



INSTITUTO POLITÉCNICO DE BEJA

Escola Superior de Tecnologia e Gestão

Mestrado em Engenharia de Segurança Informática

**FORENSIC ACQUISITION OF FILE SYSTEMS WITH PARALLEL
PROCESSING OF DIGITAL ARTIFACTS TO GENERATE AN EARLY
CASE ASSESSMENT REPORT**

André Hakime Dutra

Beja, Portugal

2021

INSTITUTO POLITÉCNICO DE BEJA
Escola Superior de Tecnologia e Gestão
Mestrado em Engenharia de Segurança Informática

**FORENSIC ACQUISITION OF FILE SYSTEMS WITH PARALLEL
PROCESSING OF DIGITAL ARTIFACTS TO GENERATE AN EARLY
CASE ASSESSMENT REPORT**

André Hakime Dutra

Orientado por:

Professor Armando de Jesus Ventura, IPBeja

Professor Mario Jorge Costa Candeias, IPBeja

Dissertação apresentada na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja como requisito parcial para obtenção do grau de Mestre em Engenharia de Segurança Informática

Resumo

Aquisição forense de sistemas de arquivos com processamento paralelo de artefactos digitais para gerar um relatório de análise preliminar

A evolução da maneira como os seres humanos interagem e realizam tarefas rotineiras mudou nas últimas décadas e uma longa lista de atividades agora somente são possíveis com o uso de tecnologias da informação – entre essas pode-se destacar a aquisição de bens e serviços, gestão e operações de negócios e comunicações. Essas transformações são visíveis também em outras atividades menos legítimas, permitindo que crimes sejam cometidos através de meios digitais.

Em linhas gerais, investigadores forenses trabalham buscando por indícios de ações criminais realizadas por meio de dispositivos digitais para finalmente, tentar identificar os autores, o nível do dano causado e a história atrás que possibilitou o crime. Na sua essência, essa atividade deve seguir normas estritas para garantir que as provas sejam admitidas em tribunal, mas quanto maior o número de novos artefatos e maior o volume de dispositivos de armazenamento disponíveis, maior o tempo necessário entre a identificação de um dispositivo de um suspeito e o momento em que o investigador começa a navegar no mar de informações alojadas no dispositivo.

Esta pesquisa, tem como objetivo antecipar algumas etapas do EDRM através do uso do processamento em paralelo adjacente nas unidades de processamento (CPU) atuais para para traduzir múltiplos artefactos forenses do sistema operativo Windows 10 e gerar um relatório com as informações mais cruciais sobre o dispositivo adquirido. Permitindo uma análise antecipada do caso (ECA) ao mesmo tempo em que uma aquisição completa do disco está em curso, desse modo causando um impacto mínimo no tempo geral de aquisição.

Palavras-chave: forense digital, ediscovery, edrm, triagem forense, forense em sistemas Windows, processamento paralelo, aquisição forense, artefactos forenses, cibercrime.

Abstract

Forensic acquisition of file systems with parallel processing of digital artifacts to generate an early case assessment report

The evolution of the way humans interact and perform routine activities have changed on the past decades, and a long list of tasks are now only possible with information technology, among those can be highlighted the shopping of goods and services, business operations and communication. These transformations are also reflected in other less legitimate activities, allowing crimes to be performed through digital mediums.

In brief, digital forensics investigators work searching for footprints of criminal actions performed on digital devices to, ultimately, try to identify the authors, the level of damage and the history behind that enabled the activity to be performed. In essence, this activity must follow strict rules to ensure the court admissibility. The increasing volume of newer artifacts and larger available storage drives, the longer time needed between the identification and preservation of the potential devices and the moment that the investigator starts navigating through the sea of information stored.

This research has the objective to anticipate some steps of the EDRM workflow, by utilizing parallel processing of the current CPUs to parse multiple Windows 10 forensic artifacts and generating a report with the most crucial information about the preserved device to enable an early case assessment at the same time that a full disk preservation is in course, hence with a minimal impact on the overall preservation time.

Keywords: digital forensics, ediscovery, edrm, forensic triage, windows forensics, parallel processing, forensic preservation, forensic artifacts, cybercrime.

Acknowledgements

First of all, I would like to thank my father Noilton Hakime Dutra, who started to show me the beauty that exists in every time you learn a new thing, I wish he could be present to read this work.

I would also like to give a special thank my mother Martha Borin, the strong woman that worked hard every day of her life so I could have access to the university to achieve all my dreams.

To my partner, Deise Costa, who was always on my side every time I lost track of my tasks and was present to support, incentive and not let me give up.

When I started the master's degree in Engineering in Information Security, I didn't expect this experience to be so intense and to bring this incredible knowledge, I would like to thank from the deep of my hearth to my friend and teachers Mario Candeias, Armando Ventura and Rui Silva, and all the other teachers that made this course possible.

Finally, I would like to thank all my friends and colleagues that supported me along the way by lending equipment, time, and acknowledged every celebration that I missed while I was busy with my research.

Agradecimentos

Primeiramente, gostaria de agradecer ao meu pai Noilton Hakime Dutra, que me mostrou desde pequeno a beleza que existe em cada vez que se aprende uma nova coisa. Eu gostaria muito que ele pudesse estar presente para ler esse trabalho.

Eu também gostaria de deixar um agradecimento especial à minha mãe, Martha Borin, a mulher forte que trabalhou muito cada dia de sua vida para que eu tivesse acesso à universidade e pudesse alcançar todos os meus sonhos.

Sou grato à minha companheira, Deise Costa, quem estava sempre ao meu lado me dando suporte quando eu me perdia durante a pesquisa, me incentivando e não deixando eu desistir.

Quando comecei o mestrado em Engenharia em Segurança da Informação, eu não esperava que essa experiência fosse tão intensa e trouxesse todo esse incrível conhecimento. Eu gostaria de agradecer do fundo do meu coração aos meus amigos e professores Mario Candeias, Armando Ventura e Rui Silva, assim como a todo o corpo docente que tornou esse curso possível.

Finalmente, eu gostaria de agradecer a todos os meus amigos e colegas que me auxiliaram ao longo do caminho, seja emprestando equipamentos, tempo ou compreendendo minha ausência em cada celebração que perdi por estar ocupado com minha pesquisa.

Index

Resumo	i
Abstract.....	iii
Acknowledgements	v
Agradecimentos	vii
Index.....	ix
Index of Figures.....	xiii
Index of Tables	xv
List of Abbreviations	xvii
Introduction	1
Research delimitation	3
I. Objectives	3
II. Justification	4
III. Hypothesis and previous research	5
Methodological Procedure	7
1. Introduction to Digital Forensics	9
1.1. Forensic acquisition and preservation types and procedures	11
1.2. The fraud triangle.....	14
1.3. Basic concepts	16
1.3.1. Forensic Artifact.....	16
1.3.2. Metadata	16
1.3.3. Algorithmic Hash.....	18
1.3.4. Chain of Custody	19
1.3.5. Windows Registry	20
1.3.6. File Signature	21
1.4. Digital Forensics Investigation Frameworks	21
1.4.1. Identification.....	23
1.4.2. Preservation.....	23
1.4.3. Collection	25
1.4.4. Examination	28

Index

1.4.5.	Analysis	31
1.4.6.	Presentation.....	33
1.5.	Early Case Assessment	34
1.5.1.	Definition	35
1.5.2.	ECA and Digital Forensics: Advantages, Challenges and implementation	36
1.5.3.	The main questions that can be answered with ECA	40
2.	Windows Digital Forensics Artifacts with low processing requirements	43
2.1.	Operating System.....	43
2.1.1.	Users (Dates and details)	43
2.1.2.	Operating System Details (Last shutdown, version, updates).....	43
2.1.3.	Hardware Details	44
2.1.4.	Time zone.....	44
2.2.	Applications.....	44
2.2.1.	Installed Applications.....	45
2.2.2.	Cloud Drives	45
2.2.2.1.	Microsoft One Drive (former SkyDrive)	45
2.2.2.2.	Dropbox.....	46
2.2.2.3.	Google Drive.....	46
2.2.3.	Skype logs.....	46
2.2.4.	iTunes Backups.....	47
2.3.	Anti-Forensic Indicators	47
2.3.1.	Recycle Bin	47
2.3.2.	File System listing and Deleted Files from the Main File Table (MFT)	48
2.3.3.	Virtual Machines	49
2.3.4.	Remote Access Tools	49
2.3.5.	Wiping Tools	49
2.3.6.	Encryption Tools	50
2.3.7.	Others: Steganography, TOR/VPN, metadata manipulation, bootable ISO	51
2.4.	Recent Open Files and Applications.....	52
2.4.1.	Recent Files	52
2.4.2.	Jump Files.....	52

2.4.3.	Link Files.....	52
2.4.4.	Recent Office Documents	53
2.4.5.	Notepad++ Sessions.....	53
2.4.6.	Downloads	54
2.4.7.	Print Spooler	54
2.4.8.	Prefetch.....	54
2.5.	Internet	55
2.5.1.	Bookmarks	55
2.5.2.	Recent URLs and Searches.....	56
2.6.	Mailboxes	56
2.7.	USB Devices List	57
2.8.	Memory.....	57
2.8.1.	Pagination	58
2.8.2.	Hibernation	59
2.8.3.	Swapfile.....	59
2.8.4.	Memory dumps.....	60
3.	Implementation of tool to perform ECA during forensic collection	61
3.1.	Definition of metrics and technologies.....	61
3.2.	When to collect the ECA data	64
3.3.	Which artifacts to collect and process.....	65
3.4.	Development.....	66
3.5.	Clearing previous execution.....	68
3.6.	Write Protection and Automount.....	68
3.7.	Start Early Case Analysis (ECA) Preservation	70
3.8.	Automatic extraction of user generated files	71
3.9.	Extraction of Master File Table (MFT).....	75
3.10.	Forensic logical copy of identified files	76
3.11.	Start Full Physical Preservation (if selected)	77
3.12.	Registry Reader.....	78
3.13.	Mailbox (PST) Parser.....	79
3.14.	Recent Files Parser	79
3.15.	Prefetch Parser	80

Index

3.16.	Google Chrome Parser.....	80
3.17.	Mozilla Firefox Parser	81
3.18.	Recycle Bin Parser.....	81
3.19.	USB Devices Parser	82
3.20.	Indexing of small items.....	82
3.21.	Report generation	82
3.22.	Finalization.....	83
3.23.	Image Viewer and Database Analysis.....	85
3.23.1.	Open the Hakime Forensics Viewer.....	85
3.23.2.	Mount the Forensic Image File	86
3.23.2.1.	Mount with the Hakime Forensics Image Viewer.....	86
3.23.2.2.	Mount with 3rd party tools.....	88
3.23.3.	Investigate the index database.....	88
3.23.3.1.	Navigate the index database with the Hakime Forensics Image Viewer 88	
3.23.3.2.	Open the index database with 3rd party tools	90
4.	Analysis of results	91
	Future challenges.....	96
	Conclusion.....	100
	Bibliographic References	102
	Appendix I – Forensic artifact locations.....	110
	Appendix II – Tabulation of execution times	128
	Appendix III – Description of Hardware utilized on tests.....	130

Index of Figures

Figure 1 Fraud Triangle	15
Figure 2 EDRM Framework (EDRM.net, 2020).....	22
Figure 3 DFRWS Framework (DFRWS.org, 2001)	22
Figure 4 - Data Culling Diagram.....	39
Figure 5 - Collection Workflow.....	63
Figure 6 - Hakime Forensics - Main interface – Confirm Mount Drives.....	67
Figure 7 - Hakime Forensics - Collection Details (ECA Modules and Case Details).....	68
Figure 8 - Hakime Forensics - Mount options and preservation methods.	69
Figure 9 - Hakime Forensics - Execution Log screen.	71
Figure 10 - Hakime Forensics - Advanced Options (Extension list, OS Artifacts, Registry Reader Artifacts and Maximum file size).....	72
Figure 11 - Hakime Forensics - Execution log screen with full disk image.....	78
Figure 12 - Excel Report (example)	83
Figure 13 - Summary Log example.....	84
Figure 14 - Hakime Forensics Image Viewer - Startup Screen	86
Figure 15 - Hakime Forensics Image Viewer - Mount Image File.....	87
Figure 16 - Hakime Forensics Image Viewer - Image File Mounted.....	87
Figure 17 - Hakime Forensics Image Viewer - Mount database file	89
Figure 18 - Hakime Forensics Image Viewer - Table List.....	89
Figure 19 - Hakime Forensics Image Viewer - Simple Query Results (example).....	90
Figure 20 - Time per Scenario (seconds)	93
Figure 21 - Laptop Asus model details	130
Figure 22 - SSD Source model details.....	130
Figure 23 - SSD Destination model details.....	131
Figure 24- HDD Destination model details.....	131
Figure 25 - Tableau T35u Bridge	132
Figure 26 - JMicron SATA to USB3.0 enclosure.....	132

Index of Tables

Table 1 - OS Artifacts JSON.....	73
Table 2 - RegReader JSON example	74
Table 3 - Summary of Scenarios Benchmark.....	94
Table 4 - Forensic Artifacts - User Profiles	110
Table 5 - Forensic Artifacts - Operating System Details.....	111
Table 6 - Forensic Artifacts - Hardware Details.....	112
Table 7 - Forensic Artifacts - Timezone.....	112
Table 8- Forensic Artifacts - Installed Applications.....	113
Table 9- Forensic Artifacts - Cloud Applications	114
Table 10- Forensic Artifacts - Skype.....	114
Table 11 - Forensic Artifacts - iTunes Backups.....	115
Table 12 - Forensic Artifacts - Recycle Bin	115
Table 13 - Forensic Artifacts - Main File Table (MFT).....	115
Table 14 - Forensic Artifacts - Virtual Machines	116
Table 15 - Forensic Artifacts - Remote Access Tools.....	117
Table 16 - Forensic Artifacts - Wiping Tools.....	117
Table 17 - Forensic Artifacts - Encryption Tools.....	118
Table 18 - Forensic Artifacts - Other Anti-forensic Indicators (Steganography, TOR, VPN, Metadata Changing, Bootable ISO and system configurations)	119
Table 19 - Forensic Artifacts - Recently Used Files	120
Table 20 - Forensic Artifacts - Jump Lists	120
Table 21 - Forensic Artifacts - Link Files (LNK)	120
Table 22 - Forensic Artifacts - Recent Office Documents	121
Table 23 - Forensic Artifacts - Notepad++ Sessions	121
Table 24- Forensic Artifacts - Downloads	121
Table 25 - Forensic Artifacts - Print Spooler.....	122
Table 26 - Forensic Artifacts - Prefetch.....	122
Table 27 - Forensic Artifacts - IE/Edge Bookmarks	123
Table 28- Forensic Artifacts - IE/Edge Recent URLs and Searches.....	123
Table 29 - Forensic Artifacts – Mailboxes	124
Table 30 - Forensic Artifacts - USB Devices List	125
Table 31 - Forensic Artifacts - Memory Dumps	126
Table 32 - Execution Times	128

List of Abbreviations

AFF4 – Advanced Forensic Format 4

BEC – Belkasoft Evidence Center

CD – Compact Disk

CFTT – Computer Forensic Tool Testing

CPU – Central Processing Unit

DFRWS – Digital Forensic Research Workshop

DVD – Digital Versatile Disc

ECA – Early Case Assessment

eDiscovery – Electronic Discovery

EDRM – Electronic Discovery Reference Model

ESI – Electronic Stored Information

EWf – Expert Witness Forensic

exFAT – Extensible File Allocation Table

FTK – AccessData Forensic Toolkit

GUI – Graphical User Interface

HDD – Hard Disk Drive

I/O – Input/Output

ISO/IEC – International Organization for Standardization

MD5 – MD5 Message-Digest Algorithm

MFT – Main File Table

MRU – Most Recently Utilized

MS – Microsoft Software

NIST – National Institute of Standards and Technology

NSRL – National Software Reference Library (of the United States)

NTFS – New Technology File System

NVMe – Non-Volatile Memory Express

OS – Operating System

PCIe – Peripheral Component Interconnect Express

PST – Personal Storage Table

List of Abbreviations

RAM – Random Access Memory

RDP – Windows Remote Desktop

SATA – Serial Advanced Technology Attachment

SHA1 – Secure Hash Algorithm 1

SHA256 – Secure Hash Algorithm 256-bit

SQL – Structured Query Language

SSD – Solid State Drive

TOR – The Onion Ring

USB – Universal Serial Bus

VNC – Virtual Network Computing

VPN – Virtual Private Network

WSL – Windows Subsystem for Linux

Introduction

Technology transformed the way that humans conduct their day-to-day activities, enabling the communication, trade, and cooperation worldwide. At the same time, it allowed crimes to be conducted throughout the digital mediums, this way the active role of investigating and preventing those crimes is becoming more relevant for most businesses.

Information security costs to prevent and investigate cybercrime is increasing, according to the latest Gartner, Inc. (May 2021) forecast, the worldwide information security spending in 2021 will be over \$150 billion, a growth of 12.4% even after the impacts of the COVID-19 crisis. While the Verizon Data Breach Investigations Report (2020) indicates that 86% of breaches were financially motivated and 30% involved internal actors.

These reports indicate that not only the number of cybercrimes is increasing, but as Hassan (2019) explains, also the “awareness of the importance of data on the part of authorities and business corporations has encouraged them to act and develop different digital forensics tools and methodologies”.

Those criminal actions leave traces, so called “forensic artifacts”, that are, in summary, any digital object with any relevance on forensic investigations, it holds footprints of usage of files, folders, devices, operating systems and/or tools, also defined by the norm ISO/IEC 27037:2012 as “information or data, stored or transmitted in binary form that may be relied on as evidence”.

Digital forensic investigations in the recent years have been facing an exponential growth of the number of forensic artifacts – according to the Digital Forensics Artifact Repository (2019), as in November 2019 there are over 525 artifacts, while the Artifact Genome Project claims to have catalogued 1,209 artifacts in May 2021.

In addition to the increasing number of forensic artifacts there is also the extensive growth of storage size and generated user data that impacts the collection, processing, and analysis time. A paper published by IDC indicates that the “Global Datasphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025”.

Following the same logic, BUNTING (2012) brings a bit more context about how forensic artifacts have been evolving, as he explains, in order to transform the operating

Introduction

systems in an easier to use, the logical way was to “store even more information about the user, such as their actions, preferences, and credentials. The result of such data storage is an environment that is loaded with artifacts, which take the form of logs, files, lists, passwords, caches, history, recently-used lists, and other data.”

Considering that these trace elements are available in the most variable forms, from raw files, logs, databases, file slack spaces and even unallocated areas of the storage device, the methodology differs to recover each type of information, and with the large types of artifacts available, it is relevant to discuss which ones are more likely to contribute to the investigation.

Triage on Electronic Stored Information (ESI) can benefit of the collection and analysis of these digital vestiges, as it brings valuable pieces of information that can help to compose the fact-finding puzzle.

Preserving electronic stored information (ESI) is the most critical part of a forensic investigation, it should be done on the first possible moment to avoid latent risk of data loss, either due to actions intending to destroy, obfuscate, manipulate or to the volatile nature of this information.

Incident response techniques can be applied to gather specific forensic artifacts on an ad-hoc basis, requiring that the investigator knows exactly which aspect to analyze, what can leave some potential evidence uncovered.

Multiple tools are available and can be used in order to reduce the human tasks on collection and parsing, but those tools are spread across different platforms, each one providing results in a different format and requiring a broad knowledge of the operating system and potentially used applications.

With greater number of companies aware of the benefits of conducting corporate investigations and the need to meet new legal requirements to ensure data privacy and security compliance, the number of investigated devices has been increasing, leading investigators to rethink their approaches on how to proceed with the EDRM steps and reduce the time spent in manual activities.

The same difficulties are reflected in public investigations, where the awareness of these methodologies bring more demand of forensic investigations in order to solve crimes.

Research delimitation

The present research intends to investigate a potential alternative workflow to forensic acquisitions of electronic stored information that would enable automated parallel parsing of windows operating system artifacts.

In order to accomplish that the research will be done through the analysis of different artifact types that can be identified in windows devices, streamline which ones can be extracted utilizing open-source tools and define which ones have valuable information with low performance impact.

Furthermore, this research is intended to provide a method to reduce the processing times and identify additional insights during the collection step in a fashion that would allow crime investigations to be performed faster providing that more relevant evidence is identified.

I. Objectives

The main objective of this research is to develop a tool that enables the user to perform forensic collections with simultaneous pre-processing of multiple forensic artifacts in order to generate a report that can bring insights to the investigator even before the end of the forensic acquisition.

The selected forensic artifacts must have a small size and be in a ready state, meaning that they should not be encrypted, corrupted, deleted or behind any digital barrier that could increase the processing time.

The tool will be designed to work on Ubuntu 20.10 (or later) based systems and perform a logical collection of an unencrypted source disk with (preferably) Windows 10 (utilizing the default NTFS or EXFAT partition systems) and the execution of the tool in a different scenario may not provide the expected results.

Below are described the minimal specifications that are expected at the delivered final version:

- Run on Linux (Ubuntu based) operating system.
- Preserve a full physical forensic copy of the source disk.
- Preserve a logical forensic copy of selected forensic artifacts and enclose it in a safe container.

Research delimitation

- Parse, Index and analyze such artifacts in parallel with the physical collection.
- Parsing should include artifacts of MS Windows 10.
- Extensibility to allow user to add/remove different artifacts other than the already built in.
- Capability of utilizing RAM disks and multi-threading.
- Write-Protect the source disk.
- Generate a summary report in readable format with basic information on index.
- Generate execution logs and hash logs to ensure the forensic-sound preservation of all available data.

In order to accomplish the development of the tool it is essential to gather a list of top-edge open-source forensic tools, libraries and techniques used to identify various types of forensic artifacts.

II. Justification

During the process of forensic imaging the main speed limitation existent that cannot be surpassed is the input-output bus of the source drive. Meanwhile, the price of high-speed destination drives, and powerful workstations is dropping, according to COUGHLIN (2016), the price per GB of solid-state disks has reached \$0.25 whilst in hard drive disks it was already \$0.033.

Spare resources in collection workstations can be used to gather various forensic artifacts and parse it while the full collection is being performed, providing inputs to the investigator in field.

This information can reveal insights to the investigator on which users have accessed the device, list of potential storage devices that have been connected to that host, encryption, recently used files and tools, indicators of anti-forensic activities among other details that would only be revealed in a next EDRM phase — Processing.

In a regular scenario, the processing step is performed out of the collection field as it is a hardware demanding activity and can take hours to be executed, but in the other hand, it can delay the identification of possible data sources available on site and potentially loose the opportunity to collect such data.

Additionally, this process can be used to extract in advance user created files and mailboxes in a form ready to be indexed in an electronic discovery review platform, allowing other members to start reviewing data while more processing costly activities are performed in the laboratory.

III. Hypothesis and previous research

Developments on past decade have been pushing the imaging speeds to the limit of the destination drives, whilst the main bottleneck is the source drive that can be slow due to multiple factors — old technology, large size or bad physical state.

Some efforts have been made to accelerate imaging speeds, among them the update of the format Expert Witness Forensic (EWF) in 2012 that would allow multiple threading for the verification step and the implementation of a new format named AFF4 (Advanced Forensic Format 4).

The format AFF4 (Advanced Forensic Format 4) has been proposed in 2009 and is still under development. This new format would allow to perform forensic acquisitions using non-linear bitstream images and use the maximum power of the forthcoming storage devices.

Incident response techniques can be applied manually to conduct Early Case Assessment (ECA), for example utilizing tools to extract specific information or conducting basic analysis on the investigated device. If not performed with caution, this will not only bring a slowness to the collection phase, but also increase the risk of damaging the data integrity.

Despite all the efforts, there is still no solution that allow the analysis and processing to be started simultaneously with the forensic collection, this research has the intention to implement a solution to achieve that milestone and quantify the speed difference between a regular forensic acquisition and an acquisition using the new approach.

The hypothesis this research intends to analyze is if a forensic acquisition tool capable to perform forensic collections and parse of artifacts in parallel could speed up the EDRM process.

The required parameters used to evaluate the hypothesis are described as follows:

Research delimitation

1. The total execution time between the start of and the end of the full disk preservation should not exceed 30% of the execution of a preservation that would not involve an ECA preservation.
2. The hash of the full disk image should remain the same both on the scenario with a full disk preservation alone and the scenario with a full disk preservation along an ECA preservation.
3. There should not be errors or bad blocks on the full disk preservation log

Methodological Procedure

At the first moment the research will utilize the method *qualitative* as it will be based in a *literature analysis* of existing frameworks that consolidate forensic artifacts for windows operating system, as well identify available open-source tools with features capable to export those aforementioned artifacts.

In the second stage of the research will be approached with the *empiric* method, as it will consist of the experimental implementation of a hypothesis that would be able to bring a solution to the investigative topic.

As a last point, to compare the stage of the actual technologies against the new approach to measure if the proposed solution support or falsify the previously presented hypothesis, in other words, if it has positive outcomes.

The intended tool is going to be developed using the programming language Python and plans to integrate various existing free and open-source frameworks, libraries, and forensic applications.

A full version of Windows 10 operating system will be installed on an SSD drive and basic user data will be created to be parsed and thus serve as base parameter on all tests.

Acquisitions will be performed using the same data source in more than one destination drives and configuration scenarios to be as close as possible to the reality, enabling and disabling the early case assessment features to compare the time spent with the collection.

All preservation scenarios should be executed utilizing the following baselines:

1. The source drive will be always the same drive in all executions.
2. On all executions with different interfaces, the same storage drives will be utilized.
3. The computer host utilized will be the same in all scenarios without changes on the software or hardware configurations.
4. No additional software will be executing during the preservation other than the strictly necessary of the preservation.

Methodological Procedure

5. The execution will be performed on a device without any network external connection

1. Introduction to Digital Forensics

This chapter intends to introduce the basic concepts that compose the digital forensics lexicon that will be used throughout the hypothesis implementation and analysis steps.

Digital forensics is a multidisciplinary domain that requires a global knowledge of information technology, cybersecurity, law and regulations. As Hassan (2019) explains, “digital forensics was originally developed to aid law enforcement agencies in applying the law and to protect society and businesses from crime.”

In the same essence, the DFRWS (2001) technical report DTR-T0010-01 defined digital forensics as “the use of scientifically derived and proved methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources”.

As the main goal of a computer forensic investigation is to preserve the digital evidence in a forensic-sound manner, – and therefore attend law requirements to be presented on court –, this implicates that the whole process from collection, analysis and presentation must be auditable, repeatable and use court-accepted forensic tools.

This goal is set to attend LOCARD’s exchange principle that every contact leaves a trace, therefore, on each interaction between a user and a device, traces will be left that can be further analyzed to reveal previous events. As SANTOS (2018) describes, this principle works in both directions, meaning that the same way the user incurs the risk of leaving traces during the action, the investigator has to be careful as “everything in touch with the crime scene can directly affect the forensic analysis”.

In the same direction, ELEUTERIO et. al. (2019) highlights that “it is not needed to utilize the computing devices from the site to verify if they contain relevant information to the warrant. In computers, for example, this action can delete/alter data stored even if the user doesn’t take any voluntary action to delete a file”.

The Computer Forensic Tool Testing (CFTT) program maintained by the United States National Institute of Standards and Technology (NIST) has published a paper “Digital Data Acquisition Tool Specification” describing the main requirements to a tool to be considered forensic accepted, from which the below are highlighted:

Introduction to Digital Forensics

- “The tool shall be able to create either a clone of a digital source, or an image of a digital source, or provide the capability for the user to select and then create either a clone or an image of a digital source.”
- “The tool shall completely acquire all visible data sectors from the digital source.”
- “All data sectors acquired by the tool from the digital source shall be accurately acquired.”
- “If there are unresolved errors reading from a digital source then the tool shall notify the user of the error type and the error location.”

In addition to that, the NIST has published a paper “NIST CFTT: Testing Disk Imaging Tools” that defines the specifications for disk imaging tools:

- The tool shall make a bit-stream duplicate or an image of an original disk or partition.
- The tool shall not alter the original disk.
- The tool shall be able to verify the integrity of a disk image file.
- The tool shall log I/O errors.

If any of the aforementioned requisites is not present, then there is a risk that the digital evidence is not admissible in court, be damaged or even destroyed.

Government and agencies like the International Organization for Standardization (ISO) have developed rigorous standards across the time to ensure that this process is followed. The ISO/IEC 27037:2012 defines:

“guidelines for specific activities in the handling of digital evidence, which are identification, collection, acquisition and preservation of potential digital evidence that can be of evidential value.

It provides guidance to individuals with respect to common situations encountered throughout the digital evidence handling process and assists organizations in their disciplinary procedures and in facilitating the exchange of potential digital evidence between jurisdictions.”

The same standards must be followed either in the preservation of data from multiple computing devices, like computers, mobile phones, tablets or basically any

electronic device with ability to store or transmit digital information that could be related to a crime or civil offense.

1.1. Forensic acquisition and preservation types and procedures

First of all, it is essential that the forensic acquisition and preservation observe the applicable laws in the territory where the data is located, meaning that the investigator must only collect data that it has a valid authorization from the competent authority. In fact, if this requirement is not attended the data can be refused in court and moreover civil actions can be taken by the data owner.

In a public investigation this authorization comes in a form of a warrant or a subpoena that will limit exactly which data can be collected and if something out of that scope is identified, an additional warrant must be assigned to that additional evidence.

In a private investigation this can be part of the contract between the individual that is utilizing the organization's devices, likewise the third party that stores the data and the investigation stakeholder.

Secondly, it is indispensable that the whole process is documented using a chain of custody that will record every step of the investigation, since the first individual provided the original source to the responsible to perform the forensic preservation until the moment the data is archived. This document must record the individual that held custody of the evidence, the start time and end time and the destination individual or safe storage facility.

“Digital forensics serves as the mechanism for understanding the technical aspects of the incident, potentially identifying the root cause, and discovering unidentified access or other malicious activity.” (JOHANSEN, 2020)

Generally, the forensic acquisition of data must use some form of write protection to avoid the evidence to be poisoned, either by direct action of the investigator or by uncontrolled results of the operating system and running applications.

Write protection can be ensured either by using a *software* that disable the writing on the source disk or via a *hardware* that will physically filter the input/output and thus prevent the source disk to be written.

ELEUTERIO et. al. (2019) defines forensic hardware as “equipment specialized in performing preservation of various types of media, always ensuring that the preserved content is not altered”.

With this in mind, each case has different aspects to be considered being necessary to determine the urgency of the incident to be investigated, what is the status of the collected devices the type of data to be preserved.

For example, when the investigation subject is a matter of cybercrime that is in progress, the protection of other relevant information assets of the organization can be critical, hence the perpetrator of the crime is still acting and can cause additional damage.

One preventive measure that can be performed in the abovementioned situation is to disable the network connection of the suspected device, but the outcome of this action will, at minimum, generate artifacts at the operating system indicating that this network was disabled, while other malware can trigger additional actions that can damage or destroy relevant evidence.

If the computer was found running and unlocked, a decision might be taken to preserve the volatile memory, running processes and applications, or to shut down the device using the operating system command to turn it off or pull out the plug.

Volatile memory can contain multiple valuable information that are important for incident response cases, and if not preserved at that moment will be lost forever, for example, it can contain traces of recently open processes or passwords that can be used to identify the cause when there is a suspicion of malware or unauthorized remote access, but it's important that this preservation is done carefully and well documented, hence this requires a software to run and save the volatile data and this action from the investigator will generate artifacts and brings a risk to alter the state of the original evidence.

As SANTOS (2018) explains, the investigator should follow the principle of maximum preservation, utilizing always the best available tools and techniques to preserve the evidence to be analyzed.

On the one hand, when the write protection is not possible or the preservation of live data is required, an additional attention to the documentation keeping record of

every detail is a must, particularly when there is a foreseen court action as a result of the preliminary investigation.

On the other hand, when the investigation is purely based on fraud or in files that can be either stored at the device or the cloud, especially if it is part of a business compliance procedure, then the preservation of volatile data can be less relevant than the preservation of the forensic integrity of the device, hence the main evidence is likely to be stored elsewhere than the memory.

Additionally, storage devices have a limited lifespan that can vary depending on the technology used, as ELEUTERIO et. al (2019) explains “the forensic examination must be performed as soon as possible after the collected device is preserved to reduce the risk of data loss caused by the natural device lifespan”.

Investigations can be classified as Public or Private investigations depending on the actors involved and the topic of interest. As a main rule, the investigation is required to be public when it involves a law enforcement agency to investigate a crime, while they can be private when it involves private interests, but it can vary by jurisdiction and laws in place.

Investigations are Public if they are conducted by a public agent representing a country, state, or city, and may follow specific criminal procedures defined by law. Among these procedures, the most important is the requirement of a warrant to allow the data to be seized and preserved, as OETTINGER (2020) emphasizes “while you may not have the search warrant requirement, you cannot seize and analyze private property.”

Public investigations usually are derived from criminal investigations that can often verse about individuals or companies, and the crime solution is the best public interest. And as HASSAN (2019) explains, it may pass three main stages: “complaint, investigation and prosecution”.

Complaint is the crime notification to the entity responsible to conduct the investigation, Investigation is the process that involves the data preservation and analysis, and Prosecution happens when relevant evidence of a criminal offense is identified during the Investigation phase.

Private Investigations are driven by business interests to investigate actions taken by individuals of the organization – for instance policy violations, unauthorized

disclosure of corporate data, damage on property or financial fraud—, or external actors – for example corporate espionage or cyber security attacks.

In any case, corporate investigations can lead to potential criminal or civil public investigations, therefore is necessary to follow digital forensics methodologies to ensure the validity of the process.

Large corporations have policies to conduct the so-called electronic discovery investigations, as HASSAN (2019) details:

“E-discovery is considered an integral part of the justice system, although the implemented digital forensics procedures in civil litigation are somehow different from the one applied in criminal cases in terms of the procedures used to acquire digital evidence, investigatory scope, and the legal consequences of the case.”

In general, both public and private investigations can be motivated by financial gain, either directly (through financial theft, embezzlement), indirectly (through tax evasion or misuse of company resources) or can have a different motivation (for example on harassment, gender discrimination, intellectual property theft or data/reputation damage through service interruption).

1.2. The fraud triangle

A commonly accepted theory that investigates the causes of fraud can well be applied to cybercrime is the so called “fraud triangle” proposed by Donald R Cressey (1953), elucidating that a fraud may happen when at least one of these factors is present:

- **Opportunity:** circumstances that enable the individual/group to conduct the fraud, either by insufficient security controls, weak recordings of trails or the fact that the entity is dealing with a change.
- **Motivation:** incentives or pressures that can change someone’s mindset towards to committing a fraud. Also called Pressure, this can involve an economic need or non-financial benefit.

- Rationalization: a more subjective reasoning that would internally justify a fraud, either by understanding that there is no other solution for the actor's problems, or that he is not being fully recognized by the organization.



Figure 1 Fraud Triangle

CRESSEY (1950) concludes on his theory that “not all trust violators refuse [...] to define themselves as criminals, but all of those encountered refused to give up their ideal or honesty”. This comes along with these three pillars, as the individual is aware that his actions are possibly illegal, but one has rationalized that is a justifiable risk.

During periods of crisis, when austerity measures like layoffs, salary reduction or promotions freeze happens widely is the perfect fertile land for the development of frauds, when more individuals see themselves as undervalued or the next on the line. Fortunately, when a crime occurs utilizing electronic devices, traces of the execution are left in a variety of forms.

Electronic devices can be incidentally used to conduct a traditional type of crime, as MACHADO AND ELEUTÉRIO (2019) elucidates, “when a document is falsified through an image editor or is altered with the use of a ballpoint pen [...] the computer is associated to the *modus operandi* of the crime”, meaning that the device is merely a supporting tool that enabled the crime, but not the only way to perform it.

1.3. Basic concepts

Identifying the relevant digital devices while onsite is primordial to avoid losing the opportunity to collect it, as in some cases this may be the only opportunity either by the legal right to collect be limited to a certain date or due to the volatile nature of many digital data.

In order to ensure the admissibility in court, there is also the prerequisite that all the documentation is correct, this includes the creation of a Chain of Custody, photographs of the evidence and place where it was collected, logs of acquisition and hash ensuring that the source matches the destination copy.

All these aforementioned concepts are essential part of the digital forensics lexicon that will be used in this research and will be briefly explained below.

1.3.1. Forensic Artifact

Is the generic term to refer to the various footprints left by the usage of an operating system, application, or device object of the analysis. Forensic Artifact is a broad term that englobes all potential *evidence* within the device.

The wide diversity of artifacts imposes a challenge to the investigator that must gather this information and link it to the case. New file types emerge every day, software changes the way it stores its configurations and usage logs, bugs can cause data to be lost.

Projects like the “Digital Forensics Artifact Repository” (created as a fork of the Google Rapid Response project) and the “Artifact Genome Project” (created by the University of New Heaven in cooperation with multiple cyber security and forensic companies) are valuable tools attempting to keep track of this myriad of evidence types and locations. They use the community potential to bring the knowledge together, so the members do not have to rebuild the wheel every time they identify new evidence.

1.3.2. Metadata

Most digital files have inherent information that tell details about the item itself, this information is called metadata and can include the creation date, last execution, author and so on.

Some of this information is preserved on the file as part of its content (for example the headers of an email message), in the operating system (with details of the creation and execution times) or in databases that are part of the application that generated the file (for example, messages on chat applications).

In time, HASSAN (2019) details that in some cases the metadata is stored in a separate file, which is the case for the Recycle Bin files, while others can have this data embedded that is created by the user, as “an MS Word file might include author name, organization name, computer name, date/time created, and comments”. While other files can have metadata created automatically by the operating system or the application used to generate it, for example, pictures captured with a phone can store “GPS coordination of a specific photo, captured camera type, and resolution”.

Preserving the metadata is important not only for the insights that this can bring to the case, but also crucial to proof the forensic sound of the acquisition in case of a court trial.

This preservation must be performed on the first possible moment to prevent the data alteration, corruption or destruction, this principle is also called as “forensic readiness” and, as HASSAN (2019) elucidates “should proceed without disrupting current operations to minimize investigation cost”.

In the same direction, SANTOS (2018) defines forensic readiness as a “organization strategy willing to be always ready to resolve computer forensics incidents [...] the organization must have procedures in place as well as trained personnel in computer forensics to start acting”.

In other words, this forensic readiness is a principle focused on corporate investigations, but can be well applied to public investigations, as the need of a trained team is essential to properly preserve the evidence with the lowest time ensuring the forensic sound process.

This concept is relevant to this paper as the copy of the whole file can take a long time, the parsing of its metadata can be accomplished quickly, in particular to the metadata available in databases or in the Main File Table (MFT).

1.3.3. Algorithmic Hash

The unidirectional hashing functions are mathematic algorithms that calculate a virtually unique value for the evidence – works on both full disk images and individual files. This value can be calculated only in one direction (from the evidence to the hash) and must adhere some international standards to ensure that one file can have only one possible hash and one hash can represent only one file.

ELEUTERIO et. al (2019) explains that hash functions “generate, from an input of any size, an output with a fixed size. In other words, a large volume of information (original information) is transformed in a small sequence of bits (hash value)”.

Hashing works as a virtual fingerprint for the forensic artifacts, because if a file has been modified the value will not match with the preserved copy, thus, while the hash remains intact it works as a proof that the acquired data has not been tampered, therefore it is an essential concept in the digital forensics.

This goes in the same direction as HASSAN (2019) explains “Hash works by implementing a hash function to convert a digital file (input) into a fixed string value (output); the resultant hash value is unique and cannot be generated again using other file or piece of data”.

Worth mentioning that the hash algorithmic functions are not impossible to fail, as there has been at least one case of collision (two different files generating the same hash string), but the chances are very low, according to BUNTING (2013) the “odds of any two dissimilar files having the same MD5, better known as a hash collision, is one in 2^{128} .”

Despite being highly improbable to happen, it has been proven that collisions can occur, a paper published in 2005 by Xiaoyun Wang and Hongbo Yu described an algorithm capable to identify a few cases where different inputs generate the same MD5 hash, and this has been extrapolated by SELINGER in 2011 to generate executable files with the same hash but different functions.

There are multiple implementations of hash algorithms, but the most broadly used in the forensics tools on the market are the MD5 (fast, but as mentioned had a reported collision), SHA1 (string of 160 bits, also had reported collision) and SHA256 (string of 256 bits).

Some forensic tools can calculate the hash using multiple algorithms to ensure that even if there is one collision on one of the cases, the other hash is available to counterproof.

Other usage of the hash is to identify known applications and operating system files (so they don't need to be analyzed) or known malware (so they can be spotted automatically), this can be achieved by using one hash set database, for example the National Software Reference Library (NSRL), provided by the National Institute of Standards and Technology (NIST).

As explained by Philipp et. al. (2005) "If you compare the reference hashes provided by the NSRL to the system files on the computer, you will be able to tell if any of the vital files have been modified or changed, an indication that they have been taken over by malware."

This concept is important for this paper as the preservation of the files must be forensic sound and the analysis should not change the preserved data.

1.3.4. Chain of Custody

Maintaining a trail that tracks crucial information of an evidence is important, especially in cases where multiple individuals and their respective devices are being investigated, but not only important, it is a law requirement in many countries so that the preservation is forensically sound.

According to BUNTING (2013), the Chain of Custody is "a concept in jurisprudence that applies to the process by which evidence is handled to assure its integrity as proof of a fact in court".

HASSAN (2019) defines the Chain of Custody as an "integral part" of the digital forensics investigations which "ultimate goal" is to ensure the integrity of digital evidence by "knowing all persons who were in contact with this evidence from its acquisition to its presentation in a court".

While Philipp et. al (2005) explains that the preservation method doesn't necessarily means a paper trail, it can be "from keeping handwritten manual logs to using software databases and bar-coding systems. What method you use depends on the size of your organization and the amount of evidence you typically handle".

The Chain of Custody must contain all steps of the evidence handling from the moment it was identified, preserved and all movements until the end of the investigation to ensure that all parties involved are properly identified and the custody was not jeopardized.

It must also preserve data to identify the preserved device (e.g.: a unique number to identify the evidence item, a description of what is the item, the manufacturer, model and serial number) as well as the changes of custody, including the date/time when the custody was transferred from one individual/entity to another and the reason of the movement.

JOHANSEN (2017) explains that “moving a piece of evidence should never be done without a reason” and the custody holders can “can either be a person or a storage place. For example, if an analyst has seized a hard drive and is moving it to a secure storage locker”.

In addition to the chain of custody, it is recommended to have a general log of collections when the investigation involves multiple individuals or devices, as well, each acquisition should have an associated acquisition log created by the tool utilized to preserve the data and photos of the hardware or screenshots of the operating system depending on the case.

The use of Chain of Custody is a relevant concept for this paper as the preservation of the files that will be present in the final report must accomplish the evidence trail requirements, including the creation of execution logs that contains details about the preserved evidence and examiner name.

1.3.5. Windows Registry

Microsoft Windows have a special database that stores configurations from multiple applications and functions of the operating system, this central repository is named “Windows Registry”.

As BUNTING (2013) highlights, “the registry is a gold mine of forensic evidence”, because at the same time that many important evidence is stored in the registry, Microsoft “discourages users, administrators included, from accessing or modifying the registry”, and as these configurations are vital to the good working of the operating system in general, backups are generated automatically.

This will be relevant for many of the analysis performed by this paper, as retrieving data from this database is relatively fast and can provide multiple information, for example the list of users in the device, recent office opened files, list of attached USB devices, last login and more that will be investigated in the proper analysis of each artifact.

1.3.6. File Signature

In most Operating Systems, the files might have an associated extension on the name to facilitate the identification of the file type to the user, as well to identify which program is best used to open it.

This extension is often associated to the program that generated the file or that can be used to open this file, but this is not the only requisite to identify the file type, as BUNTING (2013) elucidates, many files generated by known market tools “have been standardized and have unique file signatures or headers that precede their data”.

These known headers work as file signatures, and many forensic tools are capable to compare a database of known headers against the file extension to identify possible mismatches – which is an indication that the file has been renamed or generated outside of the standard.

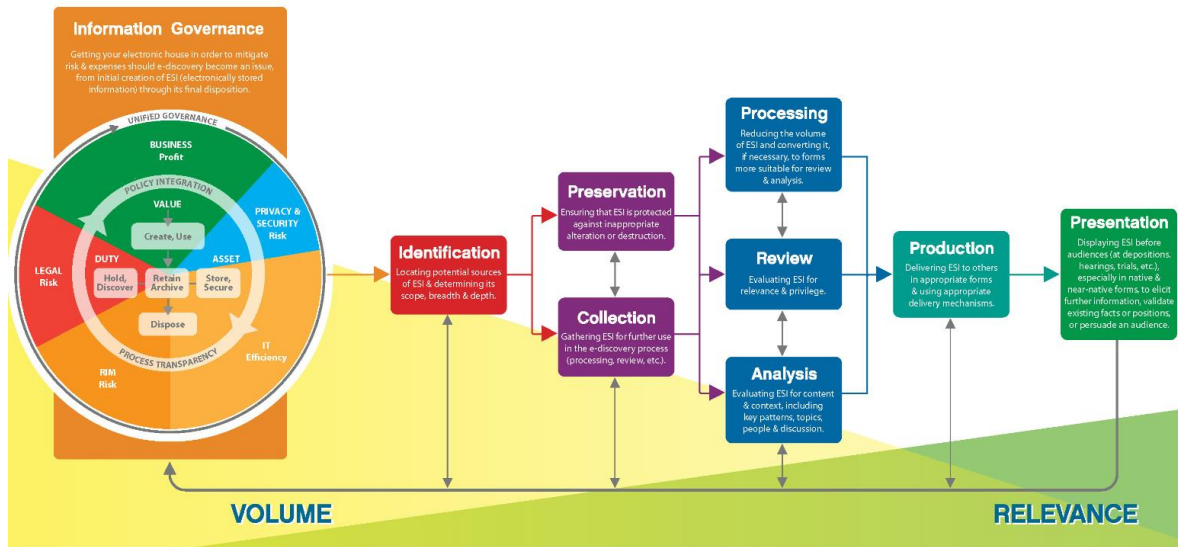
“File signature analysis will compare files, their extensions, and their headers to a known database of file signatures and extensions and report the results.” (BUNTING, 2013)

Additionally, the use of file signatures against the unallocated disk area or a memory dump can help to identify and recover deleted content. This will be explored with more details in the hibernation, pagination and memory dump parsing topic in this paper.

1.4. Digital Forensics Investigation Frameworks

There are a few frameworks that define the baseline flow for a digital investigation, being the two most popular ones the Digital Forensics Research Workshop (DFRWS) and the Electronic Discovery Reference Model (EDRM), with multiple

similarities, they both have frequent updates to attend the market demands and new processes.



Copyright ©2020, EDRM Global, Inc., Creative Commons Attribution 4.0 International, edrm.net

Figure 2 EDRM Framework (EDRM.net, 2020)

The EDRM framework has six steps (Information Governance, Identification, Preservation/Collection, Processing/Review/Analysis, Production and Presentation) while the DFRWS framework is composed of the following six steps: Identification, Preservation, Collection, Examination, Analysis and Presentation.

As HASSAN (2019) highlights:

“EDRM (www.edrm.net) is a popular standard for improving e-discovery and information governance. This is a conceptual standard for the e-discovery process that outlines standards for the recovery and discovery of digital data during an investigation, litigation, or similar proceeding”.

Besides their similarities, the main difference between them is that the DFRWS is more used by government agencies and public administration on digital forensics and incident response investigations, while the EDRM is more used by corporations and private sector investigations, in particular involving electronic discovery analysis.



Figure 3 DFRWS Framework (DFRWS.org, 2001)

For this research we will focus on the DRFWS framework, which is in the market since 2001 and provide a more high-level overview of the most important steps, as described on the following subtopics.

1.4.1. Identification

The first step of the investigative process, generally speaking has the main objective to acknowledge the basic details of the incident and identify which are the main evidence that could be relevant for the investigation.

In this step the most relevant decisions of the investigation are taken, including the limitation of the scope, interview list and prioritization of the Electronic System Information (ESI) data sources to be preserved.

SANTOS (2018) defines this step as “before starting the analysis, the investigator must identify the evidence and its location”.

Among the identification it is essential to also verify potential peripheral, external devices and loose paper that can contain relevant data, for example optical media (CD/DVD) inside the device readers, pen drives connected through USB and small devices that could be hidden in drawers, cabinets, dust bin and safes.

1.4.2. Preservation

Preservation is the step where the data is seized in order to prevent any type of modification or deletion, as well preserving the completeness of the content at the time of the investigation and the chain of custody.

As JOHANSEN (2017) highlights, “It is also critical that any users are not allowed to access a suspect system. This ensures that users do not deliberately or inadvertently taint the evidence”.

The seizure procedure may be different upon the evidence type (computer, mobile device, removable drive, etc.), the evidence state (if it is connected to a power supply or being used at the moment, as well if other network resources depend on that device), the investigation nature and the volatility of the data.

In contrast, corporate investigations can implement preservation by taking measures to prevent data modification or deletion that not necessarily involve physically

locking a device, but rather starting policies to prevent this alteration or enabling automatic backups.

As SHOOK (2014) argues:

“the preservation step represents the requirement that discoverable ESI should not be deleted. Improperly lost or deleted ESI may give rise to a claim for spoliation, which can result in an array of sanctions based upon the culpability for its loss and the importance of the lost data.”

Chiefly, if a standard electronic device is identified offline, it is optional to either seal it and take to a forensic laboratory for collection without external interference and with a reduced time pressure or to conduct the collection on spot, avoiding the risk of data being lost or damaged on transportation.

On the contrary, if the data is stored in an electronic device that is identified active and the investigation verse over a subject that volatile memory information is essential, the preservation of this volatile data should be done before shutting it down.

As detailed by HASSAN (2019):

“Upon arriving to the crime scene, the suspect digital device should be examined by a well-trained technician to ensure the digital evidence is acquired/preserved in a forensically sound manner.”

To put it differently, it is important to perform the preservation in a forensically sound manner, with legal permission from the proper authority, upon the first possible moment to prevent the risk of the data being tainted directly or indirectly, but this preservation must also take into account that the digital device that contains that data might be a critical component of an organization that shouldn't be taken offline unless there is an inherent risk of not doing it.

Additionally, depending on the investigation nature it could be relevant to state new policies and controls to ensure that logs are being generated to monitor potential new outcomes of the investigation, along with the prevention that specific logs or files to be deleted.

1.4.3. Collection

Different from the Preservation step, which focus on separating the devices and ensuring that the evidence is preserved from any sorts of damage, the Collection phase is when the data is actually copied in a forensic sound manner to a secondary storage drive.

Depending on the nature of the investigation these two phases can overlap, as the device can be preserved and immediately the collection be performed on site.

Collection can be defined as duplication of the original data in a forensic sound manner, this can be done by a copy bit-by-bit from one storage device to another (also known as physical drive copy), a copy bit-by-bit from one storage device to a forensic image container (also known as physical image copy), a copy of the visible content (logical image copy) or the last, a simple copy of the files with well documented process.

Among the aspects to be taken in account, the preservation of the chain of custody is the most important, as JOHANSEN (2017) reminds “any break in the chain of custody can lead to the piece of evidence being excluded from ever being admitted into the proceedings”.

In all cases, the copy must be performed using forensic processes, preserving as much of the metadata as possible and leaving the minimum changes on the operating system files, and the source hard drive should be write-protected (with a few exceptions, for example when it is not possible to create the image with the device offline because of encryption or imaging of cloud data that requires a different process).

“If the digital forensics examiner does not take care to preserve the evidence at this stage, there is the possibility of contamination that would result in the evidence being unreliable or unusable.” (GERARD JOHANSEN, 2020).

In general, it is preferred to produce a physical image copy, as this copy preserves all data visible and not visible to the operating system and provides liberty to the investigator, as the data on this image is not object of tampering and as consequence the process to create working copies would be as simple as copy paste it into a different destination.

In this direction, HASSAN (2019) emphasizes that “Examiners usually use hardware duplicators or software imaging tools like the DD command in Linux to duplicate drives”.

The decision between a software imaging tool and a hardware duplicator should be considered based on subjective parameters, as long as the minimal features are present: write protection on the source, capacity to perform a copy bit by bit, capacity to preserve metadata, capacity to generate an audit trail and finally, capacity to generate some hash authentication.

Write protection is the capacity to protect the data source against unintended writes that could be generated either by human interaction or by the applications in the operating system where it is attached to, this can be achieved, again, via software or hardware:

- Hardware write-blockers: physically intercepts the electric data signals that could alter the source. This is possible when the imaged storage media can be removed from the host device. Write blockers can work independent of the operating system or even be part of the imaging duplicator hardware.

ELEUTERIO et. al (2019) explains that “hardware write blockers for hard drives are more common and easy to utilize. Connected between the hard drive and the computer, this sort of equipment ensures that data will not be written to the respective drive, thus enabling only reading”.

Additionally, there is a type of specialized hardware commonly referred as forensic duplicator, that can perform both the write-protection and the data copy.

- Software write-blockers: is a logical protection, similarly to the hardware, it avoids the writing on the source data, but this is OS dependent, on Linux it is possible to avoid that all external devices are mounted unless specified, while on Windows it depends on the connection port being used. Specialized software, like *FastBloc SE* from OpenText, can prevent the writing by intercepting the write calls at system level. One advantage is that it can be used on medias that cannot be removed from the host device.

Likewise, there is a category of specialized software usually referred as Forensic Operating System, usually based on Linux, that have some kernel level protection to prevent that a drive will be written without an active action from the investigator.

The forensics analysis should always be performed on the forensic image rather the original evidence, which is usually an evidence container that presents multiple features to ensure that the original data is properly preserved with all its metadata and structure, among them are an acquisition log (that maintains the audit trail including all errors, sizes, hash, start and end time, and so on), as well an optional encryption and compression.

ELEUTERIO et. al (2019) explains that “because of how fragile and sensible digital media and storage devices can be, forensic examinations must, if possible, be performed in faithful copies from the original material”.

The most common forensic image formats are described below:

- **Raw images:** are not a forensic image *per se*, but a copy bit by bit from an evidence source to a destination, is the oldest format to be used in forensic investigations, as it can be achieved using a tool native on Unix operating system named “dd”.

“The dd utility was created in the 1970s on early UNIX systems for byte-order conversion and block copying. [...] The program simply takes blocks of data from a source, optionally performs a conversion or transformation, and then places the blocks in a specified destination (on another device or in a file)” (NIKKEL, 2016)

There are a few variants of the “dd” utility that have been designed with a forensic objective, worth mentioning “dcfldd” and “dc3dd”, that offer image verification, progress monitoring, logging and hashing.

- **EnCase Expert Witness Format (EWF):** evidence container format developed by OpenText (former Guidance Software) that supports metadata preservation, hashing, splitting of files, password protection, compression and verification. This is one of the standards more used in the private sectors as is fully compatible with most forensic acquisition tools and is the default for the EnCase forensic software suite. The

generated files of a forensic image from a full disk are usually with the extension “.E01” or more recently “.EX01” after an upgrade on the technology, while the forensic image of a logical folder/file is usually saved as .L01 (Logical Evidence File, or LEF).

“A reverse engineered, open-source library and tools, libewf was created in 2006 by Joachim Metz and support can be compiled into Sleuth Kit” (NIKKEL, 2016)

- **AD1 (FTK):** other proprietary evidence container, the AccessData FTK (Forensic ToolKit) is a competitor of EnCase that has almost the same characteristics, with a tool free (but not open source) named “FTK Imager”, it can acquire both Logical and Physical file systems as well as network share data.
- **Advanced Forensic Format (AFF):** open-source format that meets all the features as the other proprietary formats. The most recent stable version is AFFLIBv3, but a new version AFF4 is under development since 2009 (COHEN, et. al. 2009) and the standard paper promises new features that can transform the industry, as it would the universal preservation of logical and physical formats utilizing image streams and map streams, resulting in a smaller format with a more organized data – subsequently faster indexing.

1.4.4. Examination

Also known as pre-processing or filtering phase, this step consists in utilizing forensic analysis tools to identify reduce the volume of data to be analyzed, as well to parse and link potential artifacts from the various data sources.

ELEUTERIO et. al (2019) defines this step as “basically assembly of all information present on the data copied on the previous preservation steps”.

The examination can be performed Live (usually on cyber security incident responses the investigator may access the device live and preserve valuable information or prevent additional destruction) or *postmortem* (also called dead box analysis, is performed when the system is identified offline or does not require urgent intervention).

CANDEIAS (2015) clarifies that the forensic tools can be “characterized according to the preservation approach, being classified as *post-mortem* analysis or *live forensics*. [...] volatile information is not captured during a post-mortem preservation, hence the device is turned off”.

As LOSSIO (2021) highlights, “the use of the storage device itself to perform forensic procedures may be faster, but it can make the entire process of preserving the evidence perish”, therefore the decision to proceed with a Live or a *postmortem* preservation must be taken with clear knowledge of the risks and benefits of each approach.

Some of the common tasks of the examination phase may include:

- Indexing: consists in converting the readable content of the forensic image in a searchable format, usually storing the information in a database that can be queried (with words or other criteria) to facilitate the analysis step.

As ELEUTERIO et. al. (2019) defines, the indexing of “data present on the storage devices consists of reading all data (bits) from the device, identify all alphanumeric results and organize it so the data can be quickly recovered”.

- Hash analysis: calculate the unique electronic signature (hash) of the items and compare it with known files that can be either excluded from the analysis (for example files that are intact parts of the operating system or common applications) or be highlighted as relevant evidence (for example a known file from a data leak or a known malware on a cyber security incident response).
- Signature analysis: comparison of the header (first bits of information from a file) with a known list of headers to identify the potential file type and highlight mismatches with file extensions.
- Identify and parse mailboxes: when email messages are stored in a local storage device, they commonly have a format that can be parsed and indexed to become searchable.
- Internet Browser and Chat Parsing: most common internet browsers utilize some format of database to store the history of accessed URLs and

use some specific directory to temporarily store the accessed websites, the same applies to chat applications. This data can be converted in a format readable and searchable.

- Identification of common applications: when an application is installed on a device it leaves traces in multiple locations and these traces can be easily identified and organized, enabling the investigator to find malicious usage.
- Recover deleted files: when a user deletes a file from the file system it usually remains on the unallocated area. Comparing a known file signature can often identify files on the unallocated area and recover these files even after deletion.
- Parse Registries: windows operating system has a special database that saves multiple types of configurations from the OS, applications and user preferences, these are called Registry Hives and may contain multiple relevant bits of information, for example the last shutdown date, traces of deleted software, previously connected USB Devices, recently opened files, etc.
- Recover data from memory dumps, pagination and hibernation: operating systems can save parts of the RAM in the local storage when the amount of available memory is not sufficient for the high intensity tasks (also known as pagination), or when the device is shut down for quickly starting with the same state (known as hibernation) or when an issue happens (memory dumps), and same techniques of recovering deleted files can be applied to recover information on these artifacts.

There are many tools specialized in parsing these artifacts, for example the proprietary EnCase, Forensic Toolkit (FTK), Belkasoft Evidence Center (BEC) and the open-source Autopsy.

In this regards, HASSAN (2019) explains that “Forensic tools can also perform searches within the acquired image file using keyword search terms or phrases. This will effectively speed up the investigation and help investigators to find relevant information quickly.”

With the evidence indexed it is possible to run searches to identify more easily the artifacts related to the specific investigation, in particular when the case objective is already known.

1.4.5. Analysis

After all the data is indexed and the various artifacts are parsed, begins the analysis phase, that consists in the investigator search for the relevant artifacts in the evidence, this can be done using multiple approaches.

“Once the examination phase has extracted potentially relevant pieces of data, the digital forensic examiner then analyzes the data in light of any other relevant data obtained.” (JOHANSEN, 2020).

The analysis of the data may be widely different depending on the scope of the investigation and the potential evidence found and may or may not involve analysis of different aspects of the data. To put it differently, in an investigation versing about a financial fraud may not require analysis of network logs whereas the most relevant data tend to be on mailboxes, logical files or databases.

Some of the artifacts that can be object of the analysis are described below:

- Search for keywords: as all files from the filesystem have been indexed on the Examination phase, it is possible to define a set of potential keywords that can lead to specific artifacts. Techniques like Regular Expressions, Lucene (depending on the forensic tool), Concept Searching and Similar Document Detection can be applied to speed up the process and filter the results properly.
ELEUTERIO et. al. (2019) highlights that “searching the content of a storage device utilizing keywords is a very efficient method to identify interesting files. Once the data is indexed, multiple searches can be quickly performed, because, the content of the drive is now structured, it will not be necessary to go through all content every time a new keyword needs to be searched”.
- Mailboxes: on white-collar crimes, the investigation of email exchange might be crucial to identify potential evidence, the same can apply to social engineering attacks and where, as JOHANSEN (2017) explains

“malicious insiders may have sent or received communication that was inappropriate or violated company policy”.

- Search for known anomalies: examples may include hidden data, processes and emails running overnight and known malware files/indicators.
- Search for specific periods of time: filter loose files or email messages on metadata fields that can delimitate some timeframe or ordinated from most recent to last recent and vice versa.
- Identify the most recently opened files: identically to the previous statement, in various scenarios the most relevant data can be identified on the recent files folder or on registries that store the recent activities on windows.
- Identify connected external storage: registry hives on Windows can store the list of external devices connected through USB ports, and this include not only storage devices (thumb drives, external drives), but also any sort of peripheral (smartphones, tablets, printers, etc.).
- Analyze memory and memory dumps: Windows can store parts of the RAM in disk in various situations, from them it's wise to highlight the pagination (usually used when the amount of RAM is not sufficient to hold all data from the active processes), the hibernation (when a computer is going to be shutdown it can store the active processes and files to enable a quick return to activity) and the memory dumps (usually when a hardware or software failure occur, the system tries to save the contents of the memory at the moment to future investigation). The memory dumps can contain active documents (even if they have not been saved), open websites, chat messages, typed passwords, copied text from the buffer and more.
- Find encrypted files: a user or application can encrypt files to protect them from other users' access, some techniques can be applied depending on the encryption algorithm to identify the type or even to bypass the protection.

Bear in mind that, as ELEURETIO et. al. (2019) emphasizes “some forensic suites, like FTK, can detect the presence of some types of encryption, but not all possible cases”.

- Most recent or Suspicious websites: as presented on topic, modern internet browsers store website information on *SQLite* databases, while internet explorer stores this information on windows registry. Within these artifacts is possible to identify suspicious websites accessed, downloaded items and favorite items.
- Read documents and messages: extracting the content of documents and email messages and ingesting it on a database enables the investigator to easily navigate through the items, sort them via metadata or search specific terms.

Some of these tasks can be automated so the analysis is performed with reduced effort by presenting the most relevant artifacts first, most forensic suites in the market are capable to parse the information, but it is responsibility of the investigator to filter it.

To sum up, the investigator will have to interpret each aspect of the examined evidence and construct a logical line of thinking to conclude which individual artifacts have connection to the specific crime.

1.4.6. Presentation

Also referred as Reporting, this phase of the investigation is the final piece, where all the substantial evidence is put together in an unbiased way to address the facts related to the incident.

This is usually put in a form of a written report with description of each relevant detail in a clear and concise way along with all supporting evidence that can be referenced with their respective logs, hashes and chain of custody.

The report must also include a description of the process, in other words, the description of all tools used to preserve and analyze the digital evidence (their function on the investigation along with version and date of the execution), description of the evidence (preservation method, description of the hardware or any unique

identification details or pictures), dictionary of forensic expressions that may not be known by the readers and ultimately a conclusion.

Some investigations can involve the presentation of the results by testifying in a criminal or civil proceeding, as JOHANSEN (2017) explains, “It is during this testimony that the forensic examiner will be required to present the facts of the forensic examination, in much the same dispassionate manner as the report. The examiner will be required to present facts and conclusions without bias and may be limited as to what opinions they testify to”.

Private investigations may require the results to be presented to internal members of the company that hired the investigation, to the board or to the law counsel.

1.5. Early Case Assessment

The proliferation of computing devices and cloud technologies have enabled the communication and storage of multiple aspects of human activities. Many of this data can be considered relevant to an investigation, thus the preservation and review become necessary.

A problem arises when the volume of data to be preserved is so large that the investigators have to look for, as per Dijk et al (2015) words “a needle in the haystack but also that they do not know what the needle looks like”.

As Martin and Cendrowski (2014) emphasize, “from a litigation and investigation perspective, there is more data available about a subject than at any previous time in history”, and this is the tip of the iceberg, as not only the volume of data is sparse, but also not standardized.

A simple communication can store part of this data on a smartphone, an attachment on an email message, a subsequent file saved on a hard drive and edited later on the cloud and none of these sources need to use a common file format or the same encryption protocols.

In this situation, the volume of data present on a digital forensics investigation can be very high, as SHOOK (2014) highlights: “workers create large volumes of data – word processing files, spreadsheets, presentations and even small databases. In addition, these computers create some data on their own: cookies from Internet sites

that have been visited, logs reflecting user activities such as files that have been opened and edited, etc.”.

This is when Early Case Assessment (ECA) become handful, as it can provide tools to enable the easy and quick understanding of the data outline before the actual data review, saving time and providing additional insights to the investigator.

1.5.1. Definition

Early Case Assessment (ECA) is a type of triage technique that is used to provide a hint of the data available in a data source in order to address common questions on the preliminary phase of the investigation.

The most common definition of ECA is that this is an industry-specific nomenclature used to “describe a variety of tools or methods for investigating and quickly learning about a Document Collection for the purposes of estimating the risk(s) and cost(s) of pursuing a particular legal course of action.” (Grossman and Cormack, 2013)

In the same direction, Dijk et al (2015) argue that ECA is a “type of exploratory search”, indicating that the data is retrieved to bring insights when the investigator does not have a clear understanding of which data is being analyzed beforehand.

LAYKIN (2013) clarifies that ECA has evolved around the litigation processes “where a company that is in receipt of a lawsuit performs early diligence and early investigative work to determine the extent of its exposure, its ability to respond, and to assess whether it is in its best interest to settle the lawsuit.”

The term ECA is more commonly used in eDiscovery investigations, while in computer forensics and cybersecurity incident response fields the Early Case Assessment is sometimes referred as “triage” and has the main objective to prioritize and filter the data to the investigation.

“Computer forensic investigators can assist with this early case assessment work by providing objective insight into the disposition of electronic data and assist the team in formulating its conclusions so that a proper response to the lawsuit can be formed, whether it be a settlement offer or the response will take the company into full-blown litigation.” (LAYKIN, 2013)

Grossman and Cormack (2013) define eDiscovery (short for electronic Discovery) as “the process of identifying, preserving, collecting, processing, searching, reviewing, and producing Electronically Stored Information that may be Relevant to a civil, criminal, or regulatory matter.”

To deal with hundreds of thousands of documents, the logic action is to try to categorize the known data, identify milestones and key individuals that are involved on the actions.

In the same direction, LAYIKIN (2013) highlights that the transition from computer forensics to eDiscovery “required ordinary lawyers of every stripe and all of their support personnel to suddenly be conversant in the world of technology”.

A great example of when an ECA can be applied was described by Sondhi S., Arora R. (2016):

“an attorney who is working on a case involving wrongful termination of an employee [...]. After processing the documents and email [...], it may be discovered that contrary to the expectation, over 90% of the information processed pertains to irrelevant or non-responsive content such as sales interactions, marketing campaigns and pictures of corporate events.”

In this scenario ECA can be used to filter the data before the analysis to include only documents and email messages from within the organization or more specifically for the involved suspects instead of process the whole dataset.

1.5.2. ECA and Digital Forensics: Advantages, Challenges and implementation

Early Case Assessment procedures are essential to face the ever-growing digital artifacts that can be subject of a forensic investigation. With the adequate execution, it is possible to reduce the volume of manually reviewed data, immediately retrieve relevant data and discover additional insights.

The main advantages that can be accomplished by the use of ECA to gather information identified on the collected devices are described below:

- High level knowledge of the data outline: techniques can be applied to identify the overall metrics of the data present within the collected devices, as well correlate with other devices from the same investigation

and data visualization tools can improve the reports. This is useful to estimate the cost, time and team size required to review the information.

- Reduce the volume of reviewed data: the most traditional techniques applied to reduce the volume of reviewed data are the filtering (by metadata, like date, file signature, extension, folder) and the application of keyword lists. These approaches can be very time-cost effective as SCHULER et.al (2009) explains, “by narrowing the population of documents to review, the overall costs and time are reduced.”
- Identify additional custodians: devices can often have additional user accounts present that indicate other users that utilized the operating system, as well, email and chat messages can indicate additional users to be considered part of the investigation, this goes in the same direction of SCHULER et.al (2009) affirmations that “ECA is useful in not only reducing the overall population of data before the review, but also providing you the ability to identify key and previously unidentified custodians.”
- Identify additional data and devices: the analysis of basic artifacts can enable the identification of previously connected USB storage devices, access to network shared files and presence of backups from mobile devices. All these data sources can contain relevant information that could not be identified otherwise.
- Reduce data privacy issues: the high-level data can indicate the presence of private data that can be subject of data privacy regulations or be considered out of the scope of the investigation.

Great part of the ECA procedures can be automated, thus reducing the number of human-hours analyzing the data assets to search for basic information, but this is done only after the Collection step on the EDRM framework, to be more specific, during the Processing phase.

As SHOOK (2014) explains, civil litigation matters can have dozens or hundreds of custodians and each individual can have multiple electronic devices storing data, but only part of these files are likely to be relevant to the investigation, and “assuming again that these files total even 2 GB in size, preserving the entire 120 GB of a hard drive would result in an enormous over-preservation of data. Multiplied across dozens or hundreds

of custodians (and again by dozens or hundreds of cases), both the time and cost to create these images, along with the extra preservation, processing, and review costs for the additional data, can be enormous”.

The pure fact that the ECA is performed after the collection step imposes the risk to infringe the “first possible moment preservation” principle for the devices and custodians not identified directly on site with the smoking gun.

In other words, because the identification of additional devices, data sources and custodians happen during the Processing phase, these evidences are at stake of being tampered or destroyed during the time between the collection beginning and the processing end.

This risk can be mitigated by applying legal hold methodologies, such as the implementation of audit logs, regular backups or disabling deletion on email and shared servers, but other types of damage and risks still remain valid (e.g.: physical damage, malware, data wiping, devices being stolen, etc.).

The other way around, if data from an incorrect custodian is accidentally preserved, ECA can be applied to identify this issue in an early stage and avoid rework and data privacy issues.

Digital forensics investigations often encounter multiple actors that are potentially involved in the same actions, what can exponentially increase the volume of data to be collected and analyzed. Ritu Aurora (2016) clarifies that when the volume of data is too large to be easily managed in conventional software tools it is called *Big Data*.

The challenges to deal with Big Data can be various, from the time taken to execute a search across millions of documents, database limitations, data normalization (uncommon encodings, different time zones, different metadata fields) and as Ritu Aurora (2016) highlights, “transferring large amounts of data from a storage resource to the computational resource over limited network bandwidth”.

A common procedure is to apply a hash set to identify and potentially remove known bad items (operating system files) or to identify known evidence (known leaked files on a data leak investigation) or already processed data (hash of documents that belong to another investigated individual).

The conjunction of the approaches to narrow down the data to be processed using readily available characteristics is known as data culling, this includes filtering the data by document types, periods, folders, individuals and remove duplicated items.

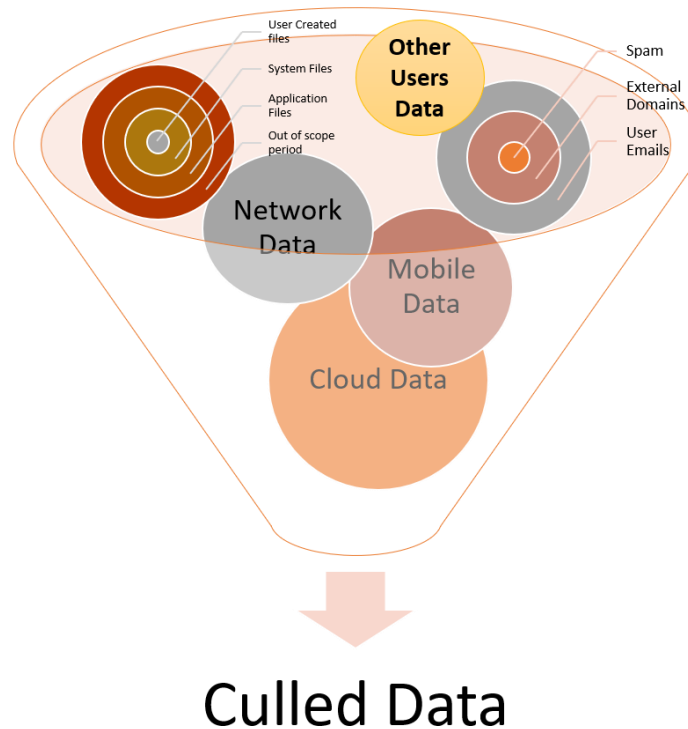


Figure 4 - Data Culling Diagram

In the same direction, Philipp et. al (2009) indicates some approaches that can be taken to analyze mailbox data, by limiting the documents by sender/receivers “If you see known players in the situation, review all of their e-mail boxes” or defining a set of keywords “that relate to the situation and use those to winnow down the e-mail”.

Depending on the collected data, system files and applications it is possible to automatically recover forensic artifacts that can be insightful to the investigation, for example the date of the last shutdown, list of applications present, most recent opened files or the usernames utilizing the device.

As Sondhi S. and Arora R. (2016) highlights, the ECA preliminary processing might be helpful to identify potential obstacles that must be overcome during the next investigation phases, “such as unexpected formats and password-protected files, thereby, enabling proactive measures to address those challenges”.

In addition to these described strategies, it is possible to apply machine learning methods, for example entity extraction, data classification, optical character recognition and language identification to create additional filtering parameters.

“Applications that allow you to identify relevancy early in the case limits your need to review non-responsive documents and could even allow you to capture potentially responsive documents earlier in the process.” (Karen Schuler et.al. 2009)

For the scope of this work, part of the data culling cannot be fully applied, as the execution time of some of the described actions can be higher than the collection itself, but in essence will be used for the parameters that it is possible – for example on filtering of ready metadata for loose files and the filtering for folders – and can come handy to the final report with metrics.

Finally, all this data can be condensed in dashboards for easy data visualization that can report the metrics and highlight the main relevant high-level details that can contribute to future informed decisions.

1.5.3. The main questions that can be answered with ECA

Utilizing ECA techniques to generate dashboards and reports is an easy way to organize the data in a way that a preliminary review can be performed to provide general insights and inform the initial decisions of the investigation.

As LAYKIN (2013) says, “the purpose of this is to assemble data in the form of either accounting data, e-mail, transactional data, or other documents”, this means that not only data from one single individual or a single data source can be used, but also the data from other individuals, transaction databases or even external sources (also known open-source intelligence – OSINT).

With that said, there are many questions that can be answered seamlessly automatically utilizing ECA, from those we can highlight:

- Are there encrypted or compressed files?
- Which kinds of documents are present?
- Which are the users present?
- Are there USB connected devices?
- What are the most recent programs and files executed?
- Does the device have traces of Cloud applications?

- When was the last shutdown?
- Are there databases or other files that require different review approach?
- Are files hosted in an uncommon location?
- Are there anti-forensic indicator signs? (VPNs, encryption tools, wiping tools)
- What is the volume of data?
- Are there backups from mobile devices?
- Are there files on the recycle bin?
- What are the most recent accessed URLs?
- Are there any recoverable items on the *hibernation*, *pagination* and *swap* files?
- Operating system details (hostname, time zone, version, hardware)
- Does this machine have Virtual Machines?
- Are there any Remote Access tools present?
- Are there any tools to create bootable ISO?
- Are there mailboxes present? If so, what are the most recent communications there?

To be able to develop a tool capable to answer these questions it will be necessary to investigate the operating system and applications and understand the place where this information is stored, as well the methods to parse and recover it in a format that can be used to produce a human-readable report.

2. Windows Digital Forensics Artifacts with low processing requirements

Microsoft Windows is present in 77.9% of desktops worldwide as in October 2019 according to StatCounter, being the most frequent source devices for computer forensic analysis, therefore it is the operating system with the most well documented evidence, therefore was selected as the main driver on this investigation.

Forensic artifacts in Windows systems are stored in a multitude of locations and it can vary depending on the version, user configurations and applications installed.

This chapter intends to present the various windows forensic artifacts that can be collected and parsed in a small amount of time and was implemented in the ECA tool used on this thesis.

2.1. Operating System

This section will present some of the operating system artifacts that can be analyzed to spot different aspects of the device usage, including the list of potential users that had access do the host, as well with details of the operating system itself.

2.1.1. Users (Dates and details)

Users are the accounts – local or remote – that accessed the device in a certain period. Among the many vestiges of usage, highlights the Last Login Date, Last Password Change, and the List of Users.

According to Philipp et.al. (2009), “you can determine the first time a user logged into a system by viewing the creation date of the user’s directory.”

Also, SANS clarifies that the OS only stores the most recent usernames, number of logins and account groups while only the last login and last password change times are stored in the registry, no historical information available.

More information can be found on the Table 4 - Forensic Artifacts - User Profiles.

2.1.2. Operating System Details (Last shutdown, version, updates)

The Microsoft Windows Operating System store various usage information that can be important in a triage to classify the sort of data to expect, among them we can

highlight the computer name, last shutdown, the OS Version and the last installed updates.

This information is stored on the “SYSTEM” and “SOFTWARE” registry hives, as well in event logs.

More information can be found on the Table 5 - Forensic Artifacts - Operating System Details.

2.1.3. Hardware Details

From Windows 7 and newer versions, if the user has executed directly or indirectly the energy report with the command “*powercfg -energy*”, a file is created with details of the hardware where the operating system sits. This report brings valuable details, for example the Computer Name, Model, Manufacturer, Bios Version, as well a list of processes running when the report was generated.

The “HARDWARE” Registry Hive is volatile and only available on live systems, as BUNTING (2012) states it “is a dynamic key with no source hive file at the physical level. It is created as a dynamic key in RAM when Windows boots. When the system shuts down, the data in this key is gone”.

The same information is showed in a live environment with the command “*systeminfo*” or by opening the *msinfo32.exe*.

More information can be found on the Table 6 - Forensic Artifacts - Hardware Details.

2.1.4. Time zone

One of the most important data that is stored on an operating system is the time zone that it is configured, if during a forensic investigation this is not considered, all the dates included in the metadata can be incorrectly interpreted, leading to inaccurate conclusions.

This information is stored in the registry hive “SYSTEM” and as CARVEY (2007) states, “can be extremely important for establishing a timeline of activity on the system”.

Detailed information of where the registry is stored can be found on the Table 7 - Forensic Artifacts - Timezone.

2.2. Applications

Depending on the scope of the forensic investigation, might be extremely important to know the current and past installed applications, as depending on what is identified can indicate misuse of corporate assets, anti-forensics practices, as well indicate the existence of electronic information stored in different data sources.

2.2.1. Installed Applications

The registry of current and past installed applications leaves vestiges in multiple parts of the operating system, where it is possible to identify the name of the application, the installed version, the installed date, and last access date, depending on the OS version and available artifact.

It is also possible to identify the most recent executed programs via the Windows Explorer window, this information is stored in the “*SOFTWARE*” registry hive and as HASSAN (2019) explains, it “keeps a record of all executable programs recently launched in addition to the frequency of usage (number of executions) for each recorded program”.

More information can be found on the Table 8- Forensic Artifacts - Installed Applications.

2.2.2. Cloud Drives

Data can be stored out of the investigated device through Cloud Drives and the early identification of those data sources is very important to avoid data deletion or modification. There are many different providers, and their artifacts are stored in several locations, this article will cover some of the most common.

Depending on the installed application, this information may reside in the web history only. The parsing of the data sources can take some time, but the simple indication of the usage can be easily identified on the operating system.

2.2.2.1. Microsoft One Drive (former SkyDrive)

Microsoft OneDrive have some temporary data stored locally and have logs in the “*SOFTWARE*” registry hive.

According to Skulkin and Courcier (2017):

“In the Windows 10 operating system, OneDrive is the default location to save new files, rather than these being saved in My Documents on the local computer, which was previously the default. This means that, unless the user has manually changed their settings, there should be a wealth of forensic information available via OneDrive.”

2.2.2.2. Dropbox

One of the most popular cloud file sharing application, as Skulkin and Courcier (2017) inform “In 2016, Dropbox had 500 million users worldwide, and this number is climbing”, most information is stored in the *SQLite* databases “*filecache.db*” and “*sigstore.db*” at the folder “%USERPROFILE%\AppData\Local\Microsoft\Dropbox”. Adequate tools can be used to parse these databases and extract the list of all files/folders along with their sizes.

2.2.2.3. Google Drive

Cloud storage solution created by Google, offer free storage for users of their cloud applications. Most execution data is stored either in the “*SOFTWARE*” registry hive or the *NTUSER.DAT*, as well with logs on the folder “%USERPROFILE%\Google Drive” or the *SQLite* databases “*sync_config.db*” and “*snapshot.db*” on the folder “%USERPROFILE%\AppData\Local\Google\Drive”.

More information can be found on the Table 9- Forensic Artifacts - Cloud Applications.

2.2.3. Skype logs

Chat logs are incredibly useful to get some clue about evidence identified within the device, users can exchange files and messages discussing about them. Skype comes pre-installed on the most recent versions of Microsoft Windows and there are multiple parsers available.

Depending on the version the location where the skype profile is stored may be different, but by standard is under “%USERPROFILE%\AppData\Roaming\Skype\”, older versions have this information stored in a database named *main.db*, most recent versions can store this data in other locations (*chatsync* log files, pagination and hibernation files for example).

More information can be found on the Table 10- Forensic Artifacts - Skype.

2.2.4. iTunes Backups

If the user of the device has an Apple device that is connected to the computer through iTunes, it leaves some vestiges that can be useful.

When the devices are mounted in the operating system, this information can be stored in other data sources that are explained in the USB Devices List section. If the user performs a backup of the device, it is stored by default in the user profile applications data folder.

More information can be found on the Table 11 - Forensic Artifacts - iTunes Backups.

2.3. Anti-Forensic Indicators

Anti-Forensic is the act of an individual to try to cover its digital tracks against a potential forensic investigation. There are many methods that can be applied to reduce the chance of a digital investigation identify specific artifacts, but even the act of hiding this information leave traces that can indicate this activity.

HASSAN (2019) explains that the anti-forensics techniques goal is “to destroy or conceal digital evidence, thus frustrating forensic investigators and increasing the time needed to perform the initial analysis”.

Additionally, the user can try to obfuscate the data by changing its metadata to indicate other ownership or date, hiding the document within an encrypted container or completely removing the file metadata. This is what Philip et. al. (2009) classifies as an “obscurity method is used by someone to try to obscure the true nature or meaning of some data [...] resulting in a file that will be either misinterpreted or disregarded in subsequent forensic analyses”.

This section will comprehend some of the most common techniques and methods to identify and try to recover some of the data.

2.3.1. Recycle Bin

The most typical anti-forensic process that users perform is to try to delete files by simply sending them to the trash bin in an attempt to get rid of evidence.

Microsoft created this function to create a buffer area where the users have the option to recover files after deletions by mistake. As BUNTING (2013) describes, “when a file is in the Recycle Bin, the user has the option of restoring the file to its original location”.

From a technical perspective, when a file is moved to the *\$Recycle.Bin* folder, its metadata and the contents are divided into two files, respectively the files starting with *\$R* refer to the actual file and its contents, while the files starting with *\$I* refer to the metadata that used to belong to that item (including the original file path, deletion date and file name).

On the most recent versions, the *\$Recycle.Bin* folder is organized in a way that the files are usually in a subfolder that refer to an SID (Security Identifier) that refers to a GUID, this detail is important because it refers to ownership, meaning that each user on a device has their own recycle bin folder.

More details can be identified on the Table 12 - Forensic Artifacts - Recycle Bin.

2.3.2. File System listing and Deleted Files from the Main File Table (MFT)

The *\$MFT* (Master File Table) and *\$MFTMirr* (Master File Table Mirror) are stored at the root of the main partition, not as an actual file but as a reserved area on the partition. It contains the list of all allocated files and folders within the NTFS partition, as well some of the metadata.

Extracting the MFT information enables the easy identification of files that are indexed (sometimes including files that have been deleted), as well the creation of an event timeline.

According to BUNTING (2013), the *\$MFTMirr* “contains a backup copy of the first four entries of the MFT”. Also, there are other files that can be considered on an analysis, for example the *\$Secure* and the *\$LogFile*, but those require different approaches to parse and collect information.

Some details about these artifacts are available at the Table 13 - Forensic Artifacts - Main File Table (MFT).

2.3.3. Virtual Machines

Virtualization is a technology that enables that a computer to emulate one or more computers within it, allowing the user to have multiple flavors of operating systems and encapsulating the usage data within each container.

“Each self-contained ‘virtual machine’ runs like a separate physical computer, and has its own virtualized computing resources, including virtual CPU, virtual hard disks, virtual memory”. (XIAODONG LIN, Introductory Computer Forensics, 2018)

More experienced users can create Virtual Machines within the device, to obfuscate its suspect activities by storing them on those machines instead of the local host. This can be done using a tool like VMware, Virtual Box or using the new Windows Subsystem for Linux (WSL).

CARVEY (2007) highlights that each virtual machine has a virtual memory file, on VMWare these contents are stored in “a file with the ‘.vmem’ extension. The format of this file is very similar to that of a memory dump”.

More details of these artifacts are available at the Table 14 - Forensic Artifacts - Virtual Machines.

2.3.4. Remote Access Tools

Remote Access technologies can be used to enable access to a host machine in order to use its resources without being physically present.

This can be misuse for a few reasons, among them we can highlight the use of the host device by external unauthorized users (including the direct or indirect access provided by a malware) and the use of an additional host hide or exfiltrate information through Remote Access Tools. There are many in the market and the footprints on each.

Some of the most common technologies are Terminal Server, VNC, TeamViewer and Windows Remote Desktop (RDP).

More details of these artifacts are available at the Table 15 - Forensic Artifacts - Remote Access Tools.

2.3.5. Wiping Tools

Advanced users can try to completely remove their data from the filesystem, to do that they use a technique known as wiping that consists in fully replacing the file bytes (or a whole partition/disk) with other information (can be a sequence of “0”, “1” or random combination of both).

After the sectors have been overwritten (wiped), there is no possibility to recover the data that was previously there, but, as Philipp et. al. (2009) explains, you still can “determine whether wiping tools have been installed by reviewing the programs that exist and have existed on the disk”.

Some of the most common tools that have secure deletion capability are *CCleaner*, *Eraser* and *File Shredder*.

More information about how to identify those artifacts are available in the Table 16 - Forensic Artifacts - Wiping Tools.

2.3.6. Encryption Tools

Users can encrypt physical drives, logical partitions, or files to avoid external users to analyze them without knowing the proper key. Even though decrypting is a difficult task, it is possible to identify evidence of the encryption.

Microsoft has an embedded option to encrypt drives called BitLocker, but it is only available for corporate and advanced versions of Windows (Pro, Enterprise, Education and Ultimate editions).

As HASSAN (2019) highlights, “BitLocker uses the AES encryption algorithm with a 128-bit key size by default; however, you can strengthen the encryption by changing the key length to 256 bits for enhanced security”, this along with the requirement of a TPM (Trusted Platform Module) chip makes virtually impossible to decrypt the data without the key.

Similarly, CANDEIAS (2015) recalls that “Microsoft has developed *BitLocker* to enable encryption both on fixed hard drives and removable unities, such as the external drives and USB pen drives, in this case named as ‘*BitLocker To Go*’”.

Some freely available opensource alternatives that offer similar encryption level and features widely used are:

- TrueCrypt: According to HASSAN (2019), this was the “most popular open-source encryption program (used for file and disk encryption) [...]. TrueCrypt development ended suddenly in 2014”.
- VeraCrypt: is a fork of TrueCrypt that is being maintained with backwards compatibility.

Additional details on how to identify the artifacts of those tools are available in the Table 17 - Forensic Artifacts - Encryption Tools.

2.3.7. Others: Steganography, TOR/VPN, metadata manipulation, bootable ISO

Advanced users can use many other complex techniques to hide data, and in this section will be provided some details of the techniques and how to identify them.

Use Steganography tools like *QuickStego*, *Stegosuite* or *OpenStego* to insert text or other files within other files. Philipp et. al. (2009) defines Steganography as “the ability to hide data inside another file”. As these are usually portable tools, simply search for their respective names on the filesystem, or look for them in the **Prefetch**.

Use a metadata modification tool like *Timestomp* to manipulate file attributes, for example to change the author or last user that executed a file or assigning different timestamps of any file within NTFS file system to obfuscate the ownership and lead the investigator to ignore the file.

Use a Virtual Private Network (VPN) or *TOR* (The Onion Ring) to mask and encrypt network traffic. This can be used to exfiltrate data without being noticed by a network data loss prevention tool, below is a list of some popular VPN tools are *NordVPN*, *ProtonVPN*, *KeepSolid Unlimited VPN*, *ExpressVPN*, *OpenVPN* and *TOR*.

A user can utilize a bootable operating system to execute the investigated activities, therefore not leaving traces on the host device. Search for files with the extension *.img* and *.iso* on the filesystem, or the existence of any USB bootable creator. Below is a list of some of the most common are *Rufus*, *UNetbootin* and *Universal USB Installer*. As they are usually portable tools, the search of the executable file name or their respective **Prefetch** artifact should indicate their usage.

Table 18 - Forensic Artifacts - Other Anti-forensic Indicators (Steganography, TOR, VPN, Metadata Changing, Bootable ISO and system configurations)

2.4. Recent Open Files and Applications

There are multiple locations within the operating system that store the recently used files and applications. This information can be essential to reduce the review time on a triage perspective, as well to guide the investigation on certain directions and identify misuse of the device.

2.4.1. Recent Files

Since Windows XP, the operating system keeps record of the recently opened files and applications to facilitate the user to find a file that is used frequently or in the last sessions.

One of the methods that the operating system stores this information is through the Recent folder, which BUNTING (2012) explains that “the purpose of the Recent folder is to provide a user interface that lists documents the user has recently created or modified”.

This folder contains link files (shortcuts, that will be explained on topic 2.4.3 **Link Files**) to the actual files, and the datetime of the link file determines what was most recently opened.

Additionally, the operating system keeps track of the recent files and applications in a few Registries, that are more detailed on the Table 19 - Forensic Artifacts - Recently Used Files.

2.4.2. Jump Files

Windows 7 introduced this new feature called Jump Lists that preserves some information about recently accessed files and applications, they usually are pinned to the task bar and are visible when right-click in any software.

According to Hassan (2019), this feature “allows users to view recently viewed or accessed files for each installed application”.

More details about how to identify these files and parse are on the Table 20 - Forensic Artifacts - Jump Lists.

2.4.3. Link Files

Link Files (items with the “LNK” file extension) are logical shortcuts to other files or applications, they can be placed anywhere in the operating system and do not have a size stored, but preserve metadata of the file name, path, last access and creation date. A link file creation date can be indicative of when a tool was first installed on the operating system along with other evidence.

HASSAN (2019) explains that the link files can “be created by a user or autogenerated by Windows when a user opens a local or remote file”. This is relevant because it can indicate files that are stored outside the device (for instance, in an external USB device or in a network share).

In the same direction, MCQUAINE (2014) highlights that “LNK files are excellent artifacts for forensic investigators who are trying to find files that may no longer exist on the system they’re examining”. The author expresses, in this case, that a LNK pointer file can remain in the device even after the original file was deleted, providing traces of the existence of the subject file in the system.

A list of some folders that commonly contains link files is available at the Table 21 - Forensic Artifacts - Link Files (LNK).

2.4.4. Recent Office Documents

Similarly, to the Recent Files and the Link Files, when the user opens any file using an application from the Microsoft Office Suite in its most recent versions, it automatically creates a registry with the specific file to enable the user to select recent files on the file menu within the application.

As CANDEIAS (2015) highlights, “the system stores the registry keys referring to these items separated for each individual and for file extension, therefore belonging to the registry hive that corresponds to the user profile”.

This list is often referred as MRU (Most Recently Utilized) and more details can be found on the Table 22 - Forensic Artifacts - Recent Office Documents.

2.4.5. Notepad++ Sessions

Notepad++ is a freeware powerful text editor that is widely used, therefore present in many devices. Among the features it has the capability to open multiple text files at the same time and keep them separated in tabs.

Due to this characteristic, it caches some information, including copy of the content of the opened files, keyboard position and a list of recent files, therefore it is possible to identify some documents that were opened in a recent session in the tool.

More details are available at the Table 23 - Forensic Artifacts - Notepad++ Sessions.

2.4.6. Downloads

Recently downloaded files are files that have been saved from internet to a local device, can indicate a tool or file that the user has stored locally for future access.

This information can be stored either in a folder (with the saved files itself, by default is the “Downloads” folder under the User Profile) or on the web browser history database.

For Microsoft Internet Explorer / Edge, this data is stored in the windows Registry (detailed at the topic 2.5 **Internet**).

More information on how to find the browser history databases are at the Table 24- Forensic Artifacts - Downloads.

2.4.7. Print Spooler

When the user selects a file to be printed, in most cases the Windows Operating System will generate a temporary image of each page of the file in the print spooler folder, and delete it after some point, but a targeted forensic recovery can be applied to restore those files.

Philip et. al. (2009) explains that “print spooling is accomplished by creating temporary files that contain data to be printed and sufficient information to complete the print job” while BUNTING (2012) adds that “When the print job completes, the two spool files are deleted. The spool files have extensions of ‘.shd’ and ‘.spl’. The former is called the shadow file, and the latter is called the spool file”.

More details about this evidence are at the Table 25 - Forensic Artifacts - Print Spooler.

2.4.8. Prefetch

Prefetch files were introduced on Windows XP as a feature to load parts of commonly used applications during the windows boot process, it saves the files with the

extension .pf and includes some details like last time of execution, number of times executed.

By storing these commonly used parts (usually DLLs and configurations), as SHASHIDHAR (2015) describes, a “prefetch make it easier for software applications and programs to find what they need on the hard disk. Without them, every program would have to wait on the performance of the hard drive to find any piece of data it needs at time of startup”.

One important characteristic is that the creation date of a Prefetch file is the date when the application was closed, meaning that if an application remains open for days, its first prefetch file will be created in a different date of the application installation. Also, on Windows 8 and 10 this feature is disabled automatically if the operating system is installed in a solid-state drive.

This information is stored by default in “%SYSTEMROOT%\Prefetch” and, as HASSAN (2019) explains, the files follow a specific naming criterion “the name of the running application comes first, then comes an eight-character hash of the location where the application was run, and finally it ends with the .PF extension”.

More details are available on the Table 26 - Forensic Artifacts - Prefetch.

2.5. Internet

When the user opens the web browser to access any URL at the Internet it typically leaves evidence behind including the URL address, the access time and number of access. This topic is valid not only to analyze Internet, Internet Explorer is deeply integrated with the OS since the earlier versions of Windows, due to that, every time a file is accessed using Windows Explorer, some evidence is also stored within the Internet Browser.

2.5.1. Bookmarks

The most direct web browser evidence that could be tied to a specific user are the Bookmarks. Those are shortcuts to URLs that the user store in a place easy to access and would not require to memorize the whole address.

In IE and Edge this data is stored either in the Registry or in a database, while in Mozilla Firefox and Google Chrome this information is stored in an *SQLite* database.

More details of these artifacts are available at the Table 27 - Forensic Artifacts - IE/Edge Bookmarks.

2.5.2. Recent URLs and Searches

Recent URLs are the paths that have been accessed through the investigated device, this includes both the internet access, as well the intranet, local host and in some occasions it preserve access of local files through Windows Explorer – when the access to a file is performed via the navigation bar of Windows Explorer.

Table 28- Forensic Artifacts - IE/Edge Recent URLs and Searches

2.6. Mailboxes

Digital investigations, and especially electronic discovery investigations deeply depend on the analysis of mailboxes, where sits some of the most useful information that could give context to the analyzed data, therefore the identification and extraction of the mailboxes should be a priority over other processes.

Investigate mailboxes is very usual, as HASSAN (2019) mentions, “emails have become the primary means of communications in today’s digital age; for instance, it is rare to see a person who owns a computer, smartphone, or tablet without having an active e-mail account”. Some of main mailboxes formats that can be identified stored offline are Microsoft Outlook and IBM Lotus Notes, due to that will be the focus on this section.

Outlook mailboxes have the “.PST” file extension and are widely used both for private and corporate environments, as the tool is embedded in the Microsoft Office suite and is the standard for export from cloud providers.

Lotus Notes, in the other hand, uses the “.NSF” file extension and is more common to be found in corporate environments, as it requires an IBM Domino server and provide more security controls to the organization. One of the main security measures is that almost always present is the file encryption, as Philipp et. al. (2009) highlights, it “supports real encryption—the Lotus server and client use public key encryption algorithms that cannot be easily broken.”

More details of how to identify the presence of mailboxes are at the Table 29 - Forensic Artifacts – Mailboxes.

2.7. USB Devices List

The usage of external storage devices, especially through the USB port allows the user to keep relevant data out of the investigation umbrella. Luckily, the OS keeps track of all the connected devices, what can be used to help the investigator to identify them on site.

A historical list of connected USB devices, including USB Storage and Media Drives along with metadata of the first connection, last access, make, model and eventually volume letter, serial number or label. As Philipp et. al (2009) highlights, “Windows system registry stores the key USBSTOR, which contains information about the USB devices that have been plugged into the computer”.

These registries contain not only the list of storage devices, but also other plug and play connections (for example a digital camera or a smartphone) and the analysis can be conducted looking also the **Link Files** to identify opened items from the external storage.

Bear in mind that, according to HASSAN (2019), “not all USB device types will leave traces in Windows registry as we have described, for instance, USB devices that use media transfer protocol (MTP) when connecting with computers”.

Depending on the OS version this data is stored in different registry keys and logs, more details are available at the Table 30 - Forensic Artifacts - USB Devices List.

2.8. Memory

Pagination, Hibernation and Swap files are part of a technique named virtual memory utilized by Windows to emulate memory in the hard drive, they are usually stored at the root of the main partition (pagefile.sys, hiberfil.sys and swapfile.sys), contains the recent data that was in memory.

This technology is a common workaround utilized by Windows – and other operating systems – to overcome the limited supply of RAM, that is a very expensive type of memory, and as BUNTING (2012) describes, “When they run out of RAM, they write some of the data that is in RAM to a file whose dedicated purpose is to cache RAM memory”.

With the same fashion of the RAM, these files store the data without a structure, having the blocks allocated in the order that they are required by the applications, Philip et. al. (2009) describes that “You can think of virtual memory as a block of specialized, unallocated space that has no structure”.

The analysis of this data can bring very valuable information to the investigation, as HASSAN (2019) exemplifies, one can encounter “fragments of decrypted files can still reside there, and encryption keys or passwords (or a fragment of it) can also be found here”.

Similarly, there are also the memory dumps, files created by the operating system during one crash that can also be useful in some analysis.

Each of these types of virtual memory will be described with more details in their respective sections in sequence, but the details of where to find these artifacts are available at the Table 31 - Forensic Artifacts - Memory Dumps.

2.8.1. Pagination

The pagination file is usually stored in the “pagefile.sys”. This file is used as a supplemental memory to the RAM, meaning that every time that the operating system identifies that the amount of available RAM is not sufficient to hold the volume of data necessary to execute the needed operations, it will transfer some of the data from idle processes to this temporary file in the hard disk.

The advantage of utilizing the pagination file is that the hard disk memory is cheaper, therefore is common to have larger storage available, the disadvantage is that the memory from the hard drive is much slower than the RAM.

To the forensic point of view, Philip et. al. (2009) highlights that this is stored as a “single file is a free-form block of data much like the unallocated space, except that it holds data that was written to it as a form of secondary memory called virtual memory”.

This means that the data stored on the pagination might not be a complete snapshot of what was available on the live system, but rather only the exceeding data, as BUNTING (2012) says, “the operating system writes volatile data that is not currently in use to the swap file (pagefile.sys) to temporarily hold it there while it makes room in volatile RAM for a current task”.

2.8.2. Hibernation

Hibernation is a functionality that enables the user to shut down the computer and preserve all open processes and memory when turn it back on. CARVEY (2007) points that this “functionality is most often found on laptop systems”. This happens because it is more common for portable laptops to have the power supply removed suddenly for transport.

When the user selects to put the device in hibernation or in sleep mode, most information available on RAM will be transferred to the “hiberfil.sys” file, and when the computer is started again this will be restored to the RAM, allowing a much faster return to the activity.

This process is better described by BUNTING (2012) as follows:

“With Windows XP/Vista/7, if the system is placed into hibernation or hybrid sleep, the contents of RAM are written to the hiberfil.sys file. As the result of this process, link files, not to mention other data, can be found in this file.”

From the forensic point of view, the value of the information is similar to the pagination file, containing almost all the data that was available when the device was active the last time, also, as CARVEY (2007) highlights, “The hibernation file is compressed and, in most cases, will not contain the current contents of memory.”

This means that the hibernation file may have data from a different point in time than when the device was last shutdown.

2.8.3. Swapfile

Very similar to the pagefile, the swapfile.sys preserves data from the most recently opened applications.

The main difference, as HASSAN (2019) emphasizes, it is used to store only the “idle and other nonactive objects ejected from the RAM memory, whenever a user tries to access an idle process again, its information will get shifted to the RAM memory again”.

Despite the fact that it is usually much smaller than the other described virtual memory files – the default size is 256 MB—, it can hold useful information, in particular in regard to applications that were open on the Windows 10.

2.8.4. Memory dumps

Different from the other memory files described above, the memory dumps are created only when occur failures in the Windows operating system. These error files are normally named “MEMORY.DMP” or “minidump.dmp”.

The main difference between them is that the first will try to preserve a copy of the whole memory available at the moment of the crash, while the later, as HASSAN (2019) explains “will usually contain the programs that were running/installed at the time of the crash”.

From the computer forensics point of view this information can be very useful when the analysis requires the recovery of passwords, files that were opened on from encrypted containers, details from open connections or name of tools (in particular if the application that crashed and caused the ‘Blue Screen of Death’ was an encryption software, a steganography tool or a malware).

Philip et. al. (2009) points also that “If your case has at issue events that would have existed only in memory while the user experienced a crash, finding these files could provide valuable evidence”

3. Implementation of tool to perform ECA during forensic collection

As described in the previous chapters, the implementation of an Early Case Assessment tool should bring insights in advance, but for this work, it requires that the implementation process only data that require fast processing. This means that in some cases the information parsed will be in high-level, rather about the presence of a tool, than about its utilization or email review.

SHOOK (2014) explains that “Purpose-built eDiscovery solutions can *crawl* repositories such as email servers and archives, fileshares, SharePoint repositories, desktops/laptops so that they may be searched using keywords, dates of creation, modification or last access dates, file name or type, location, etc.”.

This is when an ECA tool shines, by utilizing automation technologies to leverage the most valuable information and provide it in an easily and quickly readable way, in contrast of standard forensic imaging procedures that will perform a full preservation and will require a separate processing phase to extract this valuable information.

For this reason, this chapter will describe the development of a tool capable of uniting both worlds, therefore it will parse the abovementioned artifacts utilizing the spare processing power of modern devices -- that support multiple cores and large amounts of RAM than the present on hardware-based collection devices – and at the same time perform a full disk forensic preservation.

3.1. Definition of metrics and technologies

For this implementation and subsequent testing steps it is utilized a laptop Asus, with the below configuration:

- CPU: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz (4 physical cores, 8 logical cores).
- RAM: 32 GB DDR4 @ 2.40 GHz.
- SSD 1 (Used for Cache 1): Samsung SSD 970 EVO Plus 500 GB (NVMe PCIe).
- SSD 2 (Used as Destination): Samsung SSD 850 PRO 1TB (SATA III).

Implementation of tool to perform ECA during forensic collection

- The evidence is stored in an external SSD (in an USB 3.0 enclosure):
SanDisk SSD PLUS 120GB (SATA III).

The tool is developed utilizing the programming language Python 3 on the Operating System Ubuntu Linux 20.10 with x64 architecture.

For the intent of this project, the collection is performed from an SSD SATA III that is inserted in a USB 3.0 enclosure, similar to what would be found in a real-world scenario. The data on this drive does not have encryption and was generated by utilization of minimal resources to fulfill all the artifacts that will be collected.

The tests will be performed in two ways, the first called “Scenario 1”, is a perfect scenario where the destination data will go to an SSD SATA III drive that is connected through a SATA port on the motherboard, additionally, the “Scenario 2” is a more realistic situation, where the destination data will go to a SATA Hard Drive connected through a USB 3.0 port.

The tests will verify if the collection time on either workflow is affected by the Early Case Assessment processing thread and quantify the execution time in both cases, therefore calculating the execution times and calculating the volume of collected data to generate an average speed.

The diagram below shows more the collection workflows, being the standard collection (without ECA) represented in yellow and the hypothesis collection (with ECA) represented in blue.

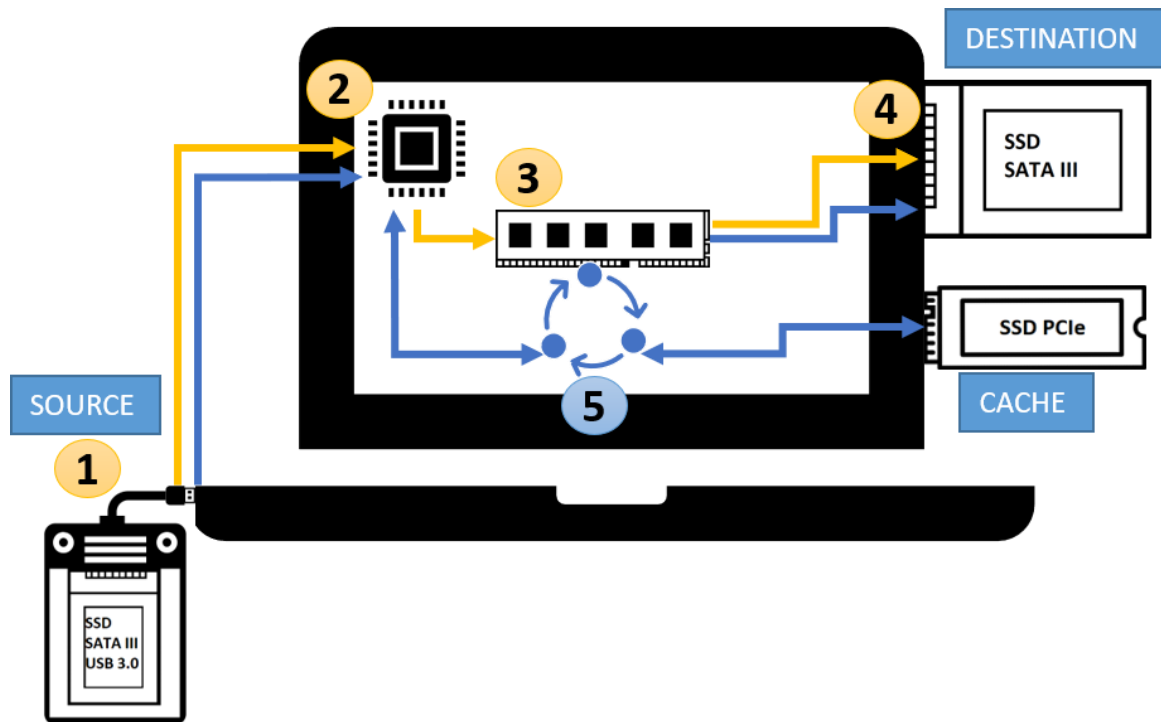


Figure 5 - Collection Workflow

In a standard forensic collection, the data flows from the Source (1), to the CPU (2) to RAM (3) to Destination (4), in a single direction as indicated by the yellow color arrows (with, of course, the exception of some forensic duplicators like the OpenText Tableau or the Forensic Falcon that have the capacity of creating a secondary destination copy in parallel utilizing a controller on the destination port to replicate the data).

The hypothesis of this thesis indicates, as visualized on Figure 5 - Collection Workflow, a slightly different approach where the data at simultaneously follows the same workflow as described above (a full copy from the source to destination) and a secondary thread highlighted by the color blue that will copy some of the items to a temporary cache drive (5), process it and parse it – this step involves the use of data in RAM and CPU processing in multiple interactions, therefore is represented with a circle of arrows in the figure –, and in the end generate a final report that is saved on the destination drive.

3.2. When to collect the ECA data

The full disk collection of an external hard drive can take many hours depending on the type of disk and how much data is in fact present, as well of the speed throughput on the output drive.

ROUSSEV (2013) argues that a forensic triage “is fundamentally a sequential model in which each stage waits for the previous one to complete before it commences. Thus, the only means to improve end-to-end latency is to speed up all stages of the process”.

The author meant that the reading speed of the source and the writing speed of the destination are the sole factors that affect how long a forensic imaging will take, which is partially true, however the objective of this research is to investigate the possibility to execute more tasks in parallel and overcome this performance bottleneck.

As Nikkel (2016) highlights, “A performance bottleneck always occurs; this is simply the slowest component in the system, which all other components must wait for. In a forensic setting, the bottleneck should ideally be the subject disk. This is the evidence source and is the only performance variable that you can’t (or shouldn’t) modify”.

With that said, there is also the need to elect the right moment to start the collection of the artifacts that will be used to perform the Early Case Assessment. This decision must take in count the following factors:

- The size of the artifact to be analyzed.
- The importance of the information that will be recovered with this parsing.
- How fast does it take to parse this artifact?
- The speed of the drive that holds the source.
- The amount of available RAM in the collection workstation.

For this analysis, the collection will follow the order based on the size of the artifact (smaller items first) and after all the ECA items are collected, then the actual full disk collection starts along with the parsing of the ECA artifacts.

The ECA data will be collected to a RAM disk (a virtual drive allocated in the RAM that works similarly to a normal partition) and in case the volume of RAM available in

the device is not sufficient to accommodate all the desired data, it can be temporarily stored in the swap partition on the cache drive (an NVMe PCIe disk described in the previous topic), depending on the amount of data selected for ECA.

The choice of a RAM disk as priority is because of the higher speeds of RAM against SSD drives and because it would not require formatting/encryption or affect the read speeds of neither the source nor destination drives.

At the end of the process, after all artifacts are collected and parsed, a report is generated and the ECA data is saved in the destination drive along with logs and hashes to ensure this task is forensic sound.

3.3. Which artifacts to collect and process

As mentioned previously, to ensure the best results faster, the selected approach consists in collecting first the light items and then the larger items to process them as fast as the hardware allows.

The first items to be copied to the cache are the registry hives, the reason for that is because they are usually very small (in general it will be less than 1 GB in total) but bring valuable information, the following topics are available there:

- USB Devices List
- Users List
- Operating system Details
- Time zone
- Installed Applications
- Recently Opened files
- Recent Office Documents
- IE/Edge Bookmarks
- IE/Edge Recent URLs and Searches

Secondly, the Master File Table, that is in general small (less than 1 GB in a regular user computer), but this can include information about the list of files present in the NTFS partition.

Third, collect the Prefetch folder, that is usually very small (only a few megabytes) but can have very important information on recent applications, in special

Implementation of tool to perform ECA during forensic collection

the portable executables like Steganography tools, TOR, Encryption software and tools to convert ISOs into bootable flash drives.

Fourth, is important to collect the top items from the recent folder (parsed link files can point to the actual items) and recycle bin. The size of these files can vary, thus some parameters might be applied to limit only to the user generated items that have up to certain size, these parameters are discussed on topic **Error! Reference source not found..**

In Fifth, collect the Memory Dumps, Pagination, Hibernation and Swap files, that are in general large (a few gigabytes), but can provide artifacts like some open files (that could have been deleted), some webpages, processes and strings that can be used to identify passwords for encrypted items.

Finally, one of the most relevant artifacts to be collected with little processing required are the Mailboxes. Their size may vary by the user, even though it could be a large item with some gigabytes, it contains information that can be very relevant for the investigation and its processing time is relatively low.

3.4. Development

In order to parse the multitude of artifacts in a quick development process, it was chosen the Python 3 programming language, that is open-source and has various solutions already available in the form of libraries.

A Graphical User Interface (GUI) was developed using the PyQt5 library that is distributed in the GNU License by the “Qt Company”, compatible with most Linux distributions and widely utilized to create user interfaces.

“Qt is probably the best library for developing GUI applications. The combination of Python and Qt, 'PyQT', makes it possible to develop applications on any supported platform and run them unchanged on all the supported platforms.” (SUMMERFIELD, 2007)

The tool, now called “Hakime Forensics” was developed in phases, starting with individual modules that can perform specific tasks and parse specialized artifacts, the next step involved a unification of the tools in a logical execution sequency and finally the creation of the GUI that would be responsible to both show the results of each executed step and receive all the required user input.

Implementation of tool to perform ECA during forensic collection

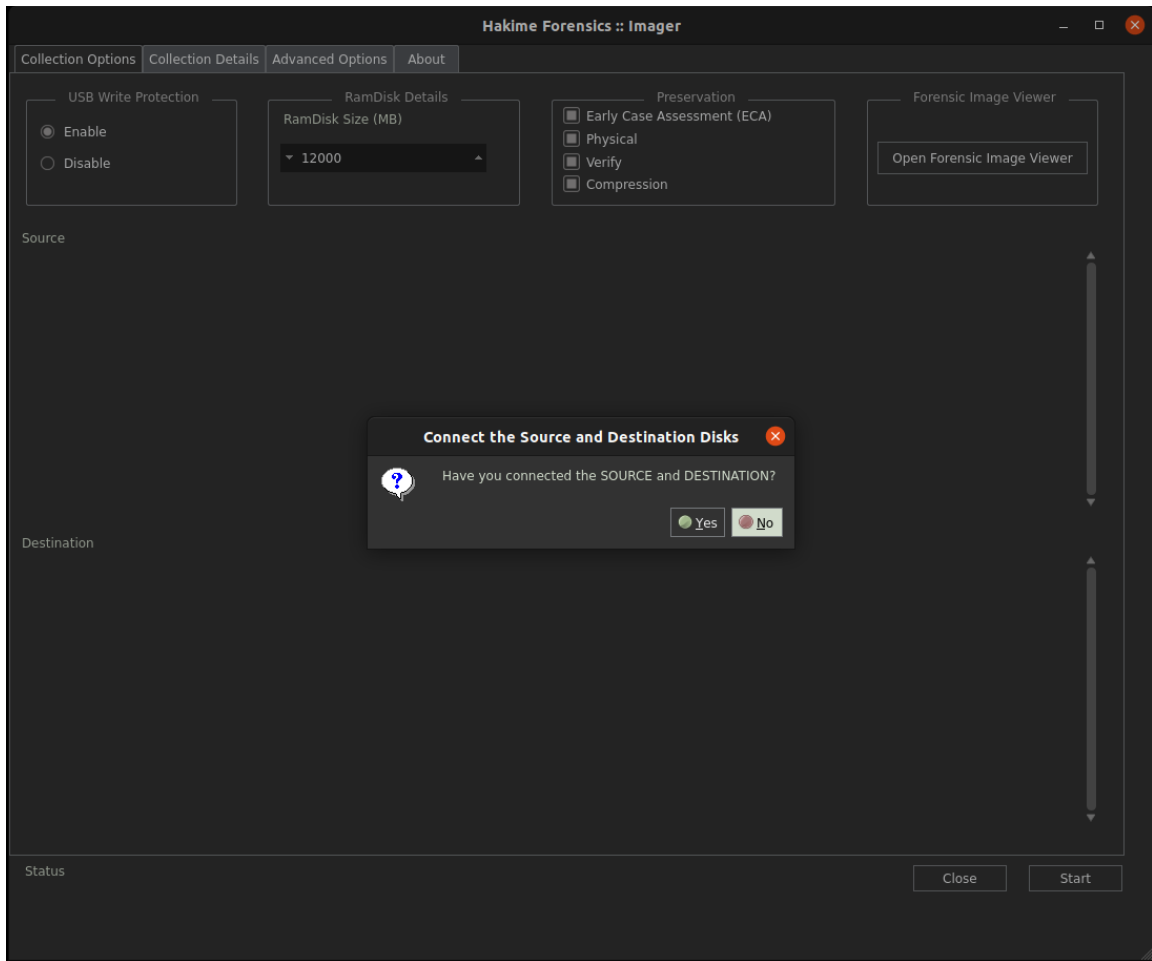


Figure 6 - Hakime Forensics - Main interface – Confirm Mount Drives

The main interface presented in the image above represents the initial status, at this screen the user can select the preservation method, RAM Disk size, and enable the USB Write protection before connecting the source and destination drives.

During the development of the tool, it was observed that depending on the selected artifact to be preserved, the execution time was deeply impacted and thus the total execution time.

An optimal solution to reduce this impact was to first preserve all the ECA selected items to memory, and then start the parsing of such artifacts while the full disk preservation is performed in parallel.

The next screenshot displays the required collection details along with the specialized modules that are described on the subsequent sections, including the basic work of each module that compose the workflow.

Implementation of tool to perform ECA during forensic collection

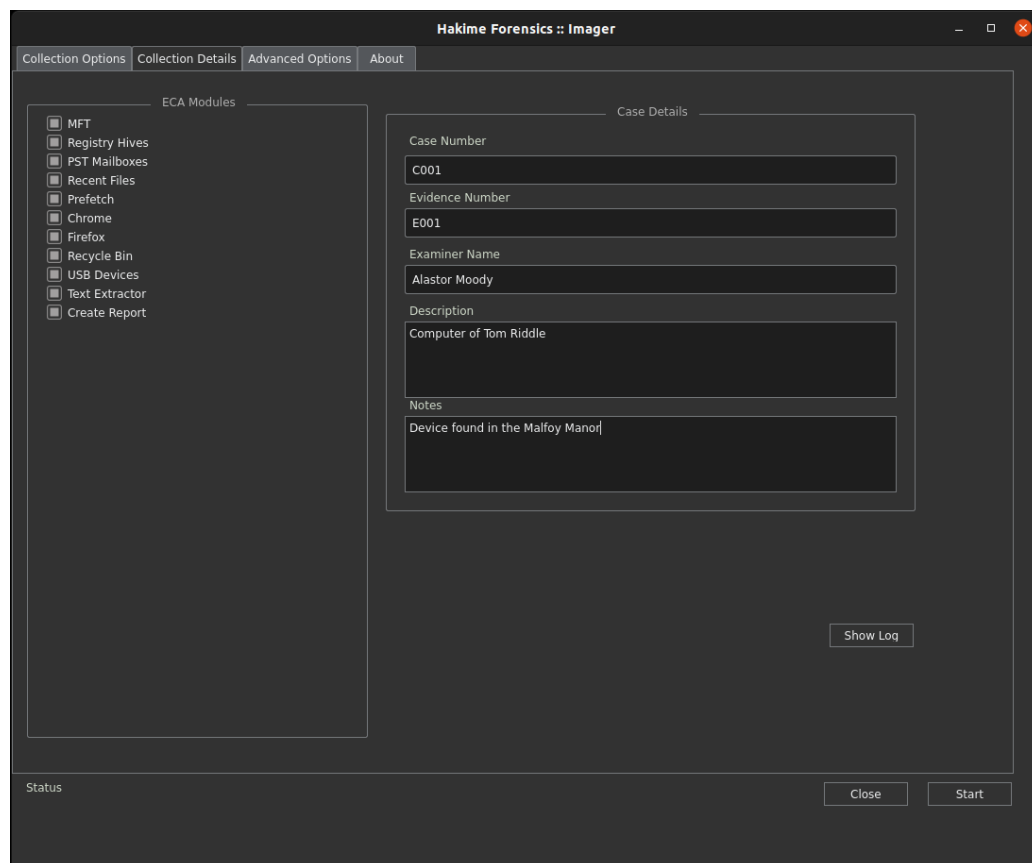


Figure 7 - Hakime Forensics - Collection Details (ECA Modules and Case Details)

3.5. Clearing previous execution

Before starting the forensic preservation, it is essential to perform some checks to ensure the environment is clear and no vestiges of previous executions are remanent, otherwise the tool may not work properly or can have some unexpected results.

This includes temporary folders, mountpoints and if the last application restart was not graceful it can still have some files on the RAM Disk.

3.6. Write Protection and Automount

To prevent that the operating system writes in the source drive, and potentially alter the evidence, the tool takes the following measures:

- Disable and mask the “udisks2” service from ubuntu, therefore disks connected via USB would not be mounted automatically.
- Mount the selected source disk as read only with the following command: “sudo mount -o ro,noload [partition] [mountpointlocation]”

Implementation of tool to perform ECA during forensic collection

Important to highlight that no software write blocking is bullet proof, therefore the utilization a hardware write-blocker is possible at the same time and the same applies to install the tool on a forensic ready distribution, that have additional kernel exceptions to prevent disks to be mounted automatically.

The image below shows how the interface presents the Source and Destination drives after the USB Write Protection is enabled.

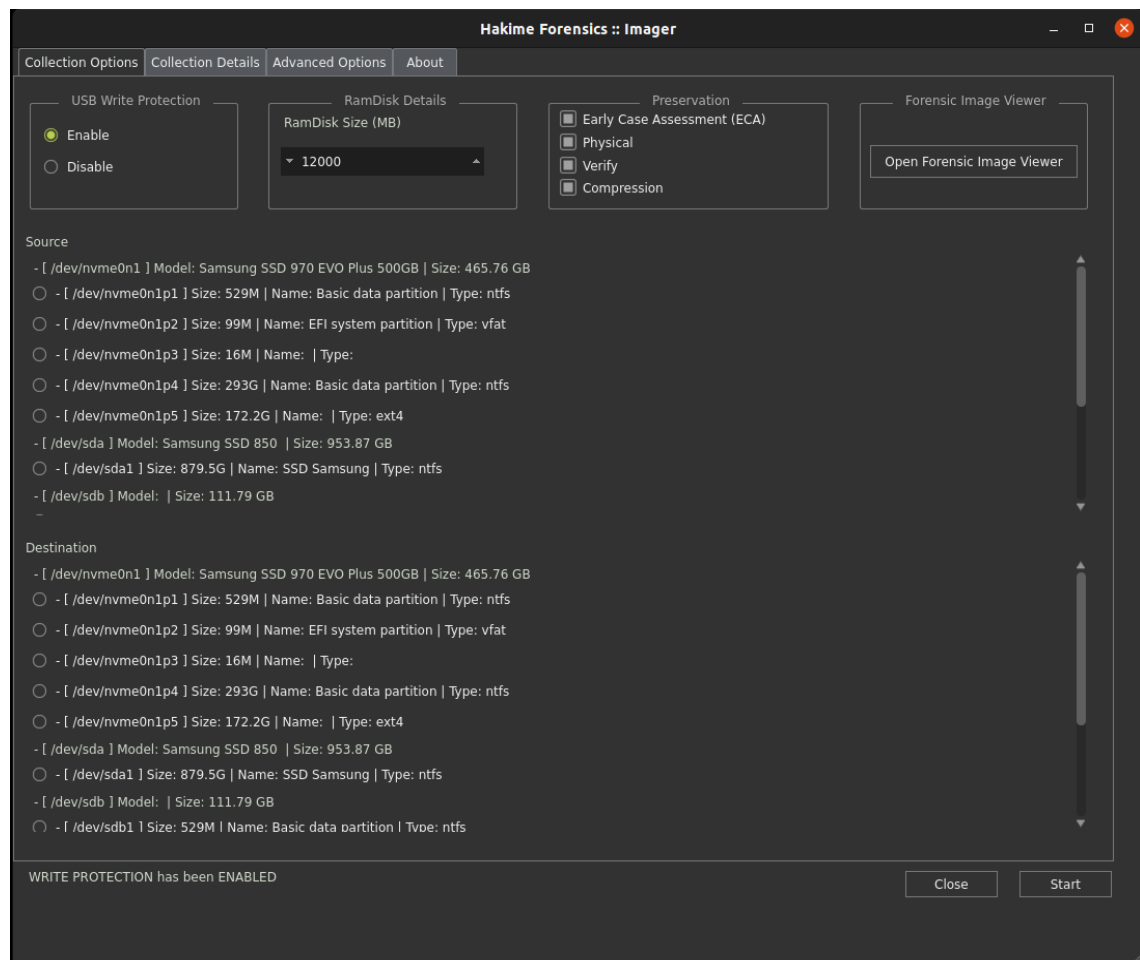


Figure 8 - Hakime Forensics - Mount options and preservation methods.

The Source partition is always mounted as WRITE PROTECTED while the Destination partition is always mounted as WRITE ENABLED, be careful to choose the correct partitions.

A detail to highlight is the status bar at the bottom of the image that shows that the selected source drive was mounted with read only permissions. Once the user selects the partition of the destination disk and confirms that it is the correct drive, this partition will be mounted with normal write permissions.

3.7. Start Early Case Analysis (ECA) Preservation

If all settings are configured and the user input the case details, then the preservation of the selected items for the Early Case Assessment can start, but this can only happen after some necessary steps are executed.

First the variable “start_time” is set, this variable stores the date and time when the start button was clicked, at the same time the variable “start_timer” is set, this variable is a sequential timer that will be used to calculate the elapsed time until the end of the preservation.

Secondly, the RAM Disk must be mounted, this is done by creating a folder on the /tmp Linux directory and mounting a file with *tmpfs*, this file should be with the maximum 90% of the size of the selected RAM Disk space defined on the GUI.

The decision to use *tmpfs* as file system was mostly based on its characteristics of faster and easier management, as explained by SNYDER (1990) “Tmpfs files are written and accessed directly from the memory maintained by the kernel; they are not differentiated from other uses of physical memory. This means *tmpfs* file data can be “swapped” or paged to disk”.

The third step is to create an image container with the NTFS file system that will have 90% of the selected size for RAM Disk, therefore is important to beware of the selected files and limit the Max File Size option to prevent that the preserved items exceed the RAM Disk allocated space.

The fourth step is to general execution log is initialized and stored in the image container, this log will present detailed errors and execution steps from the whole collection.

A fifth step is to initialize the database, this is done using the python sqlite3 library, creating all the necessary tables and columns.

In sequence, the application captures and stores more details of the source disk on the database and run a benchmark that will be used to estimate the execution time and update the progress bar.

The sixth step is to start a worker that will keep the execution logs fresh on the progress bar screen and finally start the ECA worker responsible to preserve all selected data, as shown in the image below.

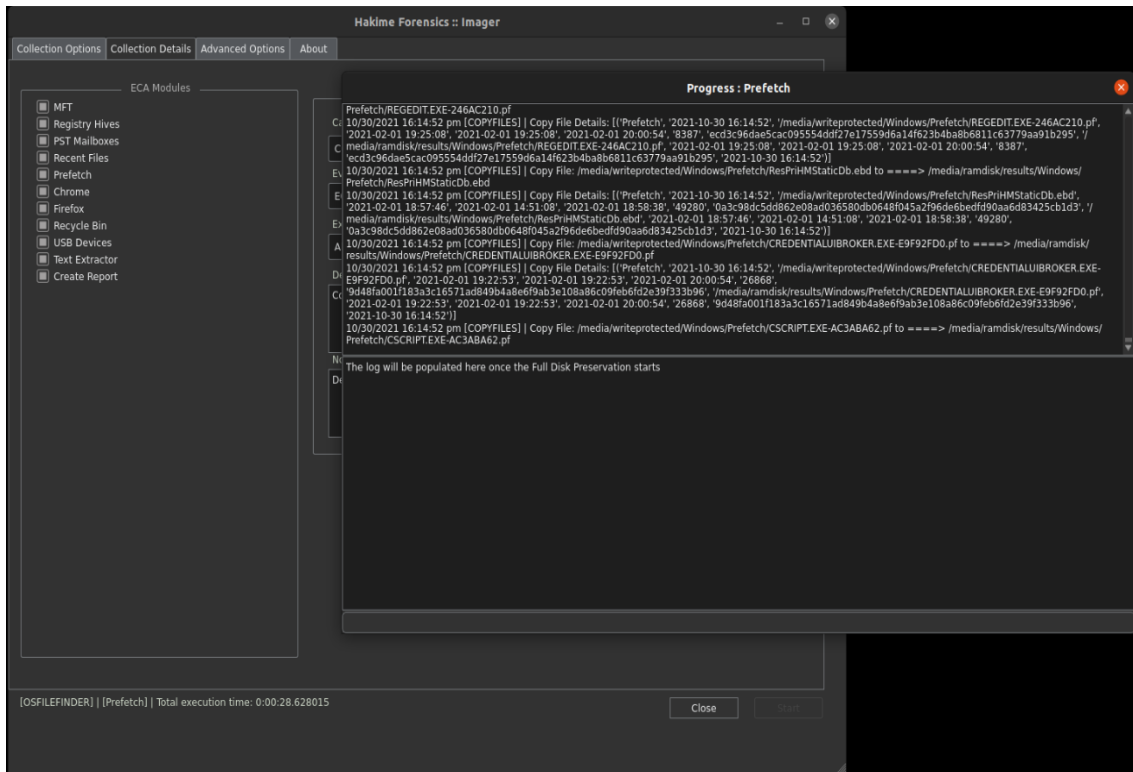


Figure 9 - Hakime Forensics - Execution Log screen.

3.8. Automatic extraction of user generated files

In the main interface there is a tab with “Advanced Configurations”, where one of the options is to import or edit a json file that has a set of selected artifacts with forensic value that can be enabled or disabled.

The application will read this json file and transform it into a dictionary, then run sequentially on each item and check if the item exists in the source drive or not, storing some basic metadata on the database.

The screenshot below shows how these configurations can be changed in the user interface.

Implementation of tool to perform ECA during forensic collection

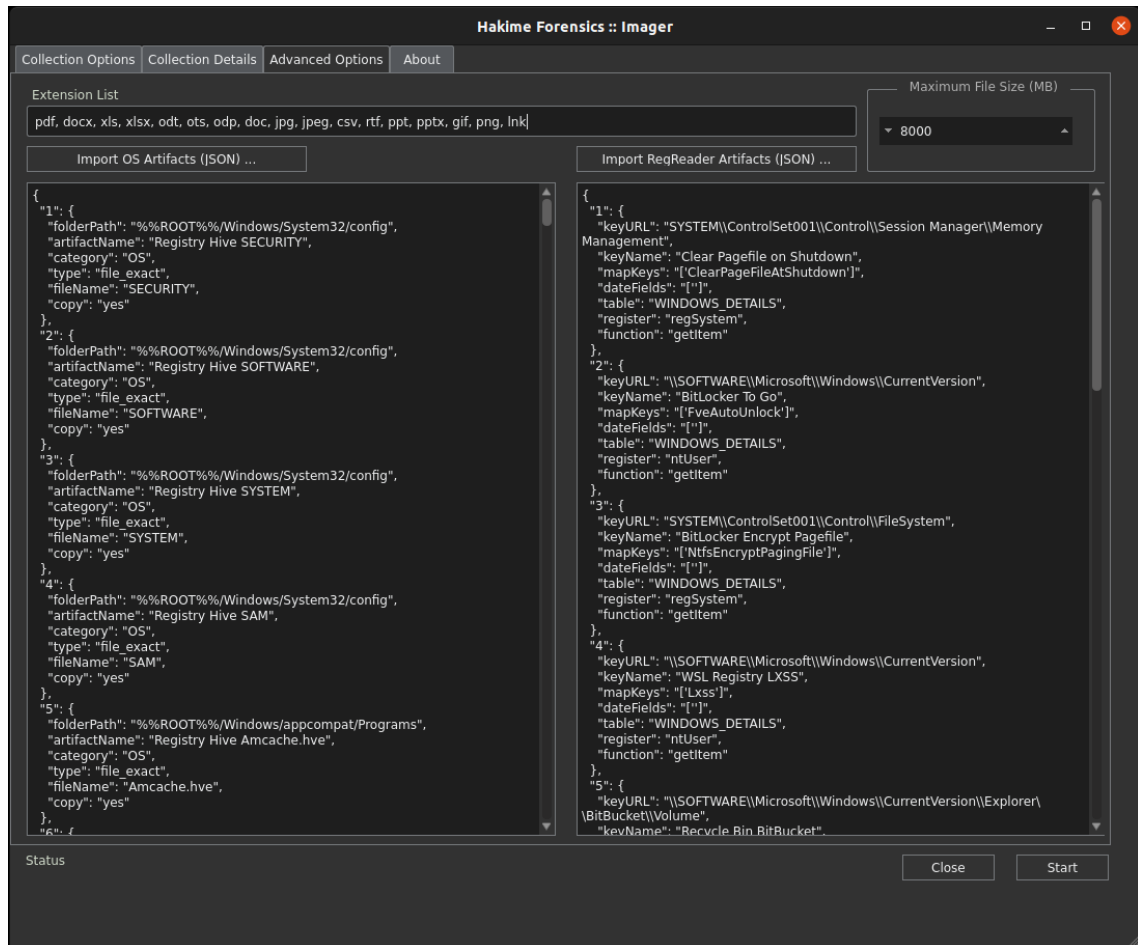


Figure 10 - Hakime Forensics - Advanced Options (Extension list, OS Artifacts, Registry Reader Artifacts and Maximum file size)

If the option to preserve is enabled, the script will call the copy file function, detailed on section **3.10 Forensic logical copy of identified files**, that will create a forensic-sound copy of the file in the container.

Usually, this step is the most time-consuming of the ECA analysis, therefore it is selected to execute first, before the physical collection starts, so lower read impact on the source drive.

As a rule, the tool is prepopulated with over 100 different artifacts, some of them can result in large items and for this reason comes with the preservation option disabled, for example the pagination and hibernation files.

These artifacts are varied between the ones preserved by 'OS File Finder' module (preserve specific files, folders or extensions based on a configurable JSON) and the 'RegReader' module (parses specific registry keys on the preserved registry hives and inserts them in the database).

The structure of the *OS Artifacts JSON* is described below:

```

{
  "1": {
    "folderPath": "%ROOT%/$Recycle.Bin",
    "artifactName": "Recycle Bin",
    "category": "User Data",
    "type": "folder",
    "fileName": "recycle_bin",
    "copy": "yes"
  },
  "2%USERFOLDER%": {
    "folderPath": "%ROOT%/%%USERFOLDER%",
    "artifactName": "Registry Hive NTUSER.DAT",
    "category": "OS",
    "type": "file_exact",
    "fileName": "NTUSER.DAT",
    "copy": "yes"
  }
}

```

Table 1 - OS Artifacts JSON

The **first part** is a unique name for the searched artifact, can be a number or string assigned by the investigator.

The **folderPath** is the folder where the contents are present. It is assigned by the relative path with the options to utilize “%ROOT%” to identify items in the root of the partition or “%ROOT%/%%USERFOLDER%” to indicate that the file is present within any user folder.

The **artifactName** is the artifact name, is an alias to refer to this artifact in the OS_FILE_FINDER table in the database.

The **category** option is similar to the artifact name used to group types of preserved data in the OS_FILE_FINDER table.

The **copy** option defines if the artifact will be copied to the forensic image or just indexed in the database. If **Yes**, the application will try to make a copy preserving as much as possible of the metadata of the file in an image file, otherwise it will only list if the file is present and store it in the OS_FILE_FINDER table in the database.

The **fileName** option is the file name, it can be either the actual file name (used as parameter when the type of search is file_exact) or an alias for the structure (when searching for all files in a directory).

The **type** is the search type, it can be configured to perform a preservation with the following options:

- folder: captures any file within the directory regardless of the name.

Implementation of tool to perform ECA during forensic collection

- `file_exact`: searches specifically for the file path (joining the details of the parameters `folderPath` and `fileName`) and preserves the item that have the exact file path.
- `file`: searches for any file with the name specified in `fileName` within the folder three under the provided `folderPath` and preserves all items that have the exact file name.
- `extension`: searches for any documents that has the extension on the Extension List (configured in the GUI) within the folder three under the `folderPath`.

The structure of the RegReader JSON is presented below:

```
{
  "1": {
    "keyURL": "SYSTEM\\ControlSet001\\Control\\Session Manager\\Memory Management",
    "keyName": "Clear Pagefile on Shutdown",
    "mapKeys": "['ClearPageFileAtShutdown']",
    "dateFields": "['']",
    "table": "WINDOWS_DETAILS",
    "register": "regSystem",
    "function": "getItem"
  },
  "2": {
    "keyURL": "SOFTWARE\\WOW6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall",
    "keyName": "Installed Software (x64)",
    "mapKeys": "['DisplayName', 'DisplayVersion', 'Publisher', 'InstallDate', 'InstallLocation', 'Version', 'URLInfoAbout']",
    "dateFields": "['']",
    "table": "INSTALLED_PROGRAMS_X64",
    "register": "regSoftware",
    "function": "getSubkeys"
  }
}
```

Table 2 - RegReader JSON example

The `'keyURL'` option should provide the path where the registry is stored within the registry hive.

The `'keyName'` option configures a readable custom alias that refers to the preserved key to be stored in the database.

The `'mapKeys'` option is used to select specific keys on a registry folder to be stored in the database.

Implementation of tool to perform ECA during forensic collection

The `dateFields` option is used when there is a known date field that is not stored as plain text in the registry hive, in this case the script will consider the listed fields as dates and try to parse the contents. The Windows Registry is very complex in this case and each application can use a different encoding or mode to store a date, therefore the parsing of the date fields contents is not always accurate.

The `table` option will select one of the existing tables on the 'forensic.db' *SQLite* database to store the contents.

The `register` option refers to the registry hive that contains the selected key and it has the following options (case sensitive):

- **regSystem**: 'Registry Hive SYSTEM'
- **regSoftware**: 'Registry Hive SOFTWARE'
- **regSecurity**: 'Registry Hive SECURITY'
- **ntUser**: 'Registry Hive NTUSER.DAT'
- **regSAM**: 'Registry Hive SAM'
- **amCache**: 'Registry Hive Amcache.hve'

The `function` has the following options:

- **getItem**: will capture the exact key URL as presented with the specific keys listed on mapKeys.
- **getSubkeys**: will capture all subkeys with the names listed on mapKeys that are available within the path of the keyURL.
- **getOfficeMRU**: Designed specifically to preserve the OfficeMRU keys that contains the most recent used documents on Microsoft Office applications.

3.9. Extraction of Master File Table (MFT)

As detailed in the section **2.3.2 File System listing and Deleted Files from the Main File Table (MFT)**, the MFT contains a list of the items that have been stored in the storage device, this list may contain active files (visible to the operating system) or inactive items (items that have been already deleted, but are still referenced in this table).

In essence, the module will first use the *"mmls"* tool from the consolidated *"Sleuth Kit"* framework to identify the offsets of partitions on the source disk, secondly use the *"icat"* tool from the aforementioned toolbox and dump the MFT in raw format and afterwards use the *analyzeMFT* python library to convert the raw dump into a CSV that can be finally stored in the database as a table.

Implementation of tool to perform ECA during forensic collection

As described by O'CONNOR (2010), "*AnalyzeMFT* is a stand-alone Python script designed to fully parse the Master File Table (MFT) from an NTFS file system and present the results in human-readable format (Kovar, 2010). *AnalyzeMFT* is constructed entirely in Python and for each MFT record can record if the entry is valid, type of record, parent folder record and sequence, standard information attributes, file name records, object IDs, birth Volume ID, Domain, flags and notes".

To summarize, the tool will dump this list of items and store it into the database, this way it is quickly available for searches for file names that could be potentially investigated further.

This module will additionally create a list with the most recently opened items and store them in the database for quick analysis. If any of the recently opened items have a file extension from the extension list configured on the "advanced configurations" tab, the module will also try to find the file in the drive and create a forensic sound copy.

3.10. Forensic logical copy of identified files

The forensic copy files module is the core of this application, developing a crucial action in the whole process.

Notably, when a preservation is called '*forensic sound*' it should ensure that the maximum of the original metadata and the whole of the content is preserved. This means that there are occasions that some of the metadata can be lost in transit, this happens because the partition table of the source drive is different from the destination drive or because the format holds metadata or permissions that the preservation tool is not capable to validate.

A first step of the preservation of the logical file is to calculate the hash of the source file before the copy starts.

Likewise, to try to preserve as much metadata as possible, the application uses the "shutil copy2" python library, this way the modification, last access, paths size and contents are preserved in full.

In addition, the application will change the OS clock to the exact moment of the creation of the file, copy it to the forensic container and change it back to present (adjusting the elapsed time in between).

As stated in the documentation, “the `shutil` module offers a number of high-level operations on files and collections of files. [...] Even the higher-level file copying functions (`shutil.copy()`, `shutil.copy2()`) cannot copy all file metadata. [...] On Windows, file owners, ACLs and alternate data streams are not copied”.

Because of this, a limitation of the ECA module is that not all metadata is preserved, but the whole content and the maximum of the metadata is preserved on this module, hence, the objective of the tool is not to replace the use of a full disk forensic imaging tool, but rather execute the ECA module in parallel with a full disk forensic imaging.

Finally, the script will hash the destination file and confirm that the values are the same, storing all this information on the database and execution log.

Noteworthy mention that even if the preservation for the ECA loses some of the metadata, the tool is intended to execute along with a full physical preservation (bit by bit) that ensures the total preservation of the metadata.

3.11. Start Full Physical Preservation (if selected)

After all files are identified and preserved in the forensic container on the RAM Disk, the tool will finally start the full disk physical preservation in a parallel thread to the analysis and artifact parsing modules.

This preservation is performed using the widely used tool *ewfacquire* and will utilize the case details provided by the user on the interface, as well generate the default hash log and save the full execution log.

Ewfacquire is a tool part of *libewf* created by Joachim Metz and Robert-Jan Mora in 2006 and is a library that “includes support for reading Expert Witness Format (EWF) image files. This was accomplished using *libewf*, which is an open-source C library that we developed”.

Defensibly, the tool will use the “empty-block” compression method to create a smaller size with compression only of non-populated content blocks, generate a SHA256 hash, create slices of 4 GB in the EWF (E01) format.

The selection of this format is rationalized because is the most used on the community, as explained by METZ (2006) “The Expert Witness Compression format (EWF) is used by EnCase (*Guidance*) and FTK (AccessData) to create bit-copies. EWF

Implementation of tool to perform ECA during forensic collection

currently is the de-facto (widely used) evidence file standard used within the forensic community”.

When the full disk imaging starts, the execution screen will populate the second block of text with the output from `ewfacquire`, as show in the image below.

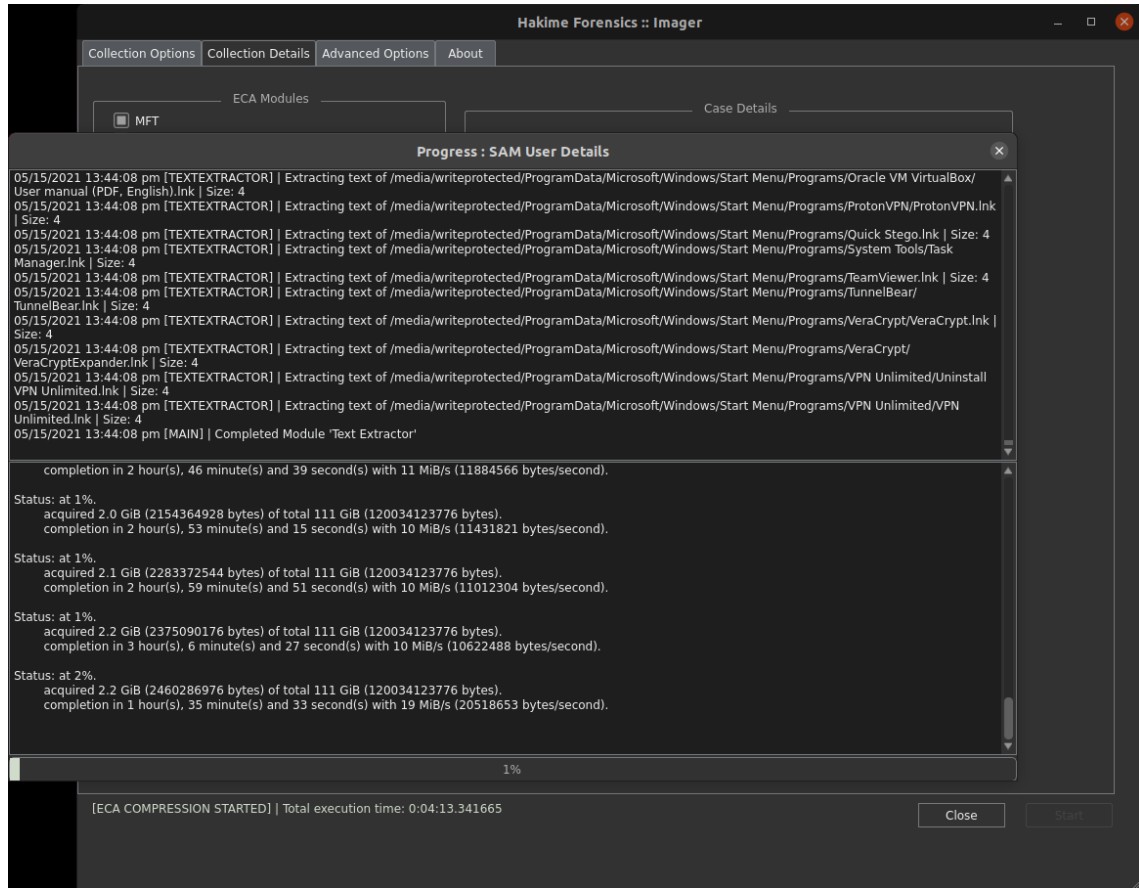


Figure 11 - Hakime Forensics - Execution log screen with full disk image.

If selected, the module will also perform a verification at the end, utilizing the `ewfverify` tool and populate the results on the same text area.

3.12. Registry Reader

One of the digital artifacts with more valuable information centered on MS Windows OS is the Windows Registry, it has multiple types of information that are essential for the functioning of vital parts of the operating system and installed applications.

The Registry Reader module uses the python library `regipy` to read the registry hives and parse this information and store it on the `SQLite` database. It has a set with 26

different core sub modules that cover various aspects of the user usage, these sub-modules cannot be individually disabled.

Additionally, this module reads a customizable json file with a set of 13 other artifacts that can be preserved, meaning that this module can be extend its functionality, allowing the user to modify the list before the execution and therefore parse new forensic artifacts.

3.13. Mailbox (PST) Parser

Users of Windows devices, in particular on corporate environments, can store mailbox information in MS Outlook and as a result have local copies of the exchanged emails on files with the PST (Personal Storage Table) format.

These files can exceed a few gigabytes of size, but the content can be crucial for an investigation nevertheless the processing and indexing time is relatively quick.

The valuable information present on this artifact is better described on the topic

2.6 Mailboxes

In essence, this module is implemented utilizing the python library *libratom* to convert the text and metadata of the email messages into a dictionary and subsequently store it in the *SQLite* database.

3.14. Recent Files Parser

The Windows operating system uses a folder named “recent files” to store shortcuts to the most recently used files. This includes not only files present on the internal storage, but also references to external storage medias and network shares.

Moreover, these shortcuts use a standard format with the “lnk” extension, also called “link files” contain some information that refer to the actual file, that can include file name, size, original location and so on.

As highlighted by MCQUAID (2014) on the Magnet Forensics website, “they are shortcut files that link to an application or file commonly found on a user’s desktop, or throughout a system and end with an .LNK extension. LNK files can be created by the user, or automatically by the Windows operating system”.

Implementation of tool to perform ECA during forensic collection

Some additional details of this artifact have been explained on the topic **2.4.1 Recent Files** as well on **2.4.3 Link Files**.

The Recent Files Parser module utilizes the python library *Inkparse3* to read this information and store on the database.

3.15. Prefetch Parser

A notable forensic artifact that can aggregate value in particular on investigations that involve malicious activities or suspicion of utilization of specific software are the Prefetches.

Microsoft creates a file with a string referring to the original executable and utilizing the “pf” extension and stores it on the folder named “Prefetch” on the Windows folder. This file not only contains the original name of the executed tool, but also the number of times it was executed and the dates of the most recent executions.

As SHASHIDAR (2015) explains, “the entire purpose of a prefetch file's existence is to decrease the startup time of a program. That means that each individual file will hold data directly related to its respective program in a format similar to a list of instructions”.

Some additional details of this artifact have been explained on the topic **2.4.8 Prefetch**.

To parse this information the first step the module uses is to execute the third-party tool named *sccainfo* (part of the *libscca* package) to recover the information present inside the file, and then it parses the recovered information, converts it to a dictionary and stores in the *SQLite* database.

3.16. Google Chrome Parser

Google Chrome is an internet browser that can be used to navigate on internet, therefore it can store valuable information about the recently visited websites, search history and downloaded items.

This information is usually stored in the “history” file within the installed folder or the default user profile depending on how the browser was configured in the device.

In any case, this file is an *SQLite* database that can be directly queried to recover the desired information.

In brief, the module Google Chrome Parser will query the tables “visits” and “downloads” from the source database and save it on the index unified database.

3.17. Mozilla Firefox Parser

Similar to the previous section, Mozilla Firefox is an internet browser that is widely used and also stores the recent data on an *SQLite* database on the user profile installation folder.

Likewise, the module Mozilla Firefox Parser will query the tables “*moz_places*”, “visits” and “downloads” from the source database and save it on the index unified database.

3.18. Recycle Bin Parser

When a user selects an item and press the button delete, by default the Windows operating system will not *in fact* delete the item, on the contrary, it will move the selected file to a special folder named “Recycle Bin” and will create multiple vestiges of this action.

Ordinarily these vestiges are intended to facilitate in case the user has a second thought and decide to restore those items back to their original location, but in the other hand this can house very valuable information.

The valuable information present on this folder and the intrinsic details from the “\$I” and “\$R” files are described better on the topic **2.3.1 Recycle Bin**.

Wherefore the parsing process for the Recycle bin is divided in three steps:

First, it will be necessary to identify the owner of the file, this is done by parsing the path of the direct subfolder where the recycled item is stored and then looking into the registry file where it contains the usernames and details of the original file owner.

In a second step, the “\$I” file is parsed to identify the original file path where the item was originally stored before the deletion.

Finally, the “\$R” file is parsed to recover the additional metadata, that may include file name, creation, last access and last modification dates.

Implementation of tool to perform ECA during forensic collection

After all details are parsed in a dictionary, the tool will store this information in the index unified database.

3.19. USB Devices Parser

In the moment that a user connects an external storage drive utilizing the USB port of a device with Windows installed some traces are automatically stored in the Windows Registry.

The level of information preserved and the location where to find these artifacts may vary vastly depending on the OS version and the external storage device itself, but fortunately there are libraries available to recover specifically this type of artifact.

The module USB Devices Parser utilizes a widely utilized python library called *usbdeviceforensics* that was refactored to return the results in the format of dictionaries as a module.

The information is subsequentially saved in the index unified database.

3.20. Indexing of small items

Items that have been stored by the OS File Finder or the MFT Parser that fall in the requisites of being small and have one of the selected extensions on the front end can have the text extracted and indexed to a faster search utilizing keywords.

To extract the text, the tool will utilize the open-source library called *Tika*, developed by the Apache Software Foundation, that is compatible with most common file types.

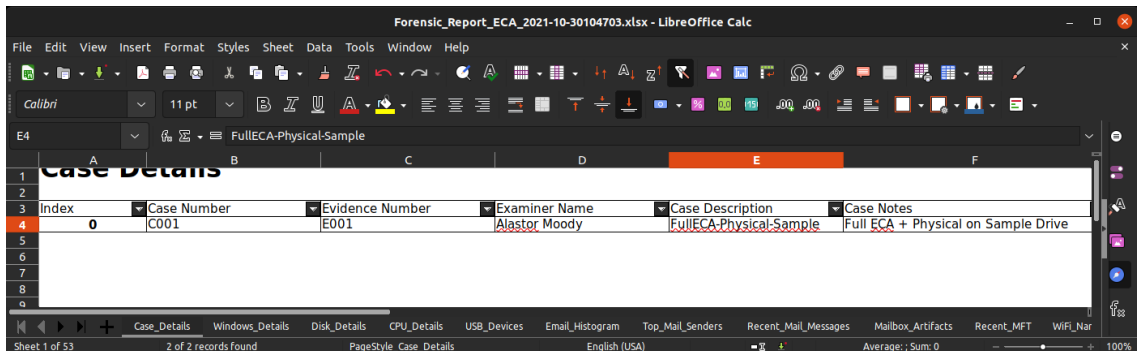
The content and additional metadata extracted from those items is stored in the index unified database.

3.21. Report generation

A report is generated in format Excel utilizing the python libraries *pandas* and *xlswriter*, containing a summary of the contents stored in the index unified database. This may include the most recently sent emails, USB devices, recently accessed URLs, case details and details of the preserved device.

Implementation of tool to perform ECA during forensic collection

This report contains the most important information ready to be used on-site without need of any extra action. Some tabs are limited for the most recent items as the intent of this report is to open quickly and bring the values in a fast easy to read format.



The screenshot shows a spreadsheet application window titled 'Forensic_Report_ECA_2021-10-30104703.xlsx - LibreOffice Calc'. The spreadsheet has a table with the following data:

Index	Case Number	Evidence Number	Examiner Name	Case Description	Case Notes
0	C001	E001	Alastor Moody	FullECA-Physical-Sample	Full ECA + Physical on Sample Drive

Figure 12 - Excel Report (example)

If the investigator intends to analyze information that is more specific or perform searches against the indexed text and metadata of the files, this information is stored at the *SQLite* database 'forensic.db' or the forensic image directly.

3.22. Finalization

After the report is created and all steps from the early case assessment are processed, the tool will perform some steps to ensure that the data is forensically sound preserved, while it still provides the relevant information quickly.

The first step is to close the database and copy it along with the final report and the execution logs to an open folder on the destination drive. Note that this is a simple copy so those items can be opened while the full disk preservation is still in process.

The next step is to close the forensic image file, compress it and copy to the same folder on the destination drive.

Finally, the tool will calculate a hash of the compressed image file and save a log with details of the preservation time.

Implementation of tool to perform ECA during forensic collection

```
Forensic Preservation
Tool Name: Hakime Forensics :: Imager
Tool Version: 0.0.1
EWF Version: ewfacquire 20140807
Tool Author: André Hakime Dutra
*****
CASE DETAILS:

Preservation Mode: Early Case Assessment (ECA):True | Physical:True | Verify:True | Compression:True
Case Number: C001
Evidence Number: E003
Examiner Name: Andre Hakime
Case Description: SSD USB to SSD Sata
Case Notes: Full Physical Image + ECA
Image Format: encase7
Disk Details: {'DEVNAME': '/dev/sdb4', 'ID_MODEL': 'SSD_PLUS', 'ID_REVISION': '04RL', 'ID_SERIAL': 'SanDisk_SSD_PLUS_000ECC550035B158-0:0',
'ID_SERIAL_SHORT': '000ECC550035B158', 'ID_BUS': 'usb', 'ID_PART_TABLE_UUID': '6223c940-9958-464b-a779-4802441762b4', 'ID_PART_TABLE_TYPE':
'gpt'}
Additional Disk Details: {'disk_details': {'/dev/sdb': {'disk': '/dev/sdb', 'disk_size_gb': '111.79', 'disk_size_gb_unit': 'GiB',
'disk_size_bytes': '120034123776', 'disk_size_sectors': '234441648', 'disk_model': 'SSD PLUS', 'disk_type': 'gpt', 'disk_identifier':
'6223c940-9958-464b-a779-4802441762b4'}}, 'partition_details': {'/dev/sdb1': {'bootable': False, 'name': '/dev/sdb1', 'start': 2048, 'end':
1085439, 'blocks': 1083392, 'size': '529M', 'partition_id': 'Windows', 'partition_id_string': 'recovery environment', 'partition_label':
'Basic data partition', 'partition_block_size': '512', 'partition_UUID': '337eed70-a334-4505-8315-2895090f4e8e\n', 'partition_filesystem':
'ntfs', 'partition_PARTUUID': '337eed70-a334-4505-8315-2895090f4e8e\n'}, '/dev/sdb2': {'bootable': False, 'name': '/dev/sdb2', 'start':
1085440, 'end': 1288191, 'blocks': 202752, 'size': '99M', 'partition_id': 'EFI', 'partition_id_string': 'System', 'partition_UUID':
'2a2eefd9-b9a8-4678-80fc-444549415068\n', 'partition_filesystem': 'vfat', 'partition_label': 'EFI system
partition', 'partition_PARTUUID': '2a2eefd9-b9a8-4678-80fc-444549415068\n'}, '/dev/sdb3': {'bootable': False, 'name': '/dev/sdb3', 'start':
1288192, 'end': 1320959, 'blocks': 32768, 'size': '16M', 'partition_id': 'Microsoft', 'partition_id_string': 'reserved', 'partition_UUID':
'eb111c66-2adf-4641-83a6-485fa0d3c45c\n', 'partition_filesystem': 'ntfs', 'partition_block_size': '512', 'partition_UUID': 'eb111c66-2adf-4641-83a6-485fa0d3c45c\n', 'partition_label': '',
'partition_filesystem': '', 'partition_block_size': ''}, '/dev/sdb4': {'bootable': False, 'name': '/dev/sdb4', 'start': 1320960, 'end':
132040703, 'blocks': 130719744, 'size': '62.3G', 'partition_id': 'Microsoft', 'partition_id_string': 'basic data', 'partition_UUID':
'ba3d3dc0-f7d9-4952-bfd5-ea3ffedc00f5\n', 'partition_filesystem': 'ntfs', 'partition_label': 'Basic data partition', 'partition_PARTUUID':
'ba3d3dc0-f7d9-4952-bfd5-ea3ffedc00f5\n', 'partition_block_size': ''}}}
*****
ECA IMAGE:

ECA Image Start: 2021-10-30 18:32:35
ECA Image End: 2021-10-30 18:43:21
ECA Image Hash SHA256: 3bd4fe3146372f2ec84bd28818f7eb3bb565646bc78a855b4127f51055f9203f
ECA Image Hash MD5: 100c425fc89b0239fblc1137d6bc0af4
ECA Image Path: /media/destination/C001-E003_2021-10-30_183235/C001-E003_2021-10-30_183235.zip
*****
FULL DISK PHYSICAL IMAGE:

Physical Image Start: 2021-10-30 18:34:04
Physical Image End: 2021-10-30 18:50:56
Physical Image Verify Start: 2021-10-30 18:51:11
Physical Image Verify End: 2021-10-30 18:51:11
Physical Image Hash SHA256: 5569031e0f97abf31633c3926af78461ea7d407bb2ed8e3a10bc7bfff2b53bf0
Physical Image Hash MD5: dc9ebfcb150155ac41608dca3bec3e33
Physical Image Path: /media/destination/C001-E003_2021-10-30_183235/Physical_Image/C001-E003_2021-10-30_183235.E01
*****
PRESERVATION DURATION:

Start Time: 2021-10-30 18:32:35
End Time: 2021-10-30 18:51:11
Total Execution Time: 0:18:36.434213
```

Figure 13 - Summary Log example.

One important aspect to highlight about the logs of the ECA Image is that the hash is not expected to match between one preservation and another, this is because the hash is calculated based on the zip file (that contains the forensic image, logs, index database and report).

The hashes of individual files present in the results folder are preserved in the 'forensic.db' and should match with each individual file, as well as against its original version on the source device or within the full disk image.

The application generates multiple logs specific for each execution step, the one presented above is the Summary Log, that is created at the end of the execution and will contain the most valuable and clean information, for example the execution start/end for each step, respective hashes and disk details. This log is named with the following standard: CASEID-EVIDENCEID_YYYY_MM_DD_HHMMSS_summary_log.txt

Additionally, the application will generate an execution log that contains full detail on every module and artifact preservation from the ECA execution. This log is

Implementation of tool to perform ECA during forensic collection

named with the following standard: CASEID-EVIDENCEID_ECA_YYYY_MM_DD_HHMMSS.log

If the execution includes a full disk image, the tool will save the execution log with the command line output from *ewfacquire*. This log is named with the following standard: CASEID-EVIDENCEID_YYYY_MM_DD_HHMMSS_collection.txt

Finally, if the collection method includes the verification, the tool will save an execution log of the verification named with the following standard: CASEID-EVIDENCEID_YYYY_MM_DD_HHMMSS_verify_execution.txt

3.23. Image Viewer and Database Analysis

The results can be analyzed in multiple ways depending on the investigator objectives. The easiest way is to use the Hakime Forensics Image Viewer, built-in function that allow the user to mount the forensic image directly and navigate through the contents without performing changes on its contents, as well mount the database and run simple queries.

3.23.1. Open the Hakime Forensics Viewer

Open the application and click on “Open Forensic Image Viewer” on the main screen or via terminal type:

```
hakimeforensics viewer
```

This will open the Hakime Forensics Image Viewer, that enables you to analyze the contents of the image and the database without changing the contents.

Implementation of tool to perform ECA during forensic collection

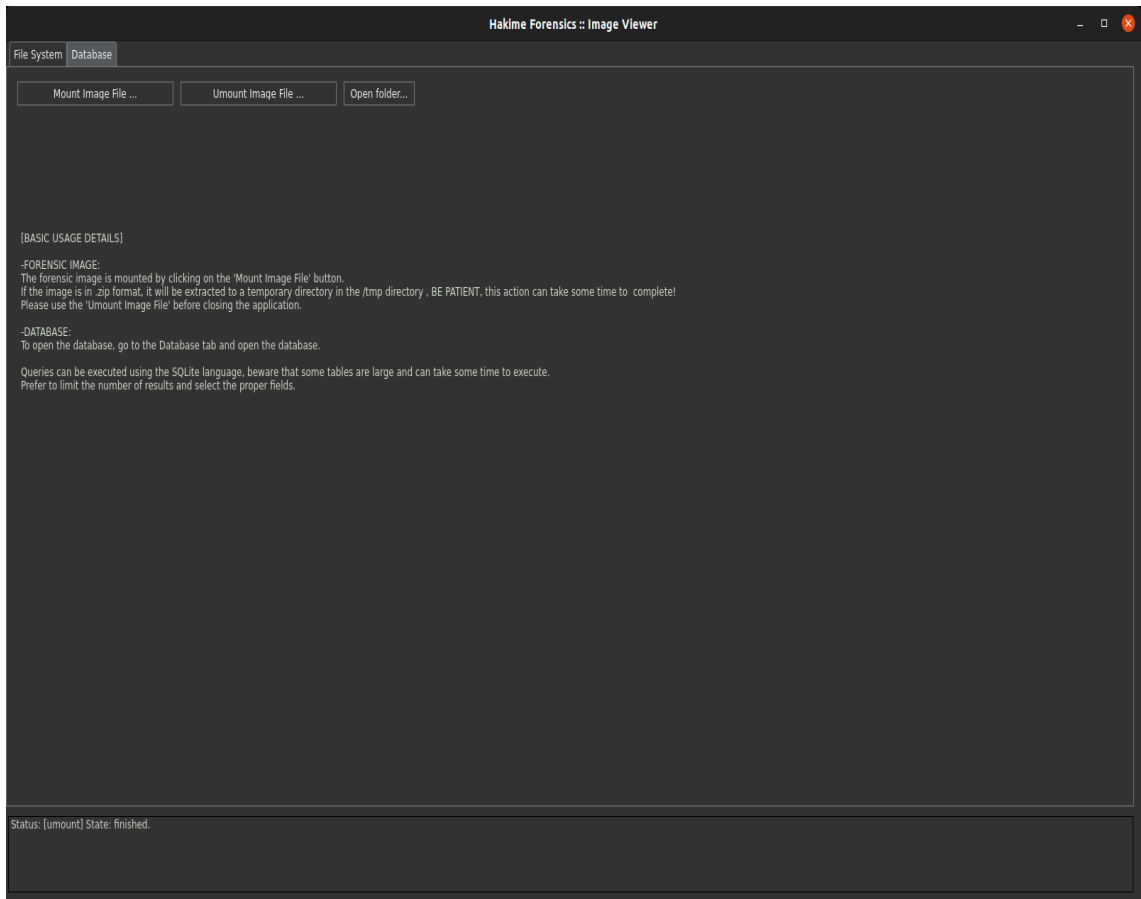


Figure 14 - Hakime Forensics Image Viewer - Startup Screen

3.23.2. Mount the Forensic Image File

3.23.2.1. Mount with the Hakime Forensics Image Viewer

To mount one forensic image, you have to click on “Mount Image File ...” and select the image file that you want to mount, as presented in the image below:

Implementation of tool to perform ECA during forensic collection

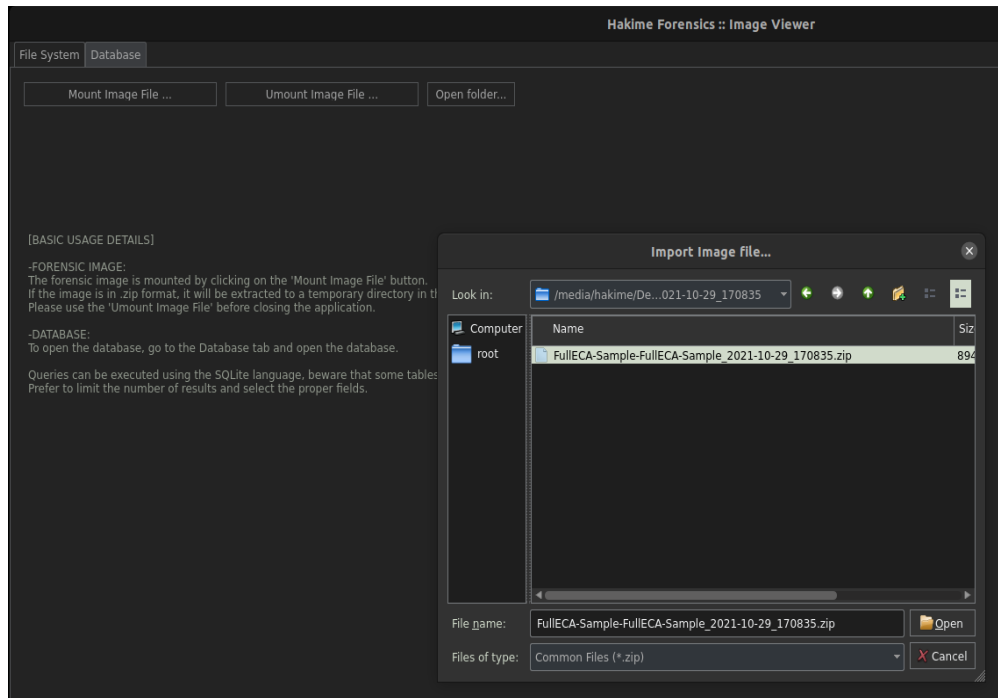


Figure 15 - Hakime Forensics Image Viewer - Mount Image File

If you select a compressed image (.zip file), it will automatically extract the .img to a temporary directory in the /tmp folder.

Once the image is mounted, the following screen will allow you to navigate through the folders. Double Click will allow you to open the desired file and save it on a place that you can view its contents.

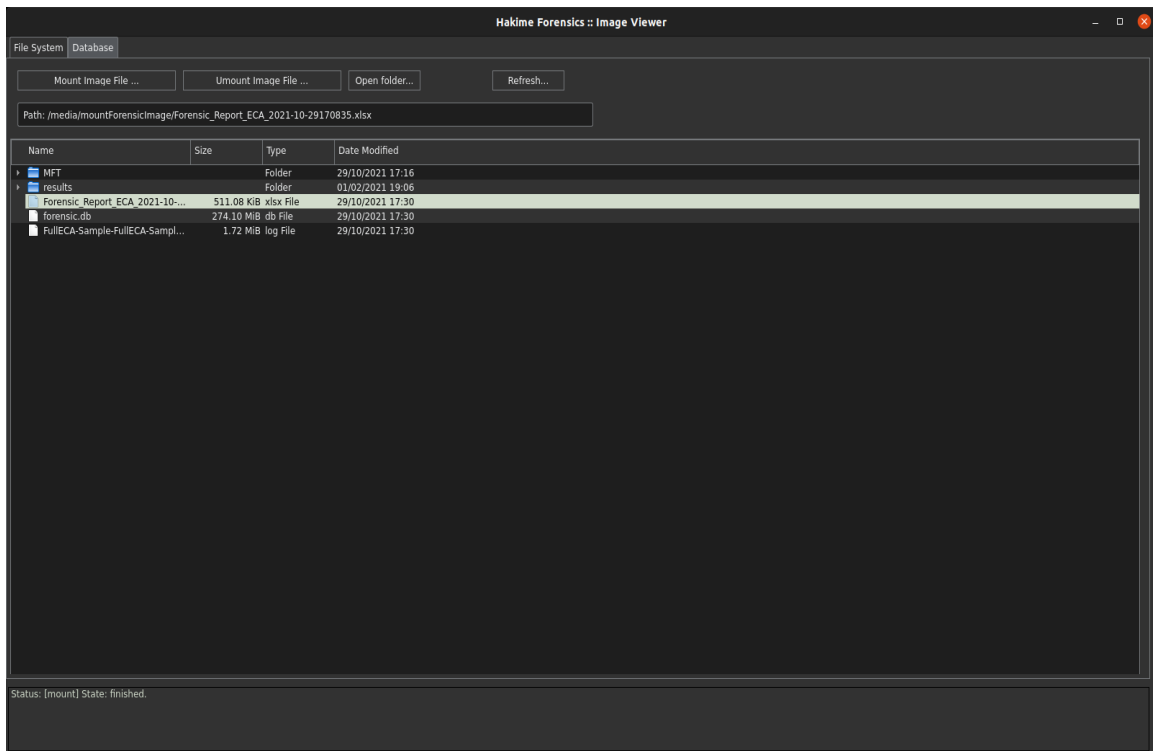


Figure 16 - Hakime Forensics Image Viewer - Image File Mounted

Implementation of tool to perform ECA during forensic collection

Optionally, you can click on “Open folder ...” to open the directory directly on Nautilus and browse through the file system.

After completing the analysis of the selected image, you can click on “Unmount Image File ...” to release the image. If you don’t click on this button, the image will be unmounted, and all temporary files will be removed when the application is closed.

3.23.2.2. Mount with 3rd party tools

As these images are created with focus on compatibility, you have the option to mount this image with your favorite tool.

On Linux, create a folder where to mount it (example /media/image), extract the *.img* file from the generated *.zip* file and use the command line:

```
sudo mount -t ntfs-3g -o ro,noload  
/media/destination/[CASE]/forensicimage.img /media/image
```

On Windows, use your preferred forensic mounting application (tested with *FTK Imager*, *OSFMount* and *Autopsy*) and mount the *.img* file as File System or Writable (don’t worry, these forensic tools will not actually write on the image but on a temporary file).

3.23.3. Investigate the index database

3.23.3.1. Navigate the index database with the Hakime Forensics Image Viewer

Open the application and click on “Open Forensic Image Viewer” on the main screen or via terminal type:

```
hakimeforensics viewer
```

This will open the Hakime Forensics Image Viewer, that enables you to analyze the contents of the image and the database without changing the contents.

This basic view enables execution of *SQL Queries* from front-end, just click on “Load Database ...” and select the ‘forensic.db’ file.

Implementation of tool to perform ECA during forensic collection

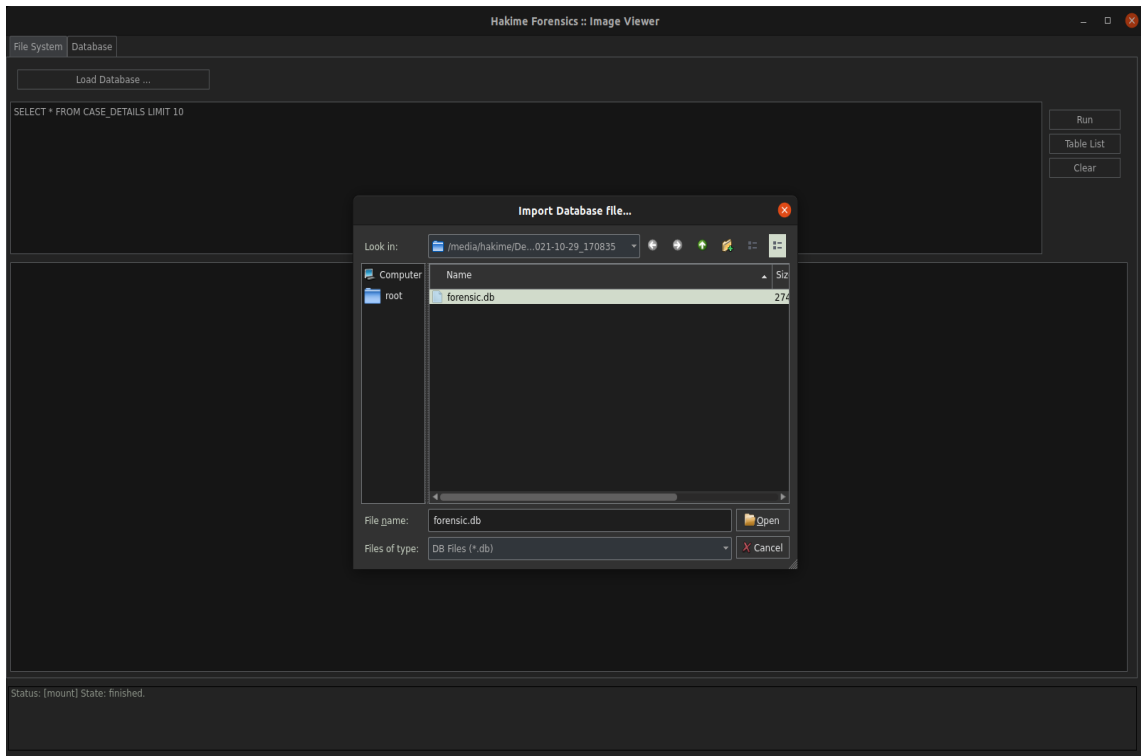


Figure 17 - Hakime Forensics Image Viewer - Mount database file

The button “Table List” on the right side of the window will show a list of available tables on the database and the number of records presents on each table. These tables contains all the preserved and parsed content.

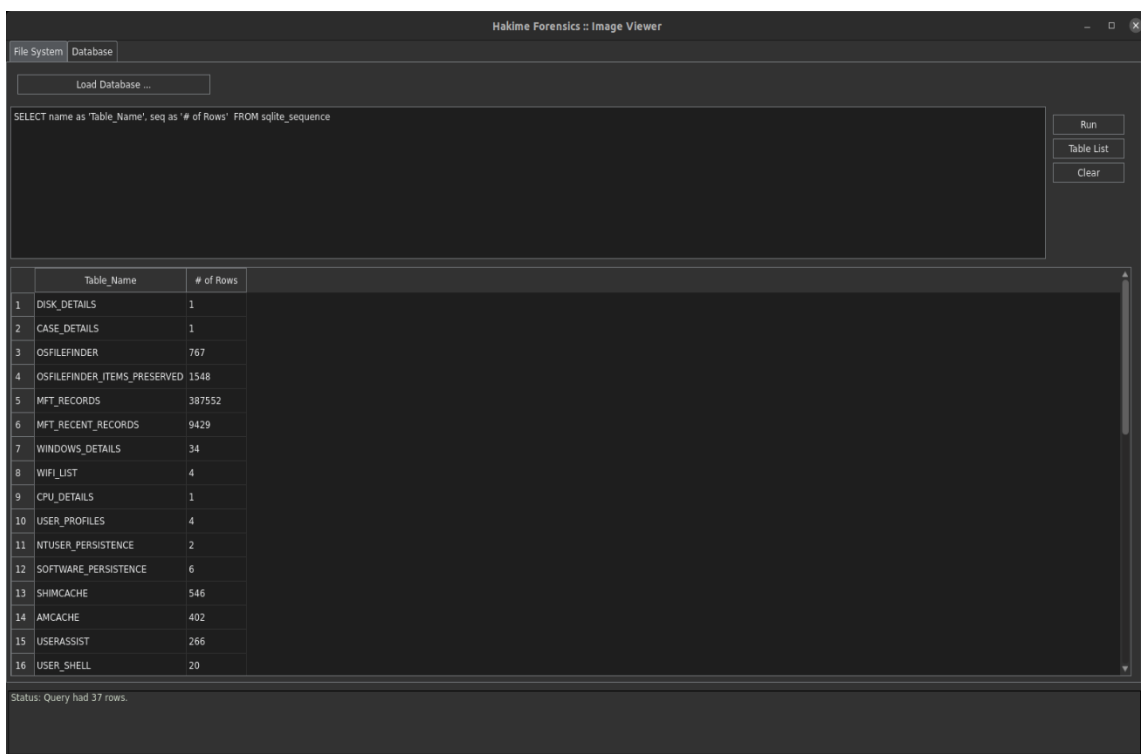


Figure 18 - Hakime Forensics Image Viewer - Table List

Implementation of tool to perform ECA during forensic collection

Queries can be written utilizing *SQLite* language and executed by clicking on “Run”. An important information is that complex queries that could generate large results can result in slowness or crashes.

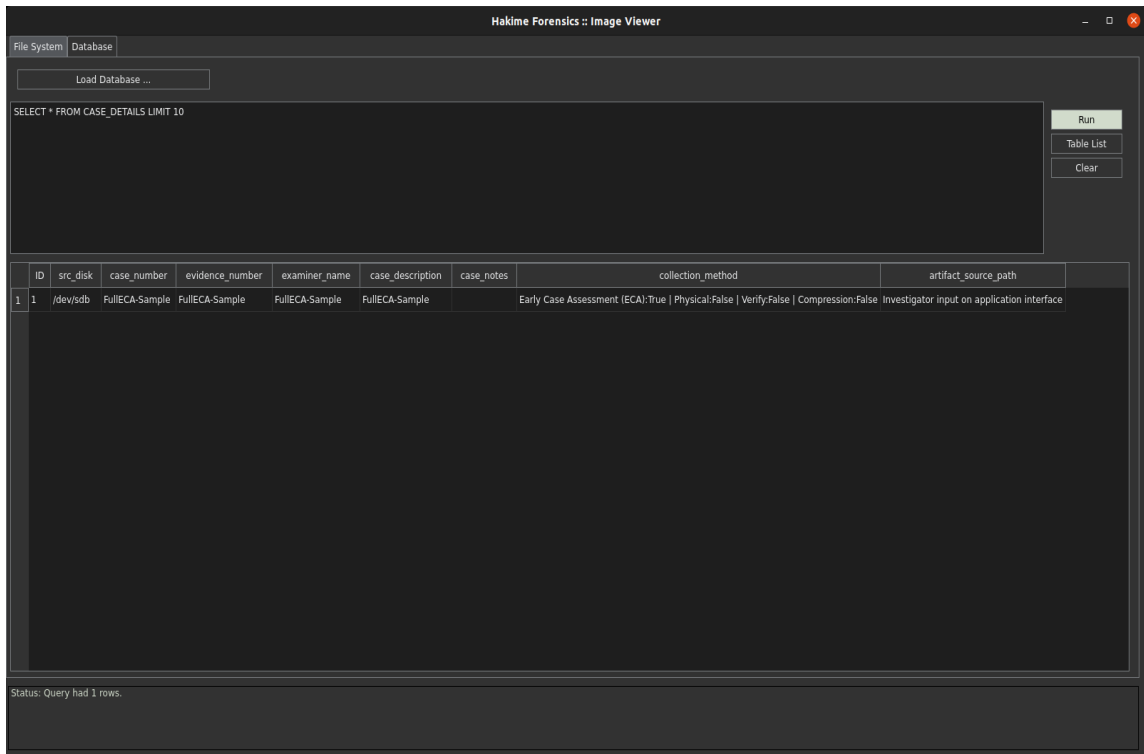


Figure 19 - Hakime Forensics Image Viewer - Simple Query Results (example)

3.23.3.2. Open the index database with 3rd party tools

Similar to the image file, the 'forensic.db' is a standard *SQLite* database, therefore it can be analyzed with any *SQLite* available tool.

This was tested with the open-source tool 'DB Browser for SQLite' both on Windows and Linux, but it should be compatible with any other *SQLite* tool on your platform.

4. Analysis of results

To compare the execution time and use common ground, all preservations were executed with the same source drive and in the same laptop, but varying the destination drives and adapter interfaces to cover as many potential scenarios as possible.

A detailed description of hardware model and details utilized to execute the tests along with respective pictures is organized on **Appendix III – Description of Hardware utilized on tests.**

The scenarios tested are described as follows:

1. #E001 – Preservation of ECA data only (all modules enabled). Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination SSD connected via SATA interface.
2. #E002 – Full physical disk preservation with verification. Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination SSD connected via SATA interface.
3. #E003 – Full physical disk preservation with verification and ECA data (all modules enabled). Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination SSD connected via SATA interface.
4. #E004 – Preservation of ECA data only (all modules enabled). Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination HDD external drive connected via USB interface (original Seagate enclosure).
5. #E005 – Full physical disk preservation with verification. Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination HDD external drive connected via USB interface (original Seagate enclosure).
6. #E006 – Full physical disk preservation with verification and ECA data (all modules enabled). Source SSD connected via USB interface (utilizing a Tableau Forensic SATA/IDE Bridge to USB3.0 model T35u), destination HDD external drive connected via USB interface (original Seagate enclosure).

Analysis of results

7. #E007 – Preservation of ECA data only (all modules enabled). Source SSD connected via SATA interface, destination HDD external drive connected via USB interface (original Seagate enclosure).
8. #E008 – Full physical disk preservation with verification. Source SSD connected via SATA interface, destination HDD external drive connected via USB interface (original Seagate enclosure).
9. #E009 – Full physical disk preservation with verification and ECA data (all modules enabled). Source SSD connected via SATA interface, destination HDD external drive connected via USB interface (original Seagate enclosure).
10. #E010 – Preservation of ECA data only (all modules enabled). Source SSD connected via SATA interface, destination SSD connected via USB interface (utilizing a JMicron JMS567 SATA/USB 3.0 Adapter).
11. #E011 – Full physical disk preservation with verification. Source SSD connected via SATA interface, destination SSD connected via USB interface (utilizing a JMicron JMS567 SATA/USB 3.0 Adapter).
12. #E012 – Full physical disk preservation with verification and ECA data (all modules enabled). Source SSD connected via SATA interface, destination SSD connected via USB interface (utilizing a JMicron JMS567 SATA/USB 3.0 Adapter).

The tool provides resulting logs with each preservation start time (counted from the moment when the “start button” is pressed), end time (that is considered when the hash of the final image is calculated) and elapsed time (defined as the difference in time between the end time and the start time).

These results for each execution have been organized and tabulated (Appendix II – Tabulation of execution times) and have been put together on the following graph.

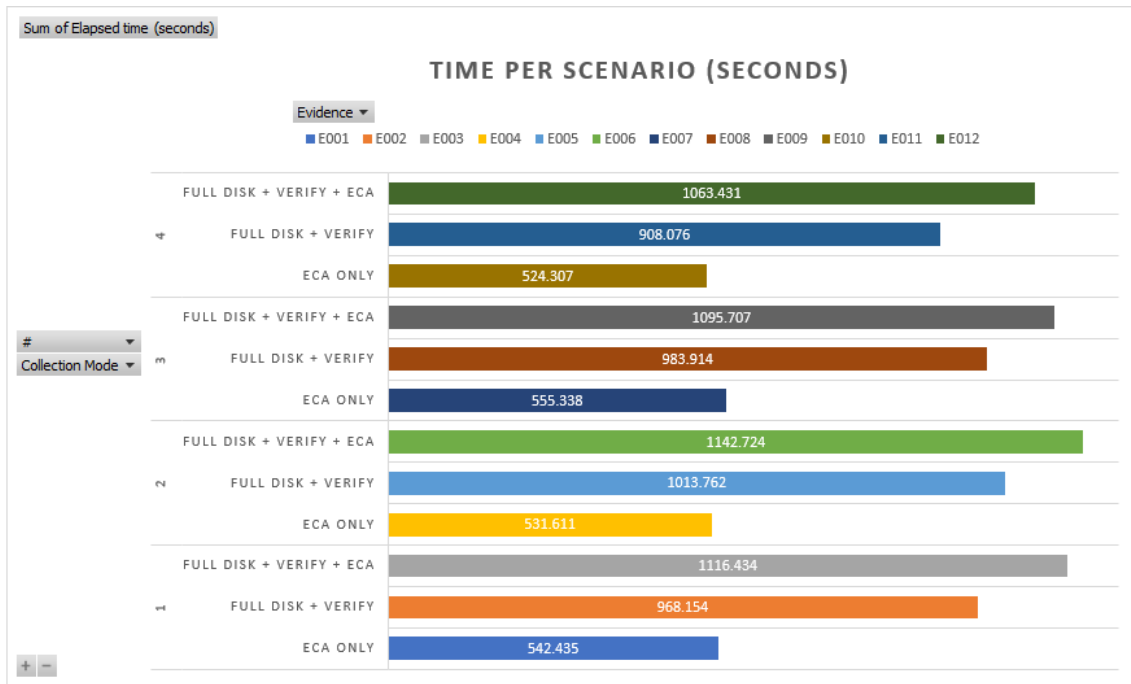


Figure 20 - Time per Scenario (seconds)

Analyzing the above graph, it is possible to conclude that the usage of the ECA modules represent an impact of, in average, 14.1% additional time in comparison with the execution of a full disk image without the pre-processing of the data, this reflects an additional 2 minutes and 16 seconds on the total preservation time if compared to the cases where ECA module was disabled.

In the other hand, if the ECA preservation and the Full Disk and Verification steps are executed separated, the total execution time is in average 6 minutes and 42 seconds longer (or 36.7% slower) than the scenarios where both functions were executed simultaneously, utilizing the spare hardware capacity in separate threads.

Nevertheless, it was observed a difference inferior to 11.6% between each tested scenarios with the same collection modes and different interfaces. This variation was expected and indicates that faster destination drives and the faster the connection interfaces the lower the general expected imaging time in ideal scenarios where the data source is the same and there is no physical damage in any of the factors.

Similarly, the usage of a source drive with lower speeds capabilities, different adapter interfaces or with damaged sectors could incur in different results.

All things considered, the table below summarizes the above-described analysis, providing a breakdown of each module, difference execution time and respective percentage:

Analysis of results

Analysis	Mode	Difference (maximum time - minimum time)	Difference (Percentage)
Difference time on interfaces	ECA Only	00:00:31	5.9%
Difference time on interfaces	Full Disk + Verify	00:01:46	11.6%
Difference time on interfaces	Full Disk + Verify + ECA	00:01:19	7.5%
Average additional execution time (separate steps)	Full Disk + Verify and separate ECA	00:06:42	36.4%
Average additional execution time with ECA (multithread)	Full Disk + Verify + ECA	00:02:16	14.1%

Table 3 - Summary of Scenarios Benchmark

In summary, the hypothesis provisioned an overhead of 30% additional time in the scenarios where the ECA parsing is executed compared to the average time needed to perform a full disk image with any standard forensic imaging tool, but the final results indicate that in average the execution time was lower than the expected threshold in all scenarios, with the maximum additional time observed in the executed scenarios of 17%.

With that in mind, considering a hypothetical scenario with the following parameters:

- The only investigated artifacts are the ones selected for ECA.
- The preservation and processing of the device is charged at a rate of 100 euros per hour.
- Each step takes exactly one hour.

Then executing this tool will have an increased cost of 14 euros on the preservation step, but would save the whole processing time, therefore would save 86 euros.

Recapitulating the main advantages of utilizing this tool instead of simply performing a forensic image and a separate processing in the laboratory, the following can be highlighted:

- Reduced overall time of preservation and processing steps.
- Consequently, reduced costs of processing time.
- Additional data sources (e.g.: network shared folders and USB devices) can be revealed while still in field.

- Errors can be quickly identified (e.g.: device details not matching with the label, suspect wiped the device and low volume of artifacts is available, list of users doesn't match with expected investigated individuals, etc.).
- Analysis of crucial details of the device can start while on site.
- A forensic-sound copy of crucial artifacts is ready in an separate container that can be used to start up the next analysis steps.
- Data is indexed and ready to search with standard tools from the market.
- Current forensic process is not changed, as the full disk is still preserved and can be processed with the standard workflows afterwards.

Conclusively, in the scenarios where the tool can preserve the ECA data, it can aggregate value to the final analysis as the final report can, at a minimum, indicate the presence of the users operating the device, different shutdown dates than expected, presence of anti-forensics techniques and potential external storage devices that could be found on the preservation location.

To put it differently, the tool attends all the requisites expected on the topic **III Hypothesis and previous** and has a final performance adequate, therefore the resulting report can be utilized on an investigation while still on the execution site and influence decisions.

Future challenges

The forensic pieces presented in this paper are intended to a situation when an initial triage is being performed, therefore the selected artifacts do not require deep knowledge of the device and case investigated to use it. With that said, it will focus on the preservation of some of the golden stones of information that can be used in the triage step, being relevant in a dead box analysis as much as in a live incident response.

Other very important files and evidence – such as the Event Logs, thumbnails and parsing of data stored within the *SQLite* databases used by various applications have been kept out of scope as they require more advanced techniques to be parsed or very broad comprehension of the environment to be utilized in its full potential.

In the same way, this research limited the analysis of artifacts present on the most recent version of the operating system Windows 10, thus evidence specific to older versions of this operating system or from different operating systems will likely not be properly parsed by the tool.

With that said, putting the tool scope limitations aside, new challenges are visible in the future of digital forensics, some of the ones that worth highlighting are the wide use of local encryption on individual laptops, the increasing use of cloud as main driver to store data, the reduced volume of recoverable data after deletion on solid state drives and data privacy regulations.

Local encryption becoming standard on devices is the highest challenge that a digital forensics investigator can find, as even if it is possible to overcome some of the encryption protocols, if the organization that manages the device does not control the encryption solution, the odds are that the investigator will be on the hands of the owner of the device.

One way that the tool can be improved in a future implementation may be to add a code that will identify potential encryption on the partition header source drive and skip the ECA steps totally if the encryption is detected, alerting the user of such limitation or, allow the user to select a partition mounted in advance, thus allowing the user to utilize third-party tools to provide the required keys to access the encrypted data.

Another limitation that can as well be solved is to adjust the tool installer, so it becomes less dependent on libraries and applications present on Ubuntu Linux, therefore possible to utilize in other Linux variants.

From the data analysis side, the tool can have the viewer improved to display some details of each preserved item once it is highlighted, this information is already present in the database and can be valuable to the investigator.

The same way, this application could go one step further in the EDRM workflow and generate a load file¹ ready in case the investigator intends to upload this preserved data directly in an eDiscovery review platform.

Likewise, the wide use of cloud services to store working data, in particular solutions that will work directly from the cloud to edit, manage, and present files and documents are a limitation that this tool may not be able to overcome, but in the other hand, the tool is already capable to identify some traces that indicate the use of cloud applications and highlight the items present on the drive that are part of the respective cloud.

In regard to the use of SSDs, the market of hard drives for personal computers is shrinking while the market of solid-state drives is increasing, this is happening because SSDs are much faster and have a lifespan higher than regular hard drives, becoming an advantage now that the prices have significantly dropped – from US\$493 per TB in 2017 to US\$ 128 in 2020 (according to a Wikibon research merging data from multiple indices), with expectation to be lower on USD\$ per TB by 2026 (according to a Research and Markets publication) and even with the shortage of covid-19 will still represent a grow of 14.94% specially because the energy consumption of a SSD for a datacenter is lower than a HDD, tendency that is already reflected in 2021, as more units of SSDs have been sold in the first quarter than HDDs (according to Trendfocus).

Computer forensics investigations are affected by this because the mode of operation of SSDs highly differ from the operation of hard drives. On regular hard drives when a user deletes a file, the item is simply flagged as deleted on the main file table so the operating system can reuse these sectors to allocate a new file, but while no new files are created, the content is still present on the unallocated area.

¹ A load file is a flat file that contains all metadata referent to a set of files and can be easily imported to databases for electronic discovery.

Future challenges

As described by VIEYRA et al. (2018) “In the case of HDDs, when the OS deletes a file, the OS updates the file allocation table and marks the area as unallocated. The underlying data is not deleted from the HDD. This becomes a problem for SSDs since they need to prepare deleted areas before allowing any new data to be saved in this area. SSDs are required to write in pages (usually 512 bytes) and delete in blocks”.

On most SSDs, on the other hand, there is a mechanism of garbage collection, often called TRIM, that runs with certain frequency and replaces the unallocated data with zeroes, highly reducing the chances of the data being recovered after deletion.

As BEDNAR et KATOS (2011) explains, an SSD disk “offers an array of logical block addresses to the host, but the internal organization depends upon complex algorithms. One disadvantage of the SSD technology over its predecessor is that existing data must be erased before blocks can be reused (e.g., they cannot simply be overwritten)”.

Similarly, aspect that can influence a digital forensics investigation is the jurisdiction and applicable laws and regulations where the data is located. Depending on the situation, laws may limit the type of data that can be preserved and analyzed, one example is the situation where an investigated uses a personal device to execute corporate activities and the warrant only allows the preservation of specific folders within the required device.

In this direction, SHOOK (2014) argues that “*Bring Your Own Device* or *BYOD* phenomenon can further complicate these issues. BYOD refers to employees using devices that are not corporate owned but are being used for business purposes. The lack of clear-cut ownership by the organization can complicate issues of the right to directly access or request data from the device, and which party may bear responsibility for data that is lost”.

Thereupon, this is a moment where the developed tool can be handy, as it will allow the investigator to limit the folders or specific file types for the early case assessment.

Even though the tool counts with an extensive number of forensic artifacts already catalogued and it was developed with the perspective of being extensible - therefore most modules use a json-like format that can be modified to point new sources of data -, this is a point that could greatly improve by the community that could

submit and create modified versions of the list to preserve specific data sources for different scenarios.

As you can see, there are multiple trends that the tool could be improved in future developments, but those all fall out of the scope planned for this paper. Some of them include the implementation of a version of this tool that works on live Windows operating systems, therefore allowing the preservation of all the data even if the drive is encrypted.

Future work may be possible to improve this list with steps to harvest value from more complex parts of evidence as well present methods to connect them in order to get better insights or automate investigation processes.

Overall, even if the tool uses more than one thread for some tasks, it still doesn't use all potential available in the hardware and could be improved to prioritize some of the tasks and run multiple parsers at the same time, this way some of the information can be displayed to the investigator even while the execution is still in progress.

Conclusion

The developed tool shows great efficacy to preserve and anticipate the analysis of valuable information from an offline disk without encryption, producing a report that can be read instantly during the preservation.

In the first place, it has potential to improve the processes, as in a normal process, the data would have to be ingested in a forensic processing tool to later be analyzed to identify multiple types of indicators, with this tool some results are rather automatically generated while the full preservation is still in course.

Additionally, the tool works enables the investigator to connect the device and proceed with a ECA only preservation for the cases when a full disk preservation is not needed or was already performed with a different method.

In either case, the ECA report generated has information that can be used to identify the presence of external storage devices, network resources, presence of multiple user profiles on the device, network connections, recently opened files and applications.

Future work could integrate the ECA database in a central application allowing data correlation with multiple custodians on a same investigation, potentially reducing the need of a forensic indexing of the image in between, reducing the costs, steps, and execution times.

Specially in scenarios where the data source is very large or have slow transfer rates, but the volume of data present is relatively small, as the full disk preservation would still have to go through all disk sectors.

This is particularly relevant when dealing with large capacity SSDs that would not have much data available to be recovered from deletion, therefore the forensic ingestion would rely only on data present on the live environment and allocated clusters.

Enforcement of data privacy regulations can compel that only specific data of a user be preserved, especially if a subpoena limits the search on a specific device, on that account, an ECA preservation would fit within the stricter requirements.

Along the way of the development and research it was clear that there are some changes on the way of digital forensics that have the potential to bring more and more limitations to the presented approach for those specific scenarios.

Among the potential challenges, it is relevant to consider the widespread use of encryption and cloud platforms by the users and organizations, as well the higher usage of multiple digital devices and applications and finally, the all-time-changing nature of information technologies.

Noteworthy that those limitations will not be exclusive of this tool, but rather a general challenge that the digital forensics community will need to overcome by creating new approaches and redesign processes.

In the long run, the presence of encrypted SSDs and higher use of cloud resources is going towards a point that dead box preservations with standard tools on the market will not be the most effective way to recover data as, in those scenarios, information will be either unreachable behind encryption or not present on the device at all.

Furthermore, depending on the user's or organization preferences, the data can be automatically backed up in different data sources that can be preserved with different approaches.

On balance, the results showed that in all scenarios the additional time used for pre-processing of the ECA data was lower than the actual full disk preservation, not presenting a relevant increase on the total execution time but providing relevant insights to the investigation in an anticipated moment.

Conclusively the data present in the report can indeed reveal additional devices to be preserved that would be potentially overlooked in a data collection which relies on the data identification solely on human prone error methods.

However, the tool could be improved to either be more integrated generating the results to a final database, or be adapted to be used in live environments generating the reports on scenarios where the device is encrypted but the encryption is suspended when the OS is logged in.

In the final analysis, while the scenarios where the storage device is not encrypted or the encryption can be suspended (either by utilizing a known encryption key or corporate policy), this tool has still an innovative approach and can bring relevant results for the long run.

Bibliographic References

ACCESSDATA. "FTK Imager User Guide". Available at <<https://accessdata.com/product-download/ftk-imager-version-4-5>>. Access in 15 oct 2021.

ALTHEIDE, C. and Carvey, H., 2011. Digital forensics with open source tools. Elsevier.

ARORA, R. ed., 2016. Conquering Big Data with high performance computing. Springer International Publishing.

ATTOE, R., 2016. Digital forensics in an eDiscovery world. In Digital Forensics (pp. 85-98). Syngress. Available at <https://www.sciencedirect.com/science/article/pii/B978012804526800006X>.

BEDNAR, P. and Katos, V., 2011. SSD: New challenges for digital forensics. In ItAIS 2011, Proceedings of the 8th Conference of the Italian Chapter of the Association for Information Systems (pp. 1-8). ItAIS. Available at <<https://lucris.lub.lu.se/ws/files/5456453/4318024.pdf>>. Access in 17/02/2021.

BELKASOFT. "Belkasoft Evidence Center User Reference". Available at <<https://belkasoft.com/downloads/info/Evidence%20Center%20Help.pdf>>. Access in 22 Aug 2021.

BERNDTSSON, Hansson, Olsson and Lundell (2008) Thesis Projects - A Guide for Students in Computer Science and Information Systems, Second Edition., Springer-Verlag.

BUNTING, S. and Wei, W., 2006. EnCase Computer Forensics: The Official EnCE: EnCase Certified Examiner Study Guide. John Wiley & Sons.

CANDEIAS, Mario. "Ataques fulminantes em sistemas operativos Microsoft Windows" , Dissertação de mestrado, Escola Superior de Tecnologia e Gestão. Instituto Politécnico de Beja. Portugal, 2015. Available at <<https://repositorio.ipbeja.pt/handle/20.500.12207/4594>>. Access in 14 dec 2019.

CARRIER, Brian. "Autopsy 4". 15 mar 2016. Available at <<http://www.sleuthkit.org/>>. Access in 26 Jul 2021.

CARROLL, L., 2013. The grossman-cormack glossary of technology-assisted review. *Federal Courts Law Review*, 7(1). Available at <https://www.fclr.org/fclr/articles/html/2010/grossman.pdf>

COHEN, Michael; GARFINKEL, Simson; SCHATZ, Bradley. Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *digital investigation*, v. 6, p. S57-S68, 2009.

COUGHLIN, Tom (2016) "The Costs of Storage". *Forbes*. Access in 09 Nov. 2019. Available at: <https://www.forbes.com/sites/tomcoughlin/2016/07/24/the-costs-of-storage/>

COWEN, David. "USBDeviceForensics". 2015. Available at <https://github.com/woanware/usbdeviceforensics>. Access at 11 jun 2020.

CRESSEY, Donald R. The criminal violation of financial trust. *American sociological review*, v. 15, n. 6, p. 738-743, 1950.

DA SILVA ELEUTÉRIO, Pedro Monteiro, and Marcio Pereira Machado. *Desvendando a computação forense*. Novatec Editora, 2019.

DB4S. "DB Browser for SQLite". 2003. Available at <https://sqlitebrowser.org/>. Access in 3 feb 2020.

EARLY CASE ASSESSMENT (ECA). <http://web.archive.org/web/20160512061452/http://www.edrm.net/resources/glossaries/grossman-cormack/eca>. Accessed 4/05/2020.

ELECTRONIC DISCOVERY REFERENCE MODEL (EDRM - 2020). Duke Law School. Available at: <http://www.edrm.net> . Access in 09 nov. 2020.

FLOYER, David. 2021, "QLC Flash HAMRs HDD". WIKIBON. Available at <https://wikibon.com/qlc-flash-hamrs-hdd/>. Access at 01/05/2021.

GANTZ J., Reinsel D., Rydning J. (2018). *The Digitization of the World - From Edge to Core*. Doc# US44413318, November 2018, An IDC White Paper, sponsored by SEAGATE. Access in 09 Nov. 2019. Available at: <https://seagate.com/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>

GARFINKEL, Simson L. HELLEWELL, Phillip. "The Advanced Forensic Format Library and Tools Version 3". 2005. Available at <https://github.com/sshock/AFFLIBv3>. Access in 14 jul 2021.

Bibliographic References

GARTNER. 2021. "Gartner Forecasts Worldwide Security and Risk Management Spending to Exceed \$150 Billion in 2021". <
<https://www.gartner.com/en/newsroom/press-releases/2021-05-17-gartner-forecasts-worldwide-security-and-risk-managem> >. Access in 14/11/2021.

PALMER, Gary. 2001 "A Roadmap for Digital Forensic Research." Digital Forensics Research Workshop (DFRWS). Technical report DTR-T0010-01, Utica, New York
https://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf)

GRAJEDA, C., Sanchez, L., Baggili, I., Clark, D. and Breitingner, F., 2018. Experience constructing the Artifact Genome Project (AGP): Managing the domain's knowledge one artifact at a time. Digital Investigation, 26, pp.S47-S58. Available at <
<https://agp.newhaven.edu/about/start/> >. Access in 31/05/2021.

HARICHANDRAN, V.S., Walnycky, D., Baggili, I. and Breitingner, F., 2016. Cufa: A more formal definition for digital forensic artifacts. Digital Investigation, 18, pp.S125-S137. Access in 09 nov. 2019. Available at:
<https://digitalcommons.newhaven.edu/cgi/viewcontent.cgi?article=1055&context=electricalcomputerengineering-facpubs>

HASSAN, N.A., 2019. Digital Forensics Basics: A Practical Guide Using Windows OS. Apress.

ISO/IEC 27037: 2012. Information Technology: Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence. International Organization for Standardization (ISO), published in October 2012.

JOHANSEN, G., 2017. Digital forensics and incident response: an intelligent way to respond to attacks. Packt Publishing Ltd.

KLOET, B., Metz, J., Mora, R.J., Loveall, D. and Schreiber, D., libewf: project info.(2008). <https://github.com/libyal/libewf>

LAYKIN, E., 2013. Investigative computer forensics: the practical guide for lawyers, accountants, investigators, and business executives. John Wiley & Sons.

LEE, R. "Windows Forensics Analysis Poster" Edition DFPS_FOR500_v4.9_4-19. 2019. Access in 09 nov. 2019. Available at: <https://www.sans.org/security-resources/posters/windows-forensic-analysis/170/download>

LIBRATOM. "Review, Appraisal, and Triage of Mail". University of North Carolina at Chapel Hill, 2019. Available at <<https://ratom.web.unc.edu/>>. Access in 14 apr 2020.

LIBSCCA. "Libscca is a library to access the Windows Prefetch File (SCCA) format". 2014. Available at <<https://github.com/libyal/libscca>>. Access in 2 may 2020.

LIN, X., Lin, X. and Lagerstrom-Fife, 2018. Introductory Computer Forensics. Springer International Publishing.

LININGER, Joseph et al. dc3dd. <https://sourceforge.net/projects/dc3dd/>. 2007.

LOCARD, Edmond, 1934. La police et les méthodes scientifiques. Rieder.

LOSSIO, Claudio, 2021. "Forensics Data Acquisition and analysis of a Ransomware". eForensics Magazine, vol 10 no. 04, pp.5-12.

LYLE, James R., 2002. Testing disk imaging tools. Digital Investigation. International Journal of Digital Evidence. Winter 2003, Volume 1, Issue 4. Available at <<https://www.utica.edu/academic/institutes/ecii/publications/articles/A04BC142-F4C3-EB2B-462CCC0C887B3CBE.pdf>>. Access in 14/10/2020.

MARTIN, J.P. and Cendrowski, H., 2014. Cloud computing and electronic discovery. John Wiley & Sons.

MCQUAID, Jamie. Magnet Forensics (2014) Forensic analysis of LNK files. Available online: <https://www.magnetforensics.com/blog/forensic-analysis-of-lnk-files/>. Fetched March 22, 2018.

METZ, Joachim (2012). Expert Witness Format version 2 (EWF 2) specification. Guidance Software. Access in 09 Nov. 2019. Available at: <https://github.com/libyal/libewf/blob/master/documentation/Expert%20Witness%20Compression%20Format%20%20%28EWF2%29.asciidoc>.

METZ, Joachim et. al. libewf: project info. <https://github.com/libyal/libewf>. 2006. Access in 5 oct 2020.

METZ, Joachim. 2019. "Digital Forensics Artifact Repository". Available at <<https://github.com/ForensicArtifacts/artifacts>>. Access in 15/11/2019.

METZ, Joachim. MORA, Robert-Jan. 2006. "An Introduction to the libewf Expert Witness Library" <<http://www.sleuthkit.org/informer/sleuthkit-informer-23.html>>. Access in 20/04/2021.

Bibliographic References

NICHOLAS, H. Dcfldd. <http://dcfldd.sourceforge.net/>, 2002. Access in 1 sep 2020.

NIKKEL, B., 2016. Practical forensic imaging: securing digital evidence with Linux tools. No Starch Press.

NIST. 2004. "Digital Data Acquisition Tool Specification". National institute of Standards and Technology. United States. Available at <<https://www.nist.gov/system/files/documents/2017/05/09/pub-draft-1-dda-require.pdf>>. Access in 29/04/2020.

O'CONNOR, T.J., 2010. Grow Your Own Forensic Tools: A Taxonomy of Python Libraries Helpful for Forensic Analysis. SANS Institute. < <https://www.sans.org/reading-room/whitepapers/incident/paper/33453> >. Access in 18/01/2021.

OETTINGER, William, 2020. Learn Computer Forensics: A beginner's guide to searching, analyzing, and securing digital evidence. Packt Publishing Ltd.

OPENTEXT. "Encase Forensic User Guide". Available at <<http://encase-docs.opentext.com/documentation/encase/forensic/8.07/Content/Resources/External%20Files/EnCase%20Forensic%20v8.07%20User%20Guide.pdf>>. Access in 7 nov 2021.

OSFMOUNT. PassMark Software. 2010. Available at <<https://www.osforensics.com/tools/mount-disk-images.html>>. Access in 18 jun 2021.

PALMER, Gary. 2001 "A Roadmap for Digital Forensic Research." Digital Forensics Research Workshop (DFRWS). Technical report DTR-T0010-01, Utica, New York. <https://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf >. Access in 29 nov. 2020.

PHILIPP, A., Cowen, D. and Davis, C., 2009. Hacking exposed computer forensics. McGraw-Hill, Inc.

PYQT5. "PyQt5 Reference Guide". Available at <<http://pyqt.sourceforge.net/Docs/PyQt5/>>. Access in 01 nov 2020.

RESEARCH AND MARKETS, 2021. Solid State Drive (SSD) Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). Research and Markets. Available at <<https://www.researchandmarkets.com/reports/4602294/solid-state-drive-ssd-market-growth-trends>>. Access in 16/03/2021.

ROUSSEV, V., Quates, C. and Martell, R., 2013. Real-time digital forensics and triage. *Digital Investigation*, 10(2), pp.158-167.

SANTOS, CLEORBETE (2018). *Segurança Digital*. 1st edition. Clube de Autores.

SARANDY, Flávio Marcos Silva; RODRIGUES, Alberto Tosi (Comp.). *Modelo básico para elaboração de um projeto de pesquisa*. UFRGS. Available at: http://www.ufrgs.br/laviecs/biblioteca/arquivos/como_fazer_%20pesquisa.pdf. Access in 09 Nov. 2019.

SCHULER, K.A., 2011. *E-discovery: creating and managing an enterprisewide program: a technical guide to digital investigation and litigation support*. Syngress.

SELINGER, Peter. MD5 collision demo. 2011. Access in 26 dec. 2020. Available at: <https://www.mscs.dal.ca/~selinger/md5collision/>

SHAABAN, A. and Saprionov, K., 2016. *Practical Windows Forensics*. Packt Publishing Ltd.

SHASHIDHAR, N.K. and Novak, D., 2015. Digital forensic analysis on prefetch files. *International Journal of Information Security Science*, 4(2), pp.39-49.

SHILOV, Anton. 2021. "SSDs Outsell HDDs in Unit Sales 3:2: 99 Million Vs. 64 Million in Q1". *Tom's Hardware*. Available at <https://www.tomshardware.com/news/ssd-market-shares-q1-2021-trendfocus>. Access in 25/05/2021.

SHOOK, James (2014). "E-Discovery Data Mapping: A Practical guide for lawyers". Amazon Kindle.

SHUTIL. "High-level file operations". Available at <https://docs.python.org/3/library/shutil.html>. Access in 11 apr 2020.

SHUTIL. 2021. High level file operations. Python Docs. <https://docs.python.org/3/library/shutil.html>

SKULKIN, O. and de Courcier, S., 2017. *Windows Forensics Cookbook*. Packt Publishing Ltd.

SNYDER, Peter, 1990, October. tmpfs: A virtual memory file system. In *Proceedings of the autumn 1990 EUUG Conference* (pp. 241-248). <http://www.sunhelp.org/history/pdf/tmpfs.pdf>

Bibliographic References

SONDHI S., Arora R. (2016) Big Data Processing in the eDiscovery Domain. In: Arora R. (eds) Conquering Big Data with High Performance Computing. Springer, Cham. https://doi.org/10.1007/978-3-319-33742-5_14

SONDHI, S. and Arora, R., 2014, July. Applying lessons from e-Discovery to process Big Data using HPC. In Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment (pp. 1-2). Available at <https://dl.acm.org/doi/abs/10.1145/2616498.2616525>

SONI, M. and Pathak, S.R., 2015. A Review of Forensic Artifacts in a Windows 8 Environment. International Journal of Computer Applications, 975, p.8887. Access in 09 nov. 2019. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.6434&rep=rep1&type=pdf>

SQLITE. SQLite Consortium, 2000. Available at <<https://www.sqlite.org/index.html>>. Access in 1 feb 2020.

STATCOUNTER, G. S. "Statcounter global stats." (2019). Available at <<https://gs.statcounter.com/>>. Access in 20/11/2019.

SUMMERFIELD, M., 2007. Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming (paperback). Pearson Education.

T35u, Tableau. "Tableau Forensic SATA/IDE Bridge". Available at <<https://security.opentext.com/tableau/hardware/details/t35u>>. Access at 6 nov 2021.

VAN DIJK, D., Graus, D., Ren, Z., Henseler, H. and de Rijke, M., 2015. Who is involved? Semantic search for e-discovery. In ICAIL DESI VI Workshop. Available at <http://users.umiacs.umd.edu/~oard/desi6/papers/vanDijk-final.pdf>

VERIZON. 2020. "2020 Data Breach Investigations Report (DBIR)". Available at <<https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf>>. Access in 21/11/2020.

VIEYRA, John & Scanlon, Mark & Le-Khac, Nhien-An. (2018). Solid State Drive Forensics: Where do we stand?. <https://www.researchgate.net/publication/325976653_Solid_State_Drive_Forensics_Where_Do_We_Stand>

WANG, X., & Yu, H. (2005, May). How to break MD5 and other hash functions. In Annual international conference on the theory and applications of cryptographic techniques (pp. 19-35). Springer, Berlin, Heidelberg. Access in 26 dec. 2020. Originally available at: <http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf>. Archived at: <https://web.archive.org/web/20050418022826/http://www.infosec.sdu.edu.cn/paper/md5-attack.pdf>

Appendix I – Forensic artifact locations

In this section are presented the tables with the various forensic artifacts classifications used in this work.

Artifact Description	Type	Location
Last Login and last password change Last login and password changes on the device	SAM - Registry Hive	➤ %SYSTEMROOT%\system32\config\SAM
List of users Default registry for user profiles in some Windows versions	SAM - Registry Hive	➤ SAM\SAM\Domains\Account\Users\
User Folders Default folders for user profiles in some Windows versions	Folder	➤ C:\Windows\profiles\ ➤ C:\Documents and Settings\ ➤ C:\Users\

Table 4 - Forensic Artifacts - User Profiles

Artifact Description	Type	Location
Last shutdown date/time Registry key with the date of the last shutdown	SYSTEM - Registry Hive	➤ SYSTEM\ControlSet###\Control\ Windows / ShutdownTime
Installed build number - indicates which operating system version is installed Registry key that shows the current version of Windows installed (build number)	SOFTWARE - Registry Hive	➤ SOFTWARE\Microsoft\Windows NT\CurrentVersion\CurrentBuildNumber

Computer Name Registry key with details of the computer name	SYSTEM - Registry Hive	SYSTEM \ CurrentControlSet \ Control \ ComputerName \ ActiveComputerName
Date of the last update Registry with details of the installation of the OS last updates	SOFTWARE - Registry Hive	SOFTWARE\Microsoft\Windows NT\CurrentVersion\InstallTime

Table 5 - Forensic Artifacts - Operating System Details

Artifact Description	Type	Location
Power Efficiency Report Report that can shows some diagnostics from the hardware	File	%SYSTEMROOT%\System32\energy-report.html
Windows update details Basic details of the windows update date	SOFTWARE - Registry Hive	SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate
Hardware details Volatile registry that can contain information of hardware (can be accessed in live environment or in registry backups)	HARDWARE - Volatile Registry only available on live systems	HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System
Computer Name and Volume Serial Number Details of the hardware where the drive was connected when the OS was installed	SYSTEM - Registry Hive and NTUSER.DAT - User Hive	NTUSER.DAT\Software\Microsoft\Windows Media\WMSDK\General\ SYSTEM\ControlSet###\Control\ComputerName\ComputerName\
Quantity CPU (Cores) in the host	SYSTEM - Registry Hive	SYSTEM\ControlSet###\Control\Session Manager\ Environment\NUMBER_OF_PROCESSORS\

Appendix I – Forensic artifact locations

Registry that contains details of the CPU		
-------------------------------------------	--	--

Table 6 - Forensic Artifacts - Hardware Details

Artifact Description	Type	Location
Time zone information Contains the time zone that the device is registered	SYSTEM - Registry Hive	SYSTEM\ControlSet###\Control\TimeZoneInformation

Table 7 - Forensic Artifacts - Timezone

Artifact Description	Type	Location
Regular application installation Default software installation registry for 32bit applications in some Windows versions	SOFTWARE - Registry Hive	SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\
Uninstalled Applications Registry of uninstalled applications in some Windows versions	SOFTWARE - Registry Hive	SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
64Bit installation Default software installation registry for 64bit applications in some Windows versions	SOFTWARE - Registry Hive	SOFTWARE\Wow6432Node\
UserAssist Registry with a list of programs executes in the device	SOFTWARE - Registry Hive	HKEY_CURRENT_USER \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ UserAssist
Installation folders Default software installation folders in some Windows versions	Folder	> \%USERPROFILE%\AppData\ > \Program files\ > \Program Files (x86)\ > \ProgramData\

Table 8- Forensic Artifacts - Installed Applications

Artifact Description	Type	Location
SkyDrive Database with settings of SkyDrive profile in some versions	Settings.dat Database File	–> Settings.dat\RoamingState (SkyDrive User Name) –> Settings.dat\LocalState\Platform (SkyDrive E-Mail Account Name)
SkyDrive configurations and logs Logs and databases that contain history of usage of SkyDrive cloud storage	Files	–> %USERPROFILE%\AppData\Microsoft\SkyDrive\logs*.log –> %USERPROFILE%\AppData\Microsoft\SkyDrive\setup\logs*.log –> %USERPROFILE%\AppData\Microsoft\SkyDrive\settings\
OneDrive logs and files Logs and databases that contain history of usage of OneDrive cloud storage	Files	–> %USERPROFILE%\OneDrive\ (stores the local files) –> %USERPROFILE%\AppData\ OneDrive\ (stores logs and cache)
OneDrive Settings Database with settings of OneDrive profile in some versions	Settings.dat Database File	–> Settings.dat\LocalState\ –> %USERPROFILE%\AppData\Local\Microsoft\OneDrive\logs
OneDrive Registry Hives	NTUSER.DAT – Registry Hive	–> NTUSER.DAT\SOFTWARE\Microsoft\Office\16.0\Common\Identity\Identities_LiveId –> NTUSER.DAT\SOFTWARE\Microsoft\AuthCookies\Live\Default\CAW
Dropbox Logs and databases that contain history of usage of Dropbox cloud storage	Database file	–> %USERPROFILE%\AppData\Dropbox*.db* –> %USERPROFILE%\AppData\Local\Microsoft\Dropbox
Google Drive Logs and databases that contain history of usage of Google Drive cloud storage	Files	–> %USERPROFILE%\AppData\Google\Drive\snapshot.db –> %USERPROFILE%\AppData\Google\Drive\sync_config.db –> %USERPROFILE%\AppData\Google\Drive\sync_config.log*

Appendix I – Forensic artifact locations

		<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Google\Drive\user_default\snapshot.db ➤ %USERPROFILE%\AppData\Google\Drive\user_default\sync_config.db ➤ %USERPROFILE%\AppData\Google\Drive\user_default\sync_config.log*
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 9- Forensic Artifacts - Cloud Applications

Artifact Description	Type	Location
Skype User profiles List of users that access Skype on some versions	NTUSER.DAT – Registry Hive	➤ NTUSER.DAT\SOFTWARE\Skype\Phone\Users\ (Skype User List)
Skype User profile Settings Database with settings of skype profile in some versions	Settings.dat database	➤ settings.dat\LocalState\skype.liveuser.CID (Skype User Name E-Mail)
Skype Installation Registry that indicates that skype was installed on the device	SOFTWARE – Registry Hive	➤ SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Components\{(UID)
Skype database Skype chat database, may contain skype conversation	Database File	➤ %USERPROFILE%\Application\Skype\<skype-name>main.db (on Windows XP) ➤ %USERPROFILE%\AppData\Roaming\Skype\<skype-name>\main.db (Windows 7, 8 and 10) ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Skype for Desktop \ IndexedDB\file__0.indexeddb.leveldb\ (Skype for Desktop version 8 or newer on Windows 10)

Table 10- Forensic Artifacts - Skype

Artifact Description	Type	Location
iPhone, iPad Mounting	NTUSER.DAT – Registry Hive	➤ SYSTEM\ControlSet001\Enum\USB\

Appendix I – Forensic artifact locations

History of mounted devices via USB		
iTunes Backups Backups of IOS devices (iPhones, iPads, apple watches and iPods)	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\ AppData \ Roaming\ Apple \ MobileSync \ Backup ➤ %USERPROFILE% \ Application Data \ Apple Computer \ MobileSync \ Backup

Table 11 - Forensic Artifacts - iTunes Backups

Artifact Description	Type	Location
Recycle Bin folder Folder where files are stored after a soft deletion	Folder	<ul style="list-style-type: none"> ➤ C:\\$Recycle.bin (Windows 7 and newer) ➤ C:\RECYCLER (Windows 2000, NT, XP and 2003)
Recycle Bin registry Configurations of the Recycle Bin on registry	SOFTWARE Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\BitBucket\Volume\ ➤ SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\BitBucket\

Table 12 - Forensic Artifacts - Recycle Bin

Artifact Description	Type	Location
MFT Main file table – may contain a list of files that have been present on the operating system	MFT file	➤ \$MFT
MFTMirr Mirror of the main file table – may contain a list of files that have been present on the operating system	MFT file	➤ \$MFTMirr

Table 13 - Forensic Artifacts - Main File Table (MFT)

Artifact Description	Type	Location
VMWare Application used to create and execute virtual machines	NTUSER.DAT – Registry Hive	➤ NTUSER.DAT\Software\VMware, Inc.\VMWare Player\VMplayer\Window position

Appendix I – Forensic artifact locations

<p>WSL folder</p> <p>Windows feature that enables user to run a different operating system as an application – files that indicate that the feature was enabled</p>	Files	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Lxss\rootfs ➤ %SYSTEMROOT%\System32\bash.exe
<p>WSL registry hive</p> <p>Windows feature that enables user to run a different operating system as an application – registries and configurations</p>	NTUSER.DAT and COMPONENTS – Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Lxss ➤ HKEY_LOCAL_MACHINE\COMPONENTS\CanoicalData
<p>Virtual Box</p> <p>Application used to create and execute virtual machines</p>	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\VirtualBox\Machines<vm name>\Logs\vbox.log ➤ %USERPROFILE%\Appdata\Local\virtualbox ➤ %PROGRAMFILES%\Oracle\VirtualBox
<p>Virtual Machine file extensions</p> <p>Presence of virtual machine files or virtual hard drives</p>	Files	<ul style="list-style-type: none"> ➤ *.VMDK ➤ *.VDI ➤ *.VHD ➤ *.OVA

Table 14 - Forensic Artifacts - Virtual Machines

Artifact Description	Type	Location
<p>Terminal Server</p> <p>Application used for remote access – registry hive</p>	SYSTEM – Registry Hive	<ul style="list-style-type: none"> ➤ SYSTEM\ControlSet###\Control\Terminal Server\fdenyTSConnections
<p>TeamViewer logs</p> <p>Application used for remote access – execution logs</p>	Files	<ul style="list-style-type: none"> ➤ %PROGRAMFILES%\TeamViewer\Connections_incoming.txt
<p>LogMeIn/Hamachi logs</p> <p>Application used to remote access or create virtual networks</p>	Files	<ul style="list-style-type: none"> ➤ %PROGRAMFILES%\LogMeIn\ ➤ %SYSTEMROOT%\system32\config\systemprofile\AppData\Local\LogMeIn Hamachi\

Appendix I – Forensic artifact locations

<p>Windows Remote Desktop (RDP)</p> <p>Session details and cache of remote access with RDP (Microsoft Windows built in tool)</p>	Files	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Local\Microsoft\Terminal Server Client\Cache ➤ %SYSTEMROOT%\System32\winevt\logs\Security.evtx (Event ID 4779 – Session Disconnected and ID 4778 – Session Connected/Reconnected)
-----------------------------------------------------------------------------------------------------------------------------------------	-------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 15 - Forensic Artifacts - Remote Access Tools

Artifact Description	Type	Location
<p>CCleaner</p> <p>Application capable of deleting files/directories with wiping standard and general deletion of registry keys and logs</p>	Folder	➤ %PROGRAMFILES%\CCleaner
<p>Eraser</p> <p>Application capable of deleting files/directories with wiping standard</p>	Folder	➤ %PROGRAMFILES%\Eraser
<p>File Shredder</p> <p>Application capable of deleting files/directories with wiping standard</p>	Folder	➤ %PROGRAMFILES%\File Shredder

Table 16 - Forensic Artifacts - Wiping Tools

Artifact Description	Type	Location
<p>Microsoft BitLocker</p> <p>Indicator of presence of bitlocker encrypted drives</p>	NTUSER.DAT and SYSTEM – Registry Hive	<ul style="list-style-type: none"> ➤ SYSTEM\ControlSet001\services\fvevol\Enum (BitLocker Drive Encryption Driver Service) ➤ SYSTEM\ControlSet###\Control\FileSystem\NtfsEncryptPagingFile (Encrypted Page File) ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\FveAutoUnlock\ (BitLocker To Go)
<p>TrueCrypt / VeraCrypt</p> <p>Default installation folder of Veracrypt/Truecrypt</p>	Folder	<ul style="list-style-type: none"> ➤ %APPDATA%\VeraCrypt\ ➤ %APPDATA%\TrueCrypt\

Appendix I – Forensic artifact locations

TrueCrypt / Veracrypt Encrypted container file for Veracrypt/Truecrypt	File	➤ Files with extension “.tc” or “.hc”
-----------------------------------------------------------------------------------------------	------	---------------------------------------

Table 17 - Forensic Artifacts - Encryption Tools

Artifact Description	Type	Location
Internet Explorer to Clear Browser History on exit Registry key that indicates that Internet Explorer was configured to automatically delete browser history	NTUSER.DAT – Registry Hive	➤ NTUSER.DAT\Software\Microsoft\Internet Explorer\Privacy\ClearBrowserHistoryOnExit
Windows to Clear the Pagefile on Shutdown Registry key that indicates that the pagination file was disabled	SYSTEM – Registry Hive	➤ SYSTEM\ControlSet###\Control\Session Manager\Memory Management\ClearPageFileAtShutdown
Steganography tools (QuickStego, StegoSuite, OpenStego) Indication of presence of steganography tools	Folder	<ul style="list-style-type: none"> ➤ C:\Program Files (x86)\Quick Stego ➤ “stegosuite-0.7-win_amd64.jar” ➤ %AppData%\Sun\Java\Deployment\Cache (Windows XP) ➤ %AppData%\LocalLow\Sun\Java\Deployment\Cache (Windows Vista/7/8) ➤ C:\Program Files (x86)\OpenStego
Timestomp, BulkFileChanger, FileDate Changer, Attribute Changer (manipulate file metadata) Indication of presence of	Folder	<ul style="list-style-type: none"> ➤ %SYSTEMROOT%\Prefetch (search for file name) ➤ %PROGRAMFILES%\Attribute Changer

timestamp manipulation tools		
VPN (NordVPN, ProtonVPN, KeepSolid Unlimited VPN, ExpressVPN, OpenVPN, TOR) Indication of presence of TOR or common VPN applications	Folder	<ul style="list-style-type: none"> ➤ %PROGRAMFILES%\NordVPN ➤ %PROGRAMFILES%\Proton Technologies\ProtonVPN ➤ %PROGRAMFILES% \VPN Unlimited ➤ %APPDATA%\Roaming\ExpressVPN ➤ %PROGRAMFILES%\OpenVPN ➤ %USERPROFILE%\Desktop\Tor Browser\
Bootable ISO to USB Creator (Rufus, UNetbootin, Universal USB Installer) Indication of usage of virtual machines or bootable devices	Folder	<ul style="list-style-type: none"> ➤ %SYSTEMROOT%\Prefetch (search for file name)

Table 18 - Forensic Artifacts - Other Anti-forensic Indicators (Steganography, TOR, VPN, Metadata Changing, Bootable ISO and system configurations)

Artifact Description	Type	Location
Open/Save MRU Registry	SYSTEM – Registry Hive	<ul style="list-style-type: none"> ➤ HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced ➤ HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU
Recent Docs / MRU registry Most recently opened items registry	NTUSER.dat – Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU (on Windows XP) ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs (on Windows 7, 8 and 10)
Recent Recently opened items and applications	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\Recent\ (on Windows XP) ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ (on Windows 7, 8 and 10)

Appendix I – Forensic artifact locations

Table 19 - Forensic Artifacts - Recently Used Files

Artifact Description	Type	Location
Windows Jump Lists Recently opened items and applications	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations

Table 20 - Forensic Artifacts - Jump Lists

Artifact Description	Type	Location
Recent Recently opened files and applications	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\Recent\ (on Windows XP) ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ (on Windows 7, 8 and 10)
Recent Office Files Recently opened MS Office files (shortcuts on folder)	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\ (on Windows 7, 8 and 10)
Desktop Shortcuts to installed applications and loose files	Folder	<ul style="list-style-type: none"> ➤ %USERPROFILE%\Desktop*.lnk (user Desktop in any Windows version)
Start Menu Installed applications	Folder	<ul style="list-style-type: none"> ➤ %PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs

Table 21 - Forensic Artifacts - Link Files (LNK)

Artifact Description	Type	Location
Recent Office Documents Recently opened MS Office documents (registries on hives)	NTUSER.DAT – Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Office\Com mon\OpenFind\Microsoft Office\Word\Settings\Save As\File Name MRU\ ➤ NTUSER.DAT\Software\Microsoft\Office\Word \FileMRU ➤ NTUSER.DAT\Software\Microsoft\Office\Excel\ FileMRU ➤ NTUSER.DAT\Software\Microsoft\Office\ver#\ PowerPoint\FileMRU ➤ NTUSER.DAT\Software\Microsoft\Office\Acces s\FileMRU

Appendix I – Forensic artifact locations

		➤ NTUSER.DAT\Software\Microsoft\Office\VERSION\UserMRU\LiveID_####\FileMRU
Recent Office Files Recently opened MS Office files (shortcuts on folders)	Folder	➤ %USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\ (on Windows 7, 8 and 10)

Table 22 - Forensic Artifacts - Recent Office Documents

Artifact Description	Type	Location
Notepad++ Sessions Recently opened txt files on Notepad++	Files	➤ %USERPROFILE%\AppData\Roaming\Notepad++\session.xml
Notepad++ Opened files but not saved Recently opened txt files that were edited but not saved	Folder	➤ %USERPROFILE%\AppData\Roaming\Notepad++\backup\

Table 23 - Forensic Artifacts - Notepad++ Sessions

Artifact Description	Type	Location
Downloads Folder Downloaded files and applications	Folder	➤ %USERPROFILE%\Downloads
Firefox Downloads Database of visited websites and downloaded files/applications	SQLite database	<ul style="list-style-type: none"> ➤ %USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<randomtext>.default\downloads.sqlite (on Windows XP) ➤ %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\downloads.sqlite (on Windows 7, 8 or 10)
IE/Edge Database of visited websites and downloaded files/applications	SQLite database	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Roaming\Microsoft\Windows\IEDownloadHistory\ (IE8-9) ➤ %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat (IE 10/11/Edge)

Table 24- Forensic Artifacts - Downloads

Appendix I – Forensic artifact locations

Artifact Description	Type	Location
Print Spooler folder Temporary storage of files sent to printer	Folder	➤ %SYSTEMROOT%\system32\spool\printers
Print Spooler registries Registries of usage of printers	SYSTEM and SOFTWARE Registry Hives	<ul style="list-style-type: none"> ➤ SYSTEM\ControlSet###\Control\Print\Environments\WindowsNTx86\Drivers\Version#\ ➤ SYSTEM\ControlSet###\Services\LanmanServer\Shares\ ➤ SOFTWARE\Microsoft\Windows NT\CurrentVersion\Print\Printers\

Table 25 - Forensic Artifacts - Print Spooler

Artifact Description	Type	Location
Windows Prefetch Recently opened applications and how many times they were executed	Folder	➤ %SYSTEMROOT%\Prefetch

Table 26 - Forensic Artifacts - Prefetch

Artifact Description	Type	Location
IE/Edge user hives History of internet access and favorites	NTUSER.DAT and UsrClass.dat – Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Favorites\Order ➤ UsrClass.dat\LocalSettings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\MicrosoftEdge\FavOrder\
Edge History of internet access and favorites	EDB database	➤ %USERPROFILE%\AppData\Local\Packages\Microsoft.MicrosoftEdge_XXX\AC\MicrosoftEdge\User\Default\DataStore\Data\nouser1\XXX\DBStore\spartan.edb
Firefox History of internet access and downloads	SQLite database	➤ %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\[profileID].default\places.sqlite

Appendix I – Forensic artifact locations

<p>Google Chrome</p> <p>History of internet access and downloads</p>	<p>SQLite database</p>	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Bookmarks ➤ %USERPROFILE%\AppData\Local\Google\Chrome\User Data\ChromeDefaultData\Bookmarks
-----------------------------------------------------------------------------	------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 27 - Forensic Artifacts - IE/Edge Bookmarks

Artifact Description	Type	Location
<p>IE/Edge Registries</p> <p>History of internet access and typed URLs both on internet explorer and windows explorer</p>	<p>NTUSER.DAT and UsrClass.dat – Registry Hive</p>	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Internet Explorer\TypedURLs ➤ UsrClass.dat\LocalSettings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\MicrosoftEdge\TypedURLs
<p>IE/Edge Folders</p> <p>URLs accessed on IE/Edge or Windows Explorer on Windows 10</p>	<p>Folder</p>	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat ➤ %USERPROFILE%\Local Settings\History\History.IE5 (IE6-7) ➤ %USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5 (IE8-9)
<p>Google Chrome Registries</p> <p>URLs accessed on Chrome</p>	<p>NTUSER.DAT – Registry Hive</p>	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Google\NavClient\1.1\History
<p>Google Chrome databases</p> <p>URLs accessed on Chrome</p>	<p>SQLite database</p>	<ul style="list-style-type: none"> ➤ %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\History (Win XP) ➤ %USERPROFILE%\AppData\Local\Google\Chrome\User Data\ChromeDefaultData\History (Win 7/8/10)
<p>Mozilla Firefox databases</p> <p>URLs accessed on Firefox</p>	<p>SQLite database</p>	<ul style="list-style-type: none"> ➤ %USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite (Win XP) ➤ %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite\ (Win 7/8/10)

Table 28- Forensic Artifacts - IE/Edge Recent URLs and Searches

Artifact Description	Type	Location
----------------------	------	----------

Appendix I – Forensic artifact locations

Outlook PST default location Email messages on Outlook PST databases	Folder	<ul style="list-style-type: none"> ➤ %USERS%\AppData\Local\MicrosoftOutlook ➤ %USERPROFILE%\Local Settings\Application Data\Microsoft\Outlook\ ➤ %USERPROFILE%\Documents\Outlook Files
Outlook registry hives Registry of permissions and installation of MS Outlook	SOFTWARE – Registry Hive	<ul style="list-style-type: none"> ➤ HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook (Outlook 2016) ➤ HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Outlook (Outlook 2013) ➤ HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Outlook (Outlook 2010) ➤ HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Outlook (Outlook 2007)
IBM Notes NSF default location Email messages on Lotus Notes (NSF) databases	Folder	<ul style="list-style-type: none"> ➤ %PROGRAMFILES%\IBM\Lotus\Data ➤ %USERPROFILE%\Local Settings\Application Data\Lotus\Notes\Data (Windows XP) ➤ %USERS%\AppData\Local\Lotus\Notes\Data (Windows Vista) ➤ %USERPROFILE%\AppData\Local\Lotus\Notes\Data (Windows 7)

Table 29 - Forensic Artifacts – Mailboxes

Artifact Description	Type	Location
USB Registry files History of installed USB devices, may include mass storage devices, smartphones, cameras, etc.	SYSTEM and SOFTWARE – Registry Hive	<ul style="list-style-type: none"> ➤ SYSTEM\ControlSet###\Enum\USB\ ➤ SYSTEM\ControlSet###\Enum\USBSTOR\ ➤ SYSTEM\CurrentControlSet\Enum\USBSTOR ➤ SYSTEM\CurrentControlSet\Enum\USB ➤ SYSTEM\MountedDevices (Volume letter and name) ➤ SYSTEM\CurrentControlSet\Services\Disk\Enum ➤ SOFTWARE\Microsoft\Windows Portable Devices\Devices ➤ SOFTWARE\Microsoft\WindowsNT\CurrentVersion\EMDMgmt
USB Mount Points Mounted storage devices	NTUSER.DAT – Registry Hive	<ul style="list-style-type: none"> ➤ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\

via USB on some Windows versions		
<p>Plug and Play log files</p> <p>Logs of plug and play actions, may include mass storage devices, smartphones, cameras, etc.</p>	File	<ul style="list-style-type: none"> ➤ %SYSTEMROOT%\setupapi.log (Plug and Play Log files on Windows XP) ➤ %SYSTEMROOT%\inf\setupapi.dev.log (Plug and Play Log files on Windows 7 and ahead)

Table 30 - Forensic Artifacts - USB Devices List

Artifact Description	Type	Location
<p>Pagination</p> <p>Temporary allocation of RAM on local storage (when amount of RAM is not sufficient), may include passwords, open applications, chats, open web pages</p>	Memory file	➤ C:\pagefile.sys
<p>Hibernation</p> <p>Temporary allocation of RAM on local storage for quick return after shutdown, may include passwords, open applications, chats, open web pages</p>	Memory file	➤ C:\Hiberfil.sys
<p>Swapfile</p> <p>Similar to pagination, but only stores priority connections.</p>	Memory file	➤ C:\swapfile.sys
<p>MemDump</p> <p>Temporary allocation of RAM on local storage (when a fatal error occurs), may include passwords, open applications, chats, open web pages</p>	Memory file	➤ %SYSTEMROOT%\MEMORY.DMP

Appendix I – Forensic artifact locations

MiniDump Similar to MemDump, but only stores small data related to error cause.	Memory file	➤ % SYSTEMROOT%\Minidump
-------------------------------------------------------------------------------------------	-------------	--------------------------

Table 31 - Forensic Artifacts - Memory Dumps

Appendix II – Tabulation of execution times

In this section the specific times for each test executed on the analysis topic are presented in the table below.

Scenario	Evidence	Collection Mode	General Start Time	General End Time	Elapsed time
[1] (USB SSD to SATA SSD)	E001	ECA Only	30/10/2021 20:04	30/10/2021 20:13	00:08:52
[1] (USB SSD to SATA SSD)	E002	Full Disk + Verify	30/10/2021 19:27	30/10/2021 19:44	00:16:54
[1] (USB SSD to SATA SSD)	E003	Full Disk + Verify + ECA	30/10/2021 20:14	30/10/2021 20:33	00:19:03
[2] (USB SSD to USB HDD)	E004	ECA Only	30/10/2021 20:44	30/10/2021 20:53	00:09:15
[2] (USB SSD to USB HDD)	E005	Full Disk + Verify	30/10/2021 20:56	30/10/2021 21:12	00:16:24
[2] (USB SSD to USB HDD)	E006	Full Disk + Verify + ECA	30/10/2021 21:14	30/10/2021 21:33	00:18:16
[3] (SATA SSD to USB HDD)	E007	ECA Only	30/10/2021 21:43	30/10/2021 21:51	00:08:44
[3] (SATA SSD to USB HDD)	E008	Full Disk + Verify	30/10/2021 21:56	30/10/2021 22:11	00:15:08
[3] (SATA SSD to USB HDD)	E009	Full Disk + Verify + ECA	30/10/2021 22:13	30/10/2021 22:30	00:17:43
[4] (SATA SSD to USB SSD)	E010	ECA Only	30/10/2021 20:04	30/10/2021 20:13	00:08:52
[4] (SATA SSD to USB SSD)	E011	Full Disk + Verify	30/10/2021 19:27	30/10/2021 19:44	00:16:54
[4] (SATA SSD to USB SSD)	E012	Full Disk + Verify + ECA	30/10/2021 20:14	30/10/2021 20:33	00:19:03

Table 32 - Execution Times

Appendix III – Description of Hardware utilized on tests

Details of hardware utilized on these tests and respective pictures.

- **Laptop**

- Model: Asus ROG GL553VD
- Description: Laptop with CPU Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz (4 physical cores, 8 logical cores), 32 Gb RAM DDR4 @ 2.40 GHz and two ports USB 3.0.
- Website with more details:
https://www.asus.com/supportonly/GL553VD/HelpDesk_Manual/



Figure 21 - Laptop Asus model details

- **Source SSD**

- Model: SanDisk SSD PLUS 120GB (SATA III)
- Description: SSD 2.5" SATA III (6 Gbit/s). With theoretical read speeds of up to 530 MB/s and write speeds of up to 400 MB/s.
- Website with more details:
https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/sandisk/product/internal-drives/ssd-plus-sata-iii-ssd/data-sheet-ssd-plus-sata-iii-ssd.pdf

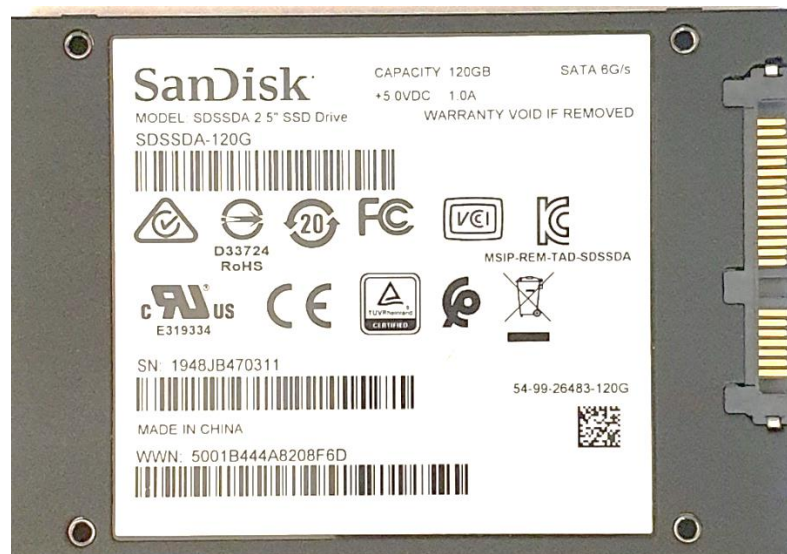


Figure 22 - SSD Source model details

- **Destination SSD**

- Model: Samsung SSD 850 PRO 1TB (SATA III)
- Description: SSD 2.5" SATA III (6 Gbit/s). With theoretical read speeds of up to 550 MB/s and write speeds of up to 520 MB/s.
- Website with more details:
<https://www.samsung.com/us/computing/memory-storage/solid-state-drives/ssd-850-pro-2-5-sata-iii-1tb-mz-7ke1tbw/>



Figure 23 - SSD Destination model details

- **Destination HDD**

- Model: Seagate Backup Plus Portable Drive 2TB USB 3.0 (SATA III)
- Description: External Hard Drive 2.5" USB 3.0 SATA III (6 Gbit/s). With theoretical read speeds of up to 90 MB/s and write speeds of up to 120 MB/s.
- Website with more details: <https://www.seagate.com/support/external-hard-drives/portable-hard-drives/backup-plus/#specs>



Figure 24- HDD Destination model details

Appendix III – Description of Hardware utilized on tests

- **Tableau Enclosure**

- Model: Tableau Forensic SATA/IDE Bridge T35u
- Description: “The Tableau Forensic SATA/IDE Bridge is a portable write-blocker that enables forensic acquisition of SATA and IDE solid-state-drives. [...] Features USB 3.0 host computer connection”.
- Website with more details:
<https://security.opentext.com/tableau/hardware/details/t35u>



Figure 25 - Tableau T35u Bridge

- **JMicron Enclosure**

- Model: JMicron JMS567 SATA/USB 3.0 Adapter
- Description: Is a Super Speed USB to Dual SATA 6Gbps ports bridge enclosure.
- Website with more details: <https://www.jmicron.com/products/list/12>

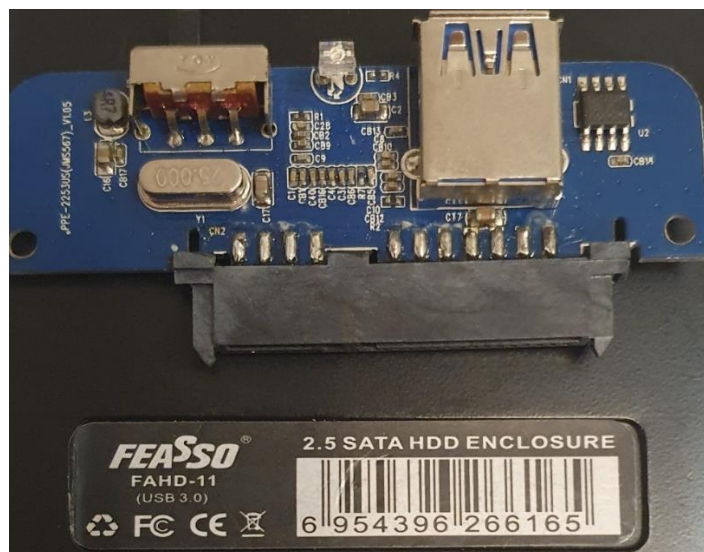


Figure 26 - JMicron SATA to USB3.0 enclosure