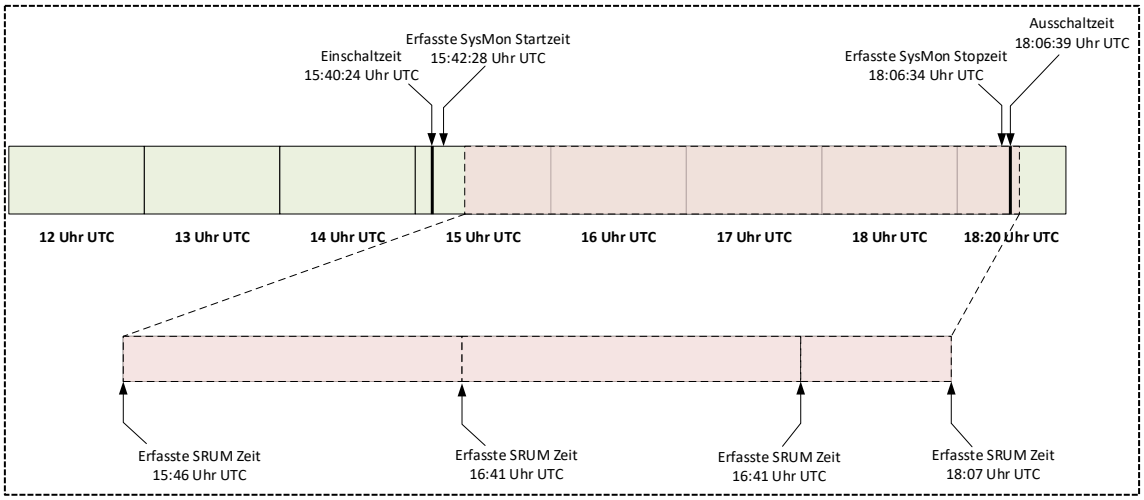

Bild 19: Zeitstrahl des Anwendungsverlaufs von KeePass

6.2.4 Erkenntnisse über KeePass

Die Auswertung von KeePass ergab, dass der App Timeline Provider zuverlässige Zeiten über die Ausführung der Anwendung liefert. Hierzu muss jedoch die Spalte ‚EndTime' anstatt ‚SRUM Entry Creation' betrachtet werden. Darüber hinaus liefern die Spalten ‚KeyboardInputS' und ‚MouseInputS' Anhaltspunkte, dass eine tatsächliche Interaktion mit dem Programm stattgefunden hat. Da keine Einträge zu KeePass in der Network-Data-Usage-Tabelle vorliegen, ist davon auszugehen, dass die Anwendung keine Internetverbindung aufgebaut hat und somit keine Daten nachgeladen wurden.

Die vier Leitfragen aus Tabelle **3** können somit positiv beantwortet werden. Demnach besitzt SRUM als Quelle für den Anwendungsverlauf von KeePass einen hohen Informationsgehalt im forensischen Kontext.

6.3 Teilauswertung von Wireshark und Mega.io

In diesem Abschnitt werden beispielhaft weitere Erkenntnisse aus den Auswertungen von Wireshark und Mega.io dargestellt. Der Fokus der Darlstellung liegt dabei auf die Portabilität von Wireshark und Dateidownloads durch Mega.io, welche es nachzuweisen galt.

6.3.1 Teilauswertung Wireshark

Im Experiment wurde die portable Anwendung Wireshark zunächst auf dem USB-Drive (E:\) gestartet. Anschließend wurde sie nach C:\..\Downloads sowie C:\..\Desktop\ kopiert und ausgeführt.

Spuren der Anwendung wurden in der Energy-Estimation-Provider-, der App-Timeline-Provider-, der Application-Resource-Usage- und der Network-Data-Usage-Tabelle gefunden. Zweitere liefert mit den Werten in der Spalte ‚EndTime' zuverlässige Ergebnisse zur Anwendungsausführung. In der Network-Data-Usage-Tabelle sind der Netzwerktyp (Kabel) sowie die gesendeten und empfangenden Datenmengen erkennbar. Diese Werte stimmen nicht mit den über das Tool NirSoft AppNetworkCounter gemessenen Datenmengen überein. Die Application-Resource-Usage-Tabelle liefert zuverlässig den jeweiligen Pfad der Anwendung, sofern diese durch das lokale System erfolgte. Die Ausführung auf dem USB-Drive konnte nur durch den Energy Estimation Provider nachgewiesen werden. Hierbei wird die C-Partition als ‚Volume3' und die E-Partition als ‚Volume5' bezeichnet. Somit ist davon auszugehen, dass die Volumensbezeichnungen in numerischer Reihenfolge vergeben werden.

6.3.2 Teilauswertung Mega.io

Bei Mega.io handelt es sich um eine Cloud-Synchronisations-Software, die auf der C-Partition installiert wurde. Spuren der Nutzung wurden in der Energy-Estimation-Provider-, der App-Timeline-Provider-, der Application-Resource-Usage- und der Network-Data-Usage-Tabelle gefunden. Da die Anwendung im Autostart des Systems konfiguriert war, wurde sie nach dem Hochfahren automatisch ausgeführt. Die Network-Data-Usage-Tabelle verzeichnete die Nutzung in Abständen von ca. 4 min. Diese Daten liefern einen detaillierten Überblick über die Anwendungsausführung und könnten zudem den Einschaltzeitraum des Systems bestätigen.

Mithilfe der Anwendung wurden Dateien aus dem Mega.io-Cloud-Speicher heruntergeladen. Die durch SRUM erhobene Datenmenge entspricht nicht der

tatsächlich übertragenen, jedoch sind die Zeiten des Up- und Downloads annähernd korrekt. Im Versuch wurden Daten um 15:47 Uhr (UTC) sowie zwischen 15:59 und 18:03 Uhr (UTC) heruntergeladen. Darüber hinaus wurden grafische Aufbereitungen wie in Bild **20** erstellt, um Anwendungen und deren Transferzeitraum von Netzwerkdaten zu identifizieren. Die erfasste Zeit weicht im Fall der Anwendung Mega.io um ca. 3 min von der tatsächlichen Zeit ab. Hierdurch kann bestätigt werden, dass der Network Data Usage Provider zuverlässige Informationen zur Dauer des Netzwerkdatentransfers liefern kann.

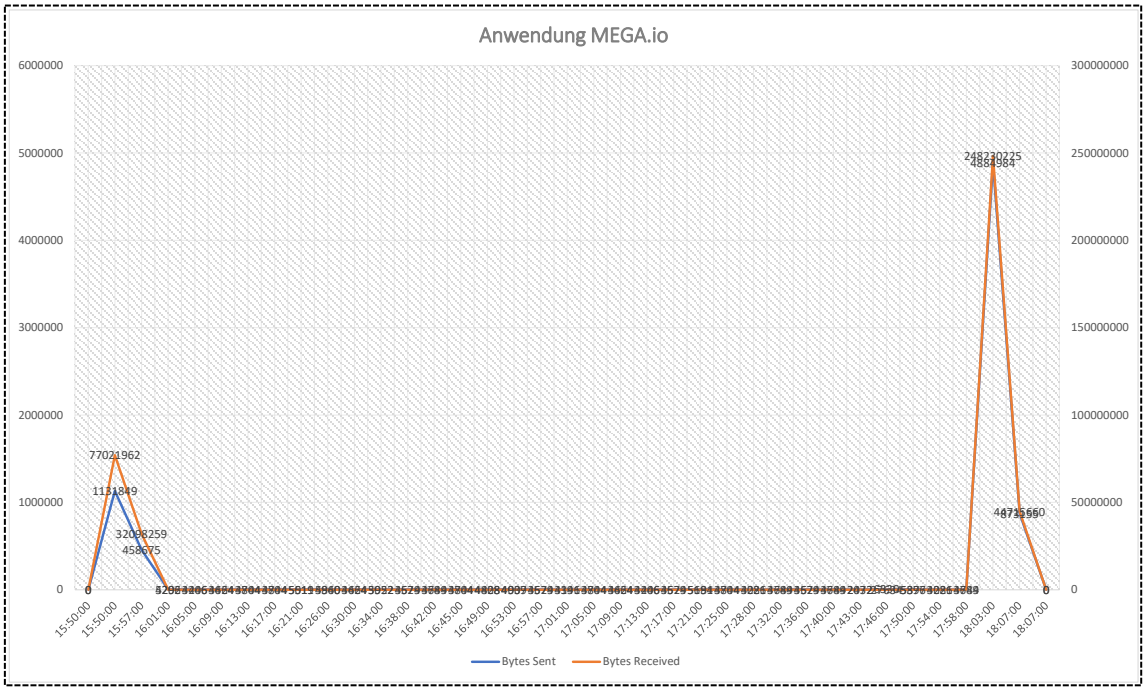

Bild 20: Datennutzung der Anwendung Mega.io

7 Bewertung von SRUM als forensisches Artefakt

In diesen Abschnitt werden gewonnen Erkenntnisse über Windows SRUM zusammentragen. Hierzu erfolgt die Beantwortung der Forschungsfragen aus Kapitel 1, zur Bestimmung des Informationsgehaltes des Programms. Die hier dargelegten Erkenntnisse beruhen auf der Auswertung aller Anwendungen im Testfall. In Kapitel 6 wurde nur eine Teildarstellung der durchgeführten Analysen dargelegt.

7.1.1 Beantwortung der Frage 1

Anhand der ersten Frage sollte ermittelt werden, ob durch SRUM zuverlässige Ergebnisse für die Aufklärung forensischer W-Fragen geliefert werden können. Hierzu wurden vier Leitfragen formuliert, die bei jeder ausgeführten Anwendung gestellt wurden. Die Überprüfung ergab, dass SRUM zuverlässig den Dateinamen und -pfad (‚Wo?‘) erfasst sowie die Zuordnung des Benutzerkontos (‚Wer?‘) ermöglicht. Eine Ausnahme bilden Anwendungen aus dem Microsoft-App-Store. Hier wird der Dateiname, jedoch nicht der Installationspfad erfasst.

Hinsichtlich der Zeit und Dauer der Ausführung (‚Wann?‘) sowie der Häufigkeit der Nutzung (‚Wie oft?‘) wurden unzuverlässige Ergebnisse geliefert. So wird keine Zusammenfassung (Summierung) der Ausführungen einer Anwendung erstellt, sofern diese Erfasst wurde. Jede erneute Verwendung wird einzeln aufgelistet. Längere Nutzungszeiten (> 1 h) werden in SRUM zuverlässig erfasst, sie müssen jedoch korrekt gedeutet werden. Ausführungen von unter 1 min werden teilweise nicht dargestellt. Zudem erfasst SRUM nicht den Startzeitpunkt, sondern die letzte Änderungszeit einer Anwendung. In einem Fall hat der App Timeline Provider die Ausführung der CMD nicht wiedergegeben.

Die genannten Erkenntnisse treffen auf folgende überprüfte Fälle zu:

- Installation und Ausführung von Anwendungen auf der C-Partition (auch nach dem Löschen der Anwendung)

- Ausführung portabler Anwendungen auf der C-Partition (auch nach dem Löschen der Anwendung)
- Ausführung portabler Anwendungen auf einem externen Gerät (auch nach dem Löschen der Anwendung)
- Installation und Ausführung von Anwendungen aus dem Microsoft-App-Store (auch nach dem Löschen der Anwendung)
- Ausführung des Befehls in der Kommandozeile und starten einer Anwendung

7.1.2 Beantwortung der Frage 2

Mit der zweiten Frage sollten weitere Anwendungsmöglichkeiten von SRUM untersucht werden. Dabei sollte festgestellt werden, welche Nutzeraktivitäten nachgewiesen werden können.

Im Test lieferte die geparste SRUM-EDB neun auswertbare Provider. Der Energy Usage Provider und der Energy Usage Provider Long Term beinhalten keine Daten. Dies liegt vermutlich an der Umgebung des Testsystems, da die VM keine Energiedaten aus dem Hostsystem erfassen kann.

Über den Provider VFUProv wurden keine Erkenntnisse gewonnen. Sein Verwendungszweck ist somit weiterhin unbekannt.

Der Network Connectivity Usage Provider liefert zuverlässige Daten über Netzwerkverbindungen. Hierbei wurden der Interfacetyp, der Netzwerkname (sofern das Registry-Hive Software mitgeparst wurde), die Verbindungsdauer und der Startzeitpunkt der Verbindung erfasst. Hieraus können der Ein- und der Ausschaltpunkt des Systems abgeleitet werden. Diese Daten sind relevant, um die Schreibzeiten von SRUM zu ermitteln.

Mithilfe des Energy Estimation Providers, des App Timeline Providers und des Application Resource Usage Providers kann der Ausführungszeitpunkt einer Anwendung eingegrenzt werden. Zudem werden der SID sowie der Dateiname und -pfad zuverlässig erkannt.

Der Network Data Usage Provider liefert den Ausführungszeitpunkt, den Anwendungspfad, den SID sowie Datenmengen, die empfangen oder gesendet wurden. Letztere sind nicht korrekt, jedoch stimmt der Zeitpunkt der Übertragung. Diese Erkenntnis beruht auf der Auswertung aller Anwendungen im Testfall.

Wird der Network Data Usage Provider mit dem Network Connectivity Usage Provider kombiniert, so können genutzte Anwendungen, der Übertragungszeitpunkt und das verbundene Netzwerk ermittelt werden. Ein Beispiel hierfür ist in Bild **21** dargestellt. Hier hat der Microsoft-Edge-Browser größere Datenmengen im Zeitraum von 15:23 bis 15:28 Uhr (UTC) über ein WLAN-Netzwerk empfangen.

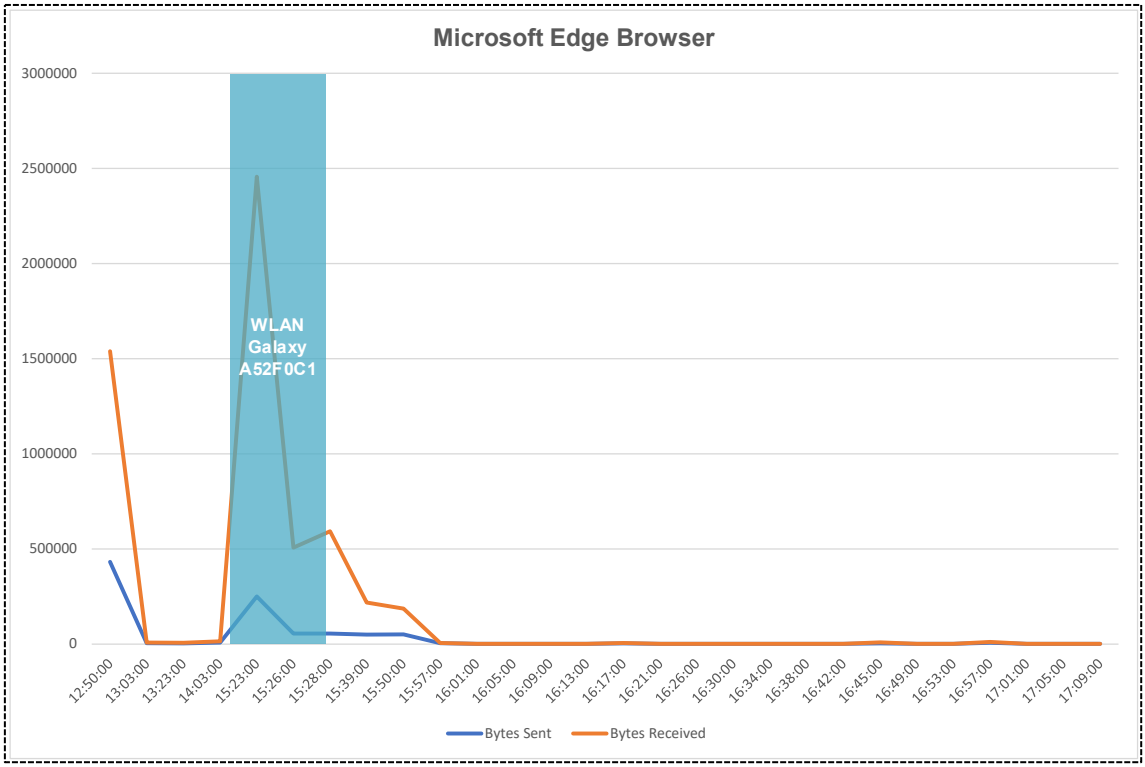

Bild 21: Datenübertragung des Microsoft-Edge-Browser

Der WPN Provider beinhaltet in den Untersuchungsergebnissen nur einen Eintrag. Dieser umfasst einen Verweis auf die Microsoft-App-Store-Anwendung Windows-Kommunikations-Apps (Hx.Outlook) sowie den SID und den Zeitpunkt der Ausführung. Während des Tests wurden Daten via Microsoft Outlook um 15:13 bis 15:15 Uhr (UTC) exfiltriert. Dabei wurde 15:16 Uhr (UTC) als Ausführungszeitpunkt im WPN Provider erfasst. Eine abschließende Bewertung

dieses Providers kann aufgrund der geringen Datenlage nicht durchgeführt werden.

Im Test wurden alle Anwendungen, die eine Internetverbindung genutzt haben, im Network Data Usage Provider erfasst. Eine Ausnahme bildet Microsoft Outlook. Dies deutet auf eine Datenschutzeinstellung aufgrund der DSGVO hin [85].

Zusammengefasst können mithilfe von SRUM folgende Fragen zuverlässig beantwortet werden:

- Wann war ein System eingeschaltet?
- Mit welchem Netzwerk (Name, Typ, Dauer) war das System verbunden?
- Wann wurden große Datenmengen gesendet oder empfangen?
- Welche Anwendung hat große Datenmengen gesendet oder empfangen?
- Welche Anwendung (Name und Dateipfad) wurde wann (zuverlässiger, stundengenauer Wert) und von welchem Nutzerkonto ausgeführt?

7.1.3 Beantwortung der Frage 3

Mithilfe der dritten Frage sollte ermittelt werden, wie zuverlässig SRUM vor Nutzereingriffen bzw. Antiforensik-Maßnahmen geschützt ist. Zur Überprüfung wurden unterschiedliche Antiforensik-Maßnahmen durchgeführt. Die Erkenntnisse werden nachfolgend dargelegt.

Während der Testdurchführung wurden sechs unterschiedliche Zustände des Systems erstellt und gesichert. Diese sind in Kapitel 5.4.2 (Tabelle **7**) aufgeführt. In den Zuständen Case01_01 und Case01_02 wurde der Test ohne anschließende Eingriffe durchgeführt. In zweiterem Fall wurden alle Daten von SRUM erfasst. In Case01_01 wurde das System unsauber heruntergefahren, was Auswirkungen auf die SRUM-EDB hatte. So wurden hier keine Informationen nach dem letzten stündlichen Schreibvorgang um 17:43 Uhr (UTC) erfasst. Eine Ausnahme bildet der Network Data Usage Provider, der in Abständen von ca. 4 min Daten bis zum unsauberen Shutdown ermittelte.

In Case01_03 und Case01_04 wurde die Historie im Taskmanager gelöscht. Anschließend fand jeweils ein sauberer und ein unsauberer Shutdown statt. Die Auswertung der geparsten SRUM-EDB ergab, dass der simulierte Löschvorgang keine Auswirkung auf das Programm hatte. So konnten keine Spuren vernichtet werden. Jedoch musste die SRUM-EDB in Case01_04 aufgrund des unsauberen Shutdowns via Esentutl repariert werden.

In Case01_05 und Case01_06 wurde das Tool CCleaner (Version 6.01) verwendet. Hierbei wurden alle Werte in den Kategorien ‚Windows Explorer', ‚System' und ‚Advanced' aktiviert. Die Auswertung ergab, dass in Case01_05 alle Daten gelöscht wurden. Somit wurde bewiesen, dass durch CCleaner die Integrität der SRUM-EDB beeinflusst werden kann. Der Case01_06 konnte nicht ausgewertet werden, da die SRUDB.dat inklusive der zugehörigen Transaktionslogs gelöscht wurde. Der Grund für die Löschung ist unklar.

7.1.4 Beantwortung der Frage 4

Mit der vierten Frage wurde darauf abgezielt, die Grenzen von SRUM zu erkennen und zu ermitteln, wie diese durch weitere Windows-Artefakte kompensiert werden können.

Eine wesentliche Beschränkung stellt vor allem die unzuverlässige Datenerfassung der Anwendungsausführung dar. So wurden Nutzungen von Programmen im Test nicht immer abgebildet. Zudem ist der exakte Startzeitpunkt einer Anwendung nicht ermittelbar.

Zur Überprüfung möglicher Kompensationen wurde der Fall Case01_06 gewählt. In diesem wurden keine SRUM-Dateien erfasst. Zudem fand ein Eingriff mittels CCleaner statt, der auch andere Windows-Artefakte beeinflusst haben könnte.

Für die mögliche Kompensation der genannten Beschränkungen wurde entschieden, eine Auswertung der ActivitiesCache-Datenbank und der PF-Dateien durchzuführen. Diese Artefakte wurden bereits in Kapitel 2.4 erläutert.

Zum Parsen der PF-Dateien wurde PECmd verwendet. Als Ausgabeformat

wurde eine CSV-Datei erstellt. Deren Überprüfung ergab, dass jede Ausführung der Anwendungen korrekt erfasst wurde. In Tabelle **12** wird veranschaulicht, dass der Nutzungszeitpunkt sowie der Dateipfad von Wireshark und der Microsoft-App-Store-Anwendung HxOutlook erkannt werden. Da PF-Dateien keine Auskunft über die Ausführungsdauer einer Anwendung geben können, ist eine Ergänzung durch SRUM denkbar.

Tabelle 12: PF-Auswertung von Wireshark und Hx.Outlook

Ausführungszeit (UTC)	Dateiname
04.07.2022 15:33	\VOLUME{01d86dc5fb3b49b4-6aff22e8}\WIRESHARK\WIRESHARKPORTABLE64\APP\WIRESHARK\WIRESHARK.EXE
04.07.2022 15:35	\VOLUME{01d88f9efbb24d69-e6fbcf32}\USERS\SRUM\DOWNLOADS\WIRESHARKPORTABLE64\APP\WIRESHARK\WIRESHARK.EXE
04.07.2022 15:37	\VOLUME{01d88f9efbb24d69-e6fbcf32}\USERS\SRUM\DESKTOP\WIRESHARKPORTABLE64\APP\WIRESHARK\WIRESHARK.EXE
04.07.2022 15:07	\VOLUME{01d88f9efbb24d69-e6fbcf32}\PROGRAM FILES\WINDOWSAPPS\MICROSOFT.WINDOWSCOMMUNICATIONSAPPS_16005.14326.20970.0_X64__8WEKYB3D8BBWE\HXOUTLOOK.EXE
04.07.2022 15:13	\VOLUME{01d88f9efbb24d69-e6fbcf32}\PROGRAM FILES\WINDOWSAPPS\MICROSOFT.WINDOWSCOMMUNICATIONSAPPS_16005.14326.20970.0_X64__8WEKYB3D8BBWE\HXOUTLOOK.EXE

Um die ActivitiesCache-Datenbank auszuwerten, war zunächst das Parsen mittels WxTCmd nötig. Das Ausgabeformat ist eine CSV-Datei. Bei deren Auswertung zeigte sich, dass Informationen über den Dateipfad, die Start- und Endzeit der Anwendung, ihre Vordergrundzeit sowie geöffnete Dokumente des Programms vorhanden sind.

Am Beispiel der Anwendung Acrobat konnte nachvollzogen werden, dass der Dateipfad und -name ‚Program Files X64\Adobe\AcrobatDC\Acrobat\Acrobat.exe' lautet. Die Anwendung hat am 04.07.2022 um 18:05 Uhr (UTC) eine PDF-Datei mit dem

Titel ‚Lehr- und Studienbriefe Kriminalistik Kriminologie. Bd. 17 Grundlagen der Kriminaltechnik II by Rabe, Frank Frings, Christoph.pdf' geöffnet. Weitere Untersuchungen der ActivitiesCache-Datenbank ergaben jedoch, dass der Zeitpunkt des Schließens einer Anwendung nicht korrekt ist. Sie erfasst demnach Ausführungen augenscheinlich nur korrekt, solange diese im Vordergrund des Benutzers erfolgt.

Zusammenfassend lässt sich sagen, dass die ActivitiesCache-Datenbank und die PF-Dateien zuverlässig die Anwendungsausführung sowie die korrekte Startzeit erfassen. Der Endzeitpunkt einer Anwendung konnte jedoch nicht ermittelt werden. Zudem wurden keine Spuren gefunden, die Informationen über Netzwerkdaten liefert. So war es nicht möglich, einen Netzwerkendpunkt (IP-Adresse) oder eine transferierte Datei (Dateiname und -typ) zu ermitteln.

8 Zusammenfassung und Ausblick

Ziel dieser Bachelorthesis war es, das Verhalten von SRUM unter Windows 11 zu überprüfen. Hierbei sollten die Möglichkeiten und Grenzen seiner Anwendung sowie die Kompensation der Beschränkungen durch weitere Windows-Artefakte erforscht werden.

Für die Untersuchung der Möglichkeiten von SRUM wurde eine Testumgebung erzeugt sowie passende Analysewerkzeuge ausgewählt. Anschließend wurde eine Voranalyse durchgeführt, durch die Probleme bei der Beweissicherung identifiziert und behoben wurden.

Nachdem eine Lösung für die Probleme gefunden wurde, konnte die eigentliche Erzeugung von Testdaten erfolgen. Hierzu wurden sechs unterschiedliche Zustände der Umgebung erzeugt, die anschließend ausgewertet wurden. Dabei wurden forensische Leitfragen beantwortet, um den Informationsgehalt von SRUM zu bestimmen.

Die Auswertung der sechs Umgebungen ergab, dass SRUM unter Windows 11 keine neuen Provider eingeführt hat. Die bekannten Provider lieferten zuverlässige Ergebnisse zur Aufklärung forensischer Leitfragen. Der Anwendungspfad und das ausführende Benutzerkonto konnten in allen Fällen nachgewiesen werden. Zudem wurde erkannt, dass SRUM, anders als teilweise in der Literatur dargelegt (s. Kapitel 3.3), nicht nur stündlich Daten erfasst, sondern der Network Data Usage Provider im Abstand von ca. 4 min Anwendungen mit einer Netzwerkverbindung speichert.

Eine mögliche Grenze von SRUM stellen zum einen die sporadisch unzuverlässigen Datenerfassungen der Anwendungsausführungen dar. Zum anderen ist der Zeitpunkt des Beendens einer Anwendung nicht ermittelbar. Eine Ausführung des Tools CCleaner zeigte, dass hierdurch Daten in SRUM entfernt werden. Das Löschen der Taskmanager-Historie hat jedoch keinen Einfluss auf das Programm.

Eine Kompensation von SRUM ist durch die ActivitiesCache-Datenbank und PF-Dateien denkbar, da der genaue Startzeitpunkt ermittelt werden kann. Jedoch können auch diese beiden Artefakte den Zeitpunkt des Beendens einer Anwendung nicht zuverlässig ermitteln. Weitere Forschungsarbeiten sind zur Kompensation denkbar.

Zusammenfassend lässt sich sagen, dass SRUM unter Windows 11 als forensisches Artefakt einen Mehrwert bei der Aufklärung von Vorfällen bietet. Durch die Auswertung können forensische Leitfragen beantwortet werden, welche zur Aufklärung eines Vorfalls dienlich sind. Jedoch konnten aus organisatorischen oder technischen Gründen nicht das gesamte Verhalten von SRUM ermittelt werden. In weiteren Forschungsarbeiten wären folgende Untersuchungen denkbar:

- Überprüfung, ob der Network Connectivity Usage Provider im Flugmodus weiterhin Daten schreibt
- Untersuchung, warum SRUM sporadische Anwendungen mit einer kurzen Laufzeit nicht erfasst
- Detailliertere Untersuchung des App Timeline Providers mit Fokus auf Abweichungen der Default-Werte in den Spalten ‚KeyboardInputS‘ und ‚MouseInputS‘

9 Literaturverzeichnis

[1] Bundesamt für Sicherheit in der Informationstechnik, „Leitfaden IT-Forensik: Version 1.0.1“, 2011. [Online]. Verfügbar unter: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf?__blob=publicationFile&v=2. (Zugriff am: 11. Juli 2022).

[2] PONS, Deutsch-Latein Übersetzung. [Online]. Verfügbar unter: https://de.pons.com/%C3%BCbersetzung/deutsch-latein/forensisch?bidir=1#la (Zugriff am: 11. Juli 2022).

[3] D. Heinson, IT-Forensik: Zur Erhebung und Verwertung von Beweisen aus informationstechnischen Systemen. Tübingen: Mohr Siebeck, 2015.

[4] A. Geschonneck, Computer-Forensik: Computerstraftaten erkennen, ermitteln, aufklären. 6. Aufl. Heidelberg: dpunkt.verlag, 2014 (Zugriff am: 08. Juli 2022).

[5] K. Kent, S. Chevalier, T. Grance und H. Dang, „Guide to integrating forensic techniques into incident response: NIST.SP.800-86“, Gaithersburg, MD, 2006. [Online]. Verfügbar unter: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-86.pdf. (Zugriff am: 12. Juli 2022).

[6] R. Adams, V. Hobbs und G. Mann, „The Advanced Data Acquisition Model (Adam): A Process Model for Digital Forensic Practice“, JDFSL, 2013, doi: 10.15394/jdfsl.2013.1154.

[7] J. Sachowski, Hg., Implementing Digital Forensic Readiness: From reactive to proactive process. Cambridge, MA: Syngress, 2016. [Online]. Verfügbar unter: http://www.sciencedirect.com/science/book/9780128044544

[8] J. Sachowski, Digital forensics and investigations: People, processes, and technologies to defend the enterprise. Boca Raton FL: CRC Press Taylor & Francis Group, 2018 (Zugriff am: 28. Juli 2022).

[9] E. Casey, Digital Evidence and Computer Crime: Forensic science, computers and the internet. Waltham, MA: Academic Press, 2011. [Online]. Verfügbar unter: https://www.elsevier.com/books/digital-evidence-and-computer-crime/casey/978-0-08-092148-8 (Zugriff am: 16. Juli 2022).

[10] Hardy Valenthio, Anti-Computer Forensics Technique: Artifact Wiping. [Online]. Verfügbar unter: https://medium.com/@18218004/digital-forensics-blog-01-anti-computer-forensics-technique-artifact-wiping-bfcf8ed25a75 (Zugriff am: 13. Juli 2022).

[11] Kessler, Gary, C., „Anti-Forensics and the Digital Investigator“, 2007.

[12] T. Rochmadi, I. Riadi und Y. Prayudi, „Live Forensics for Anti-Forensics Analysis on Private Portable Web Browser“, IJCA, Jg. 164, Nr. 8, S. 31–37, 2017, doi: 10.5120/ijca2017913717 (Zugriff am: 10. Juli 2022).

[13] K. Conlan, I. Baggili und F. Breitinger, „Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy“, Digital Investigation, Jg. 18, S66-S75, 2016, doi: 10.1016/j.diin.2016.04.006.

[14] Bundesamt für Sicherheit in der Informationstechnik, „Analyse der Telemetriekomponente in Windows 10: Konfigurations- und Protokollierungsempfehlung“, 2020. [Online]. Verfügbar unter: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/SiSyPHus/Analyse_Telemetriekomponente_1_2.pdf?__blob=publicationFile&v=3. (Zugriff am: 9. Juli 2022).

[15] Microsoft Corporation, Konfigurieren von Windows-Diagnosedaten in Ihrer Organisation (Windows 10 und Windows 11) - Windows Privacy. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows/privacy/configure-windows-diagnostic-data-in-your-organization (Zugriff am: 10. Juli 2022).

[16] Microsoft Corporation, Änderungen an der Windows-Diagnosedatensammlung - Windows Privacy. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows/privacy/changes-to-windows-diagnostic-data-collection#new-windows-diagnostic-data-processor-configuration (Zugriff am: 10. Juli 2022).

[17] M. Reuter, „Möglichkeiten zum Einsatz von Event Tracing for Windows (ETW) zur Unterstützung forensischer Analysen von Prozessverhalten in Windows 10“. Master-Thesis, Hochschule Wismar, Wismar, 2020. [Online]. Verfügbar unter: https://it-forensik.fiw.hs-wismar.de/images/a/a3/MT_MReuter.pdf (Zugriff am: 10. Juli 2022).

[18] Microsoft Corporation, Informationen zur Ereignisablaufverfolgung für Treiber - Windows drivers. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows-hardware/drivers/devtest/about-event-tracing-for-drivers (Zugriff am: 10. Juli 2022).

[19] Claudiu Teodorescu und Igor Korkin, „Veni, No Vidi, No Vici: Attacks on ETW Blind EDR Sensors: BlackHat Europe 2021“. [Online]. Verfügbar unter: https://i.blackhat.com/EU-21/Wednesday/EU-21-Teodorescu-Veni-No-Vidi-No-Vici-Attacks-On-ETW-Blind-EDRs.pdf (Zugriff am: 20. Juli 2022).

[20] Y. Khatri, „Forensic implications of System Resource Usage Monitor (SRUM) data in Windows 8“, Digital Investigation, Jg. 12, S. 53–65, 2015, doi: 10.1016/j.diin.2015.01.002 (Zugriff am: 01. Juli 2022)..

[21] Microsoft Corporation, Reliability Infrastructure. [Online]. Verfügbar unter: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/dd393053(v=ws.10) (Zugriff am: 10. Juli 2022).

[22] P. Yosifovich / A. Ionescu / M. E. Russinovich / D. A. Solomon, Hg., Windows Internals Band 1. Heidelberg: Microsoft Press; dpunkt.verlag, 2018. [Online]. Verfügbar unter: https://dpunkt.de/produkt/windows-internals/ (Zugriff am: 07. Juli 2022).

[23] Joachim Metz, Windows Search forensics: Analyzing the Windows (Desktop) Search Extensible Storage Engine database. [Online]. Verfügbar unter: https://github.com/libyal/documentation/blob/main/Forensic%20analysis%20of%20the%20Windows%20Search%20database.pdf (Zugriff am: 8. Juli 2022).

[24] Linda Taylor, ESE Deep Dive: Part 1: The Anatomy of an ESE database - Microsoft Tech Community. [Online]. Verfügbar unter: https://techcommunity.microsoft.com/t5/ask-the-directory-services-team/ese-deep-dive-part-1-the-anatomy-of-an-ese-database/ba-p/400496 (Zugriff am: 1. Juli 2022).

[25] Microsoft Corporation, Extensible Storage Engine: Summary of Active Directory Architecture. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/previous-versions/windows/it-pro/windows-2000-server/cc961824(v=technet.10)?redirectedfrom=MSDN (Zugriff am: 10. Juli 2022).

[26] Microsoft Corporation, Extensible Storage Engine Architecture: Exchange 2007 Help. [Online]. Verfügbar unter: https://docs.microsoft.com/en-us/previous-versions/office/exchange-server-2007/bb310772(v=exchg.80)?redirectedfrom=MSDN (Zugriff am: 10. Juli 2022).

[27] Informix Information Development, „C-ISAM Programmer's Manual, Version 7.2“, 2001. [Online]. Verfügbar unter: https://publib.boulder.ibm.com/epubs/pdf/7897b.pdf. (Zugriff am: 1. Juli 2022).

[28] Wikipedia, Index Sequential Access Method. [Online]. Verfügbar unter: https://de.wikipedia.org/w/index.php?title=Index_Sequential_Access_Method&oldid=219240204 (Zugriff am: 10. Juli 2022).

[29] Microsoft Corporation, Esentutl: Management and Tools / Command-Line Reference. [Online]. Verfügbar unter: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh875546(v=ws.11) (Zugriff am: 10. Juli 2022).

[30] Joachim Metz, Extensible Storage Engine (ESE) Database File (EDB) format specification: libyal/libesedb. [Online]. Verfügbar unter: https://github.com/libyal/libesedb/blob/main/documentation/Extensible%20Storage%20Engine%20(ESE)%20Database%20File%20(EDB)%20format.asciidoc (Zugriff am: 13. Juli 2022).

[31] Libyal, Library and tools to access the Extensible Storage Engine (ESE) Database File (EDB) format: libesedb. [Online]. Verfügbar unter: https://github.com/libyal/libesedb (Zugriff am: 20. Juli 2022).

[32] NIST, „Windows Registry Forensic Tool Specification - Draft 2 of Version 1.0“. [Online]. Verfügbar unter: https://www.nist.gov/system/files/documents/2018/06/28/wrt-spec-v1.0-public-draft-2.pdf (Zugriff am: 22. Juli 2022).

[33] DETECTX | Cloud Security Expert, 20181018_ExecutionIndicators_v01.xlsx. [Online]. Verfügbar unter: https://1234n6-my.sharepoint.com/:x:/p/adam/EU3Fk3ec6NdPsSQx1eA1sfwB_R_fRa4tJ4c1FR6WJlWIEA?rtime=pDn9Qzt92kg (Zugriff am: 13. August 2022).

[34] Microsoft Corporation, [MS-CDP]: Connected Devices Platform Protocol Version 3. [Online]. Verfügbar unter: https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-CDP/%5bMS-CDP%5d.pdf (Zugriff am: 10. August 2022).

[35] C. Katsavounidis, „Windows 10 ActivitiesCache.db examination“. [Online]. Verfügbar unter: https://kacos2000.github.io/WindowsTimeline/WindowsTimeline.pdf (Zugriff am: 03. August 2022).

[36] Joachim Metz, libscca/Windows Prefetch File (PF). [Online]. Verfügbar unter: https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20(PF)%20format.asciidoc (Zugriff am: 13. August 2022).

[37] Prefetch - Forensics Wiki. [Online]. Verfügbar unter: https://forensicswiki.xyz/wiki/index.php?title=Prefetch (Zugriff am: 10. August 2022).

[38] DETECTX | Cloud Security Expert, Available Artifacts - Evidence of Execution - DETECTX | Cloud Security Expert. [Online]. Verfügbar unter: https://www.detectx.com.au/available-artifacts-evidence-of-execution/ (Zugriff am: 19. Juli 2022).

[39] Anonym, Windows Prefetch File Format - Forensics Wiki. [Online]. Verfügbar unter: https://forensicswiki.xyz/wiki/index.php?title=Windows_Prefetch_File_Format (Zugriff am: 10. August 2022).

[40] A. Duranec, D. Topolcic, K. Hausknecht und D. Delija, „Investigating file use and knowledge with Windows 10 artifacts“ in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, S. 1213–1218, doi: 10.23919/MIPRO.2019.8756877 (Zugriff am: 13. August 2022).

[41] B. Singh und U. Singh, „Program execution analysis in Windows: A study of data sources, their format and comparison of forensic capability“, Computers & Security, Jg. 74, S. 94–114, 2018, doi: 10.1016/j.cose.2018.01.006.

[42] W. E. Blog, Windows 8 Task Manager In-Depth. [Online]. Verfügbar unter: https://blogs.windows.com/windowsexperience/2013/06/06/windows-8-task-manager-in-depth/ (Zugriff am: 10. Juli 2022).

[43] LIFARS, „SRUM Another Windows Time Machine“.

[44] Simon Key, SRUM Database Parser. [Online]. Verfügbar unter: https://security.opentext.com/appDetails/SRUM-Database-Parser (Zugriff am: 13. Juli 2022).

[45] David Cowen, Daily Blog #595: Solution Saturday 1/12/19 - Hacking Exposed Computer Forensics Blog. [Online]. Verfügbar unter: https://www.hecfblog.com/2019/01/daily-blog-595-solution-saturday-11219.html (Zugriff am: 13. Juli 2022).

[46] Joachim Metz, esedb-kb/System Resource Usage Monitor (SRUM).asciidoc at main · libyal/esedb-kb. [Online]. Verfügbar unter: https://github.com/libyal/esedb-kb/blob/main/documentation/System%20Resource%20Usage%20Monitor%20(SRUM).asciidoc (Zugriff am: 13. Juli 2022).

[47] T. Senjyu, P. N. Mahalle, T. Perumal und A. Joshi, Hg., Information and Communication Technology for Intelligent Systems. Singapore: Springer

Singapore, 2021 (Zugriff am: 08. August 2022)..

[48] Microsoft Corporation, NET_LUID-Wert - Windows drivers. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows-hardware/drivers/network/net-luid-value (Zugriff am: 26. Juni 2022).

[49] C. Doemel, „App Timeline Provider – SRUM Database“, AboutDFIR, 23. Mai 2022, 2022. [Online]. Verfügbar unter: https://aboutdfir.com/app-timeline-provider-srum-database/. Zugriff am: 13. Juli 2022.

[50] ArtiFast BLog, Investigating Windows System Resource Usage Monitor (SRUM). [Online]. Verfügbar unter: https://forensafe.com/blogs/srudb.html (Zugriff am: 13. Juli 2022).

[51] Brendan Mc Creesh, SRUM – Digital Forensics. [Online]. Verfügbar unter: https://digitalforensicsdotblog.wordpress.com/tag/srum/ (Zugriff am: 13. Juli 2022).

[52] Microsoft Corporation, NET_LUID_LH (ifdef.h) - Win32 apps. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows/win32/api/ifdef/ns-ifdef-net_luid_lh (Zugriff am: 26. Juni 2022).

[53] IANA, https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib. [Online]. Verfügbar unter: https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib (Zugriff am: 6. August 2022).

[54] Microsoft Corporation, Übersicht über Windows-Pushbenachrichtigungsdienste (Windows Push Notification Services, WNS) - Windows apps. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows/apps/design/shell/tiles-and-notifications/windows-push-notification-services--wns--overview (Zugriff am: 23. Juli 2022).

[55] M. Cohen, „Digging into the System Resource Usage Monitor (SRUM)“, Velociraptor IR, 30. Dez. 2019, 2019. [Online]. Verfügbar unter: https://velociraptor.velocidex.com/digging-into-the-system-resource-usage-monitor-srum-afbadb1a375. Zugriff am: 13. Juli 2022.

[56] Adam Harrison, 1234n6: Testing of SRUM on Windows Server 2019. [Online]. Verfügbar unter: https://blog.1234n6.com/2019/01/testing-of-

srum-on-windows-server-2019.html (Zugriff am: 6. August 2022).

[57] DEVGC, devgc/SrumMonkey: Tool to parse SRU database. [Online]. Verfügbar unter: https://github.com/devgc/SrumMonkey (Zugriff am: 13. Juli 2022).

[58] Eric Zimmerman, SrumECmd. [Online]. Verfügbar unter: https://github.com/EricZimmerman/Srum#repairing-the-srudbdat (Zugriff am: 13. Juli 2022).

[59] Mark Baggett, MarkBaggett/srum-dump: A forensics tool to convert the data in the Windows srum (System Resource Usage Monitor) database to an xlsx spreadsheet. [Online]. Verfügbar unter: https://github.com/MarkBaggett/srum-dump (Zugriff am: 13. Juli 2022).

[60] Microsoft Corporation, Sysmon - Windows Sysinternals. [Online]. Verfügbar unter: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon (Zugriff am: 26. Juli 2022).

[61] F. Nilsson, SysMon – A framework for monitoring and measuring real-time properties [Online]. Verfügbar unter: https://www.diva-portal.org/smash/get/diva2:535850/FULLTEXT01.pdf (Zugriff am: 26. Juli 2022).

[62] Mark Russionovich, „Tracking Hackers on Your Network with Sysinternals Sysmon: RSA Conference 2016“, Jg. 2016.

[63] NirSoft, Monitor network usage / bandwidth of every application on Windows. [Online]. Verfügbar unter: https://www.nirsoft.net/utils/app_network_counter.html (Zugriff am: 26. Juli 2022).

[64] Wireshark · Go Deep. [Online]. Verfügbar unter: https://www.wireshark.org/#download (Zugriff am: 26. Juli 2022).

[65] Kroll, Kroll Artifact Parser And Extractor (KAPE) | Cyber Risk Services. [Online]. Verfügbar unter: https://www.kroll.com/en/services/cyber-risk/incident-response-litigation-support/kroll-artifact-parser-extractor-kape (Zugriff am: 26. Juli 2022).

[66] Microsoft Corporation, Download Windows 11. [Online]. Verfügbar unter: https://www.microsoft.com/de-de/software-download/windows11 (Zugriff am: 27. Juli 2022).

[67] Microsoft Corporation, Windows 11 Specs and System Requirements. [Online]. Verfügbar unter: https://www.microsoft.com/en-us/windows/windows-11-specifications?r=1 (Zugriff am: 27. Juli 2022).

[68] VMware, „Installing Windows 11 as a guest OS on VMware Workstation Pro/Player and Fusion (86207)“, VMware, 29. Okt. 2021, 2021. [Online]. Verfügbar unter: https://kb.vmware.com/s/article/86207. Zugriff am: 25. Juli 2022.

[69] Microsoft Corporation, Windows 11 release information. [Online]. Verfügbar unter: https://docs.microsoft.com/de-de/windows/release-health/windows11-release-information (Zugriff am: 8. Juli 2022).

[70] Exterro, FTK Imager - Exterro. [Online]. Verfügbar unter: https://www.exterro.com/ftk-imager (Zugriff am: 27. Juli 2022).

[71] Exterro AccessData, FTK Imager 4.7. [Online]. Verfügbar unter: https://go.exterro.com/l/43312/2022-01-21/f6h1s3 (Zugriff am: 27. Juli 2022).

[72] Arsenal Recon, Arsenal Image Mounter. [Online]. Verfügbar unter: https://arsenalrecon.com/products/arsenal-image-mounter (Zugriff am: 27. Juli 2022).

[73] Eric Zimmerman, EvtxECmd. [Online]. Verfügbar unter: https://github.com/EricZimmerman/evtx (Zugriff am: 7. August 2022).

[74] Eric Zimmerman, PECmd: Prefetch Explorer Command Line. [Online]. Verfügbar unter: https://github.com/EricZimmerman/PECmd (Zugriff am: 13. August 2022).

[75] Eric Zimmerman, Full featured, offline Registry parser in C#. [Online]. Verfügbar unter: https://github.com/EricZimmerman/Registry (Zugriff am: 7. August 2022).

[76] Eric Zimmerman, TimeApp: Simple time and public IP app, useful for

recording the screen while interacting with a computer for later corroboration of artifacts against time. [Online]. Verfügbar unter: https://github.com/EricZimmerman/timeapp (Zugriff am: 27. Juli 2022).

[77] Eric Zimmerman, WxTCmd. [Online]. Verfügbar unter: https://github.com/EricZimmerman/WxTCmd (Zugriff am: 13. August 2022).

[78] HHD Software, Hex Editor Neo: Download Fastest Hexadecimal Viewer/Editor. [Online]. Verfügbar unter: https://www.hhdsoftware.com/hex-editor (Zugriff am: 27. Juli 2022).

[79] Mark Baggett, srum-dump/SRUM_TEMPLATE2_ORIG.xlsx at master · MarkBaggett/srum-dump. [Online]. Verfügbar unter: https://github.com/MarkBaggett/srum-dump/blob/master/SRUM_TEMPLATE2_ORIG.xlsx (Zugriff am: 13. Juli 2022).

[80] NirSoft, ESEDatabaseView v1.70: View / Open ESE Database Files (Jet Blue / .edb files). [Online]. Verfügbar unter: https://www.nirsoft.net/utils/ese_database_view.html (Zugriff am: 27. Juli 2022).

[81] Anonymous, Forensics Quickie: Mounting Split .vmdk. [Online]. Verfügbar unter: https://www.4n6k.com/2011/09/forensics-quickie-mounting-split-vmdk.html (Zugriff am: 25. Juli 2022).

[82] Solved: How to compact encrypted .vmdk Workstation 11 - VMware Technology Network VMTN. [Online]. Verfügbar unter: https://communities.vmware.com/t5/VMware-Workstation-Pro/How-to-compact-encrypted-vmdk-Workstation-11/td-p/1814650 (Zugriff am: 28. Juli 2022).

[83] GitHub, AndrewRathbun / DFIRArtifactMuseum. [Online]. Verfügbar unter: https://github.com/AndrewRathbun/DFIRArtifactMuseum (Zugriff am: 25. Juli 2022).

[84] Anonym, BLOB – IT-Forensik Wiki. [Online]. Verfügbar unter: https://it-forensik.fiw.hs-wismar.de/index.php/BLOB (Zugriff am: 28. Juli 2022).

[85] Thomas V Fischer, „Beyond Windows Forensics with Built-in Microsoft

Tooling“, 2019. [Online]. Verfügbar unter: https://deepsec.net/docs/Slides/2019/Beyond_Windows_Forensics_with_Built-in_Microsoft_Tooling_Thomas_Fischer.pdf

10 Bilderverzeichnis

11 Tabellenverzeichnis

12 Anlagen

12.1 Tabelle 13: Technische Daten des Testsystems

Tabelle 13: Technische Daten des Testsystems

Beschreibung	Daten
Arbeitsspeicher	6 GB
Festplattengröße	54 GB
CPU	Intel Core i5-6200U CPU 2,30 GHz
Netzwerkadaptertyp	NAT
Betriebssystem	Microsoft Windows 11 Pro v. 21H2 22000.778
Updatestand	KB5014668
Microsoft-Konto	srum@outlook.de

12.2 Tabelle 14: SRUM-Tabellenbezeichnungen und -Providernamen

Tabelle 14: SRUM-Tabellenbezeichnungen und -Providernamen

Tabellenbezeichnung/GUID	EWF-Providername
MSysLocales	-
MSysObjects	-
MSysObjectsShadow	-
MSysObjids	-
SruDbCheckpointTable	-
SruDbIdMapTable	-
{5C8CF1C7-7257-4F13-B223-970EF5939312}	App Timeline Provider
{7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F}	VFUProv
{973F5D5C-1D90-4944-BE8E-24B94231A174}	Windows Network Data Usage Monitor
{D10CA2FE-6FCF-4F6D-848E-B2E99266FA86}	WPN SRUM Provider

{D10CA2FE-6FCF-4F6D-848E-B2E99266FA89}	Application Resource Usage Provider
{DA73FB89-2BEA-4DDC-86B8-6E048C6DA477}	Energy Estimation Provider
{DD6636C4-8929-4683-974E-22C046A43763}	Windows Network Connectivity Usage Monitor
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}	Energy Usage Provider
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT	Energy Usage Provider Long-Term
{B6D82AF1-F780-4E17-8077-6CB9AD8A6FC4}	Tagged Energy Provider
{17F4D97B-F26A-5E79-3A82-90040A47D13D}	SDP Volume Provider
{841A7317-3805-518B-C2EA-AD224CB4AF84}	SDP Physical Disk Provider
{DC3D3B50-BB90-5066-FA4E-A5F90DD8B677}	SDP Cpu Provider
{EEE2F477-0659-5C47-EF03-6D6BEFD441B3}	SDP Network Provider

12.3 Tabelle 15: Technische Daten der Workstation

Tabelle 15: Technische Daten der Workstation

Beschreibung	Daten
Hersteller	Lenovo
Bezeichnung	ThinkPad T470/25
Betriebssystem	Microsoft Windows 10 Pro v. 21H2 19044.1645
Festplattengröße	Samsung SSD 860 QVO 1000 GB; NVM TS512GMTS30S 512 GB
CPU	Intel Core i5-6200U CPU 2.30 GHz
Arbeitsspeicher	24 GB

12.4 Tabelle 16: Windows-11-Installationsparameter

Tabelle 16: Windows-11-Installationsparameter

Settings	Wert
Language to install	English (United Kingdom)
Time and currency format	German (Germany)
Keyboard or input method	German
Version	Windows 11 Pro x64
Region	Germany
Devicename	thesis_srum
Set up	Personal Use
Account	srum@outlook.de
Let Microsoft and apps use your location	No
Find my device	No
Send diagnostic data to Microsoft	Send diagnostic data to Microsoft
Improve inking & typing	No
Get tailored experiences with diagnostic data	No
Let apps use advertising ID	No
Back up files with OneDrive	Local

12.5 Tabelle 17: Vollständiger Testplan

Tabelle 17: Vollständiger Testplan

Schritt	Handlung
1	Durchführung von Nutzerinteraktionen/Duplizieren des Ordners auf der Workstation/Duplikat als Ausgangspunkt (Zustand 1)
2	Duplizieren von Zustand 1/Sauberes Herunterfahren des Systems via GUI
3, 6, 9, 12, 15, 18	Erstellung des EWF-Datenträgerimage
4, 7, 10, 13, 16	Duplizieren von Zustand 1
5	Herunterfahren des Systems via VMware (unsauberer Shutdown)
8	Löschen der SRUM-Historie im Taskmanager/Sauberes Herunterfahren des Systems via GUI
11	Löschen der SRUM-Historie im Taskmanager löschen/Herunterfahren des Systems via VMware (unsauberer Shutdown)
14	Ausführen von CCleaner/Sauberes Herunterfahren des Systems via GUI
17	Ausführen von CCleaner/Herunterfahren des Systems via VMware (unsauberer Shutdown)
19	Ende

12.6 Bild 22: Tabellenabgleich SRUDB.dat Windows 11

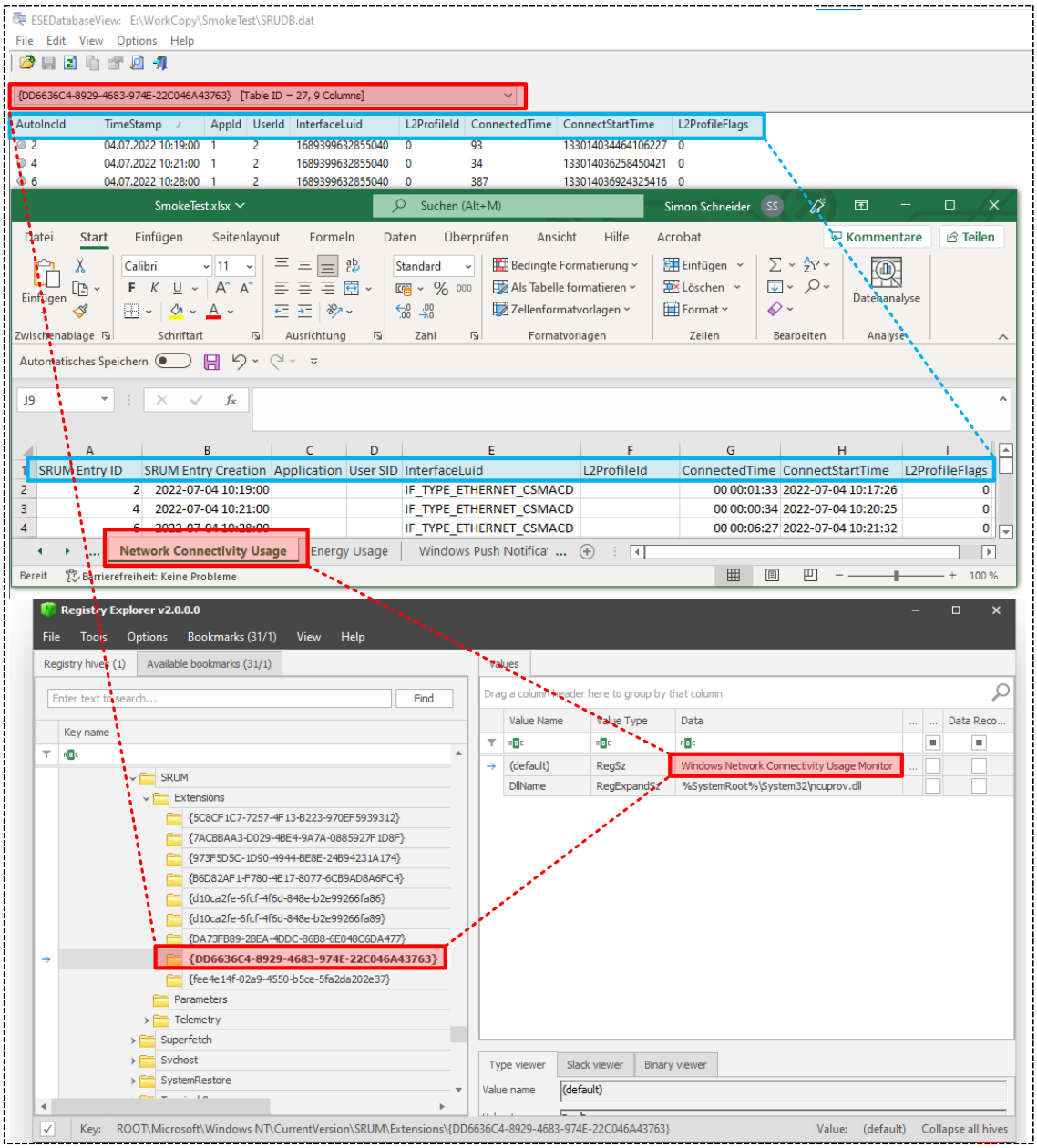

Bild 22: Tabellenabgleich SRUDB.dat Windows 11

12.7 Tabelle 18: SRUM-Tabellennamen unter Windows 10/11

Tabelle 18: SRUM-Tabellennamen unter Windows 10/11

GUID unter Windows 10	EWF Provider / Name	GUID unter Windows 11	Tabellenbezeichnung srum_dump2
{5C8CF1C7-7257-4F13-B223-970EF5939312}	App Timeline Provider	{5C8CF1C7-7257-4F13-B223-970EF5939312}	App Timeline Provider
{7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F}	vfuprov	{7ACBBAA3-D029-4BE4-9A7A-0885927F1D8F}	vfuprov
{973F5D5C-1D90-4944-BE8E-24B94231A174}	Windows Network Data Usage Monitor	{973F5D5C-1D90-4944-BE8E-24B94231A174}	Network Data Usage
{D10CA2FE-6FCF-4F6D-848E-B2E99266FA86}	WPN SRUM Provider	{D10CA2FE-6FCF-4F6D-848E-B2E99266FA86}	Windows Push Notifications
{D10CA2FE-6FCF-4F6D-848E-B2E99266FA89}	Application Resource Usage Provider	{D10CA2FE-6FCF-4F6D-848E-B2E99266FA89}	Application Resource Usage
{DA73FB89-2BEA-4DDC-86B8-6E048C6DA477}	Energy Estimation Provider	{DA73FB89-2BEA-4DDC-86B8-6E048C6DA477}	Energy Estimation Provider
{DD6636C4-8929-4683-974E-22C046A43763}	Windows Network Connectivity Usage Monitor	{DD6636C4-8929-4683-974E-22C046A43763}	Network Connectivity Usage
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}	Energy Usage Provider	{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}	Energy Usage
{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT	Energy Usage Provider Long-Term	{FEE4E14F-02A9-4550-B5CE-5FA2DA202E37}LT	Energy Usage LT
{B6D82AF1-F780-4E17-8077-6CB9AD8A6FC4}	Tagged Energy Provider	im Test nicht vorhanden	im Test nicht vorhanden
Unbekannt	Unbekannt	-	ruDbCheckpoint

12.8 Tabelle 19: Ablauf des Testfalls

Tabelle 19: Ablauf des Testfalls

händische Zeit (MESZ)	SysMon (UTC)	Bemerkung
	12:09:51	Einschalten
14:11	12:12	OpenOffice-Installation
	12:12	OpenOffice-Installation User SRUM
14:13	12:13	OpenOffice-Installation User Administrator
14:14	12:13	OpenOffice-Installation fertig
14:16	12:16	Start von OpenOffice
14:17	12:17	OpenOffice Writer im Vordergrund
14:23	12:23	Schließen von OpenOffice
14:27	12:27	Installation von Notepad++
14:28	-	Notepad++-Installfiles gelöscht
14:29	12:29	VLC-Media-Player-Installation
14:30	12:30	VLC-Start mit Musik im Vordergrund
14:41	12:41	Notepad++ zweimal gleichzeitig geöffnet
14:44	12:44	Schließen von beiden Notepad++-Anwendungen
16:00	14:00	VLC-Player geschlossen
16:02	14:02:51	Ausschalten
16:55	14:55:48	Einschalten
17:02	15:02	CMD mit Befehl ‚ipconfig‘
17:07	-	Verbindung mit WLAN-Handy-Hotspot
~ 17:09	15:07 u. 15:13	Öffnen von Outlook und Senden zweier Mails
17:11	15:11	Ausführung der CMD
17:17	15:18	Installation von Adobe Acrobat
17:17	15:17	Notepad++ geöffnet
17:20	15:20	Installation von Adobe Acrobat DC (64-bit)
17:21	15:21	Adobe-Installation fertig

17:22	15:22	Adobe öffnet PDF
17:23	15:23:23	Ausschalten
17:23	15:23:35	Einschalten
17:24	15:24	Notepad++ geöffnet
17:24	15:24:32	Ausschalten
17:24	15:24:43	Einschalten
17:25	15:26	Notepad++ geöffnet
17:26	15:26:24	Ausschalten
17:26	15:26:36	Einschalten
17:28	15:28	Notepad++ zweimal gestartet
17:28	15:28:39	Ausschalten
17:31	15:31:29	Einschalten
17:32	-	USB-Stick angeschlossen
17:33	15:33	Starten von Wireshark (Portable) vom USB-Stick
17:34	15:34	Installation von Wireshark (Portable), Extraktion nach C\..\Downloads
17:35	15:35	Start von Wireshark (Portable) auf C:\
17:37	15:37	Wireshark von Downloads nach Desktop kopieren und starten
17:39	15:39:30	Ausschalten
17:40	15:40:10	Einschalten
17:41	15:41 bis 15:42	Installation von KeePass
17:42	15:42	Starten von KeePass
17:43	15:43	Installation von MEGAsync (Mega.io)
17:44	15:44	Start/Nutzung von Mega.io
17:48	-	Start Datei-Download durch Mega.io
17:49	-	Ende Datei-Download
17:53	15:53	Ausführung der CMD, starten von Calc.exe, sofortiges schließen von Calc.exe
17:54	15:54	Ausführung der CMD, starten von Calc.exe

17:55	15:55	Öffnen des Windows-App-Store
17:55 bis 17:57	-	Download 16,4 MB Windows-App-Store (Amazon-Prime-App)
17:57	15:56	Start von Amazon-Prime-App
19:59	17:59	Ende Amazon-Prime-App
19:59	17:59	Calc.exe und CMD geschlossen
20:00	17:59	Start Datei-Download durch Mega.io
20:02	-	Ende Mega-Download
20:05	18:05	PDFs auf dem USB öffnen
20:06	18:06	Schließen von KeePass
20:07	20:07:11	Ausschalten

12.9 Tabelle 20: VM-Start- und -Herunterfahrzeiten

Tabelle 20: VM-Start- und -Herunterfahrzeiten

Uhrzeit (MESZ)	Uhrzeit (UTC)	Aktion
14:09:46	12:09:46	Einschalten
16:02:49	14:02:49	Ausschalten
16:55:43	14:55:43	Einschalten
17:23:20	15:23:20	Ausschalten
17:23:45	15:23:45	Einschalten
17:24:29	15:24:29	Ausschalten
17:24:54	15:24:54	Einschalten
17:26:21	15:26:21	Ausschalten
17:26:48	15:26:48	Einschalten
17:28:37	15:28:37	Ausschalten
17:31:19	15:31:19	Einschalten
17:39:28	15:39:28	Ausschalten
17:40:24	15:40:24	Einschalten
20:06:39	18:06:39	Ausschalten