DIGITAL FORENSICS FUNDAMENTALS

Co-funded by the Erasmus+ Programme of the European Union

TABLE OF CONTENT

1. Introduction to Digital forensics

Digital forensics is directly related to incident response, being even one of its most important components, being necessary to obtain information about events and actions performed in the system under analysis. Digital forensics is also criminally decisive, investigating past actions on the devices in an attempt to identify and collect an evidence of crime, which could help the decision in court.

"it is impossible for a criminal to act, especially considering the intensity of a crime, without leaving traces of this presence" (Doctor Edmond Locard, s.d.)[1]

Digital forensics *Is understood as a branch of forensic science that studies and applies the process of acquiring, analyzing and preserving digital evidence so that it is* legally admissible and technically irrefutable in court.

Digital Proof is any data or digital information, legally admissible (obtaining) and technically irrefutable (origin, integrity and non-repudiation).

Digital Information is all data stored or transmitted digitally, such as logs, documents, emails, database, network traffic, among many others.

The goal of digital forensics and incident response:

· Identify, collect and preserve a evidence of a cybercrime;

· Interpret, document and present evidences in such a way that it is admissible in court;

· Understand the techniques and methods used by criminals;

· Respond to incidents to prevent intellectual, financial, and reputational property losses during na attack;

· Know the legislation of various regions;

· Learn about digital platform manipulation processes, data types, and operating systems;

· Identify the appropriate tools for forensic investigation;

· Recover deleted files, hidden files, and temporary data that can be used as evidence;

· Support prosecution in cybercrime investigation;

· Protect the organisation from similar incidents in the future.

---

[1] https://www.crimemuseum.org/crime-library/forensic-investigation/edmond-locard

1.1. Responsibilities of an Expert

A forensic analyst as well as the computer expert is legally responsible for analysing the digital content of the equipment that will be entrusted to him/her, delivering a report called the digital Forensic Expertise Report. After being made aware of the penalties for refusal or non-compliance, read the following sentence before the judicial authority or magistrate, “I undertake, on my honour, to faithfully fulfil the duties entrusted to me” and sign the statement of undertaking. In this way, the expert undertakes to carry out the analysis and digital forensic report, the experts who are public servants and intervene in the performance of their duties being exempt from the term of commitment. Will be informed about the process and about the questions to be answered. The penalties for refusal or non-compliance are present in the Portuguese Code of Criminal Procedure in Article 91, no. 4. Regarding the delivery of the expert report, Article 157 of the same Code of Criminal Procedure states that the report must contain the answers and conclusions duly reasoned, and must be presented within a period not exceeding 60 days.

It is imperative that the report presents only true information, as incorrect information is punishable by imprisonment from six months to three years or a fine of not less than 60 days, as stated in Art. 360.

1.2. Digital Forensics Application

The areas of activity of digital forensics are increasingly comprehensive, where before it would only be analysed the hard disks and other media for storing information, now it may be necessary to collect and analyse data in volatile memories (performance in live-data forensics) or even network traffic data (network forensics), data stored on mobile devices (mobile forensics), data stored in distributed systems in Clouds on the Internet, among many others.

Edmond Locard (Figure 1), stated that:

"it is impossible for a criminal to act, especially considering the intensity of a crime, without leaving traces of this presence"

In crimes that somehow involve a digital component, cyberinstrumental and cyberdependent crime (R.Bravo) and according to Edmong Locard's statement, there is a greater possibility of digital evidence being left, relating the crime to the use of digital means.

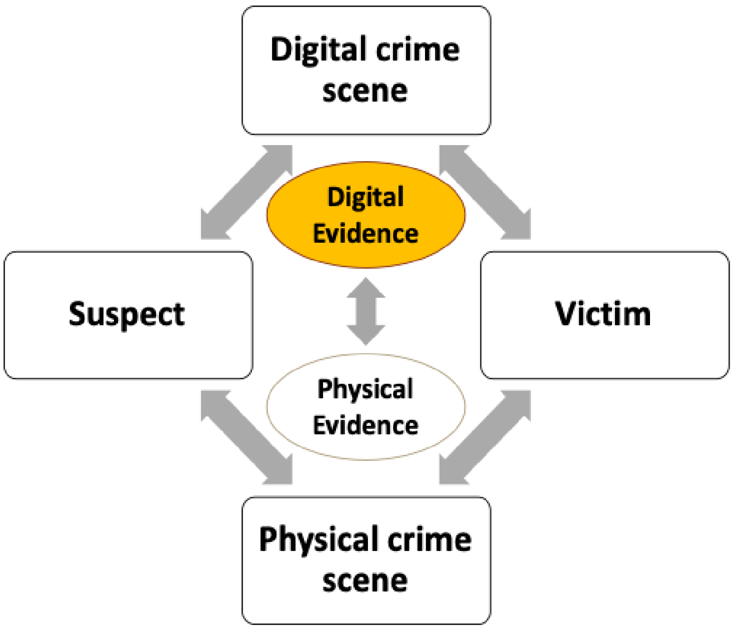


Figure 1 - Edmond Locard's scheme

Source: https://www.crimemuseum.org/crime-library/forensic-investigation/edmond-locard

Digital forensics is not considered an exact science, and it is possible that the same digital forensics report analysed by diferent people may have diferent understandings, hence the concern to ensure the basic principles of information security (Figure 2), confidentiality, integrity and non-repudiation, are essential. In this way digital forensics will necessarily be:

LEGALLY ADMISSIBLE AND TECHNICALLY IRREFUTIBLE

The use of universally accepted techniques, complying with the provisions of national laws in an attempt to give each question under investigation as complete an answer as possible, the 7 whys being the most complete form of response:

"What? Where?, When?, How?, Who?, Why? and How Many?"

Inspector-coordinator Rogério Bravo – Judicial Police

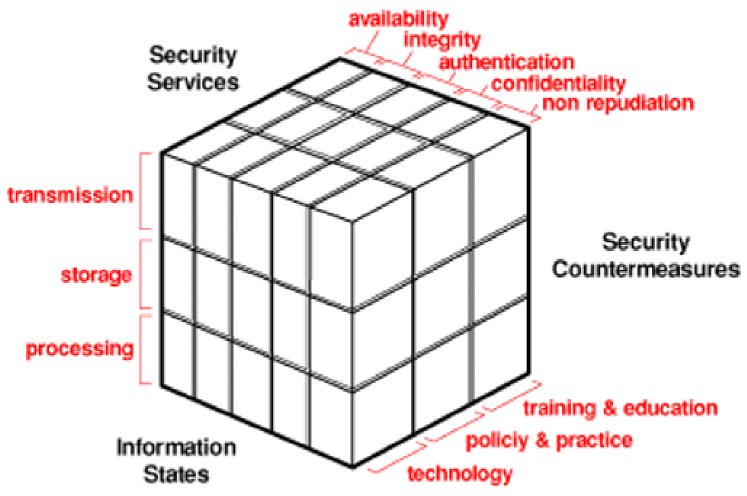


Figure 2 – Information Security

Source: John McCumber

Confidentiality

Until the court decides, the people involved are and must remain innocent, and access to their data is totally restricted to the expert himself, who cannot make it available to third parties.

Authenticity

The evidence must be authentic. Produced by people who can answer about it. Otherwise the evidence is considered irrelevant in court.

Integrity

The information contained in the devices must be maintained in its original state at all costs. The expert is responsible for the use of techniques that alter the integrity of the information.

No Repudiation

As far as the exercise of expert functions is concerned, “non-repudiation” suits the use of universally accepted techniques, making it possible to use counter-expertise to obtain the same results.

1.3. Challenges of Digital forensics

Digital forensic methods and techniques face great challenges at present, forcing the forensic investigator into a continuous need for research and improvement. Traditionally, digital forensic investigators have systematically sought a much closer look at artefacts in search of a possible clue that may be evidence of a crime. But as technologies evolve, the procedures and approaches to finding this crime evidence need to be improved and adapted. There are lots of new small challenges for digital forensics to overcome, however we present a short summary in 3 categories:

Technicals:

- Different types of storage;
- Encryption;
- Stegnography;
- Antiforensic Techniques;
- Acquisition and Analysis in Live Data Forensics;
- Concealment and Disposal of data;
- Data volatility;
- Network shares;
- ...

Legals:

- Privacy;
- Data Protection;
- Delay in adapting laws to the technological reality;
- Acting at the scene of the crime;
- Data analysis and handling;
- ...

Resources:

- The increase in the volume of data;
- Complexity of distributed data storage systems;
- ...

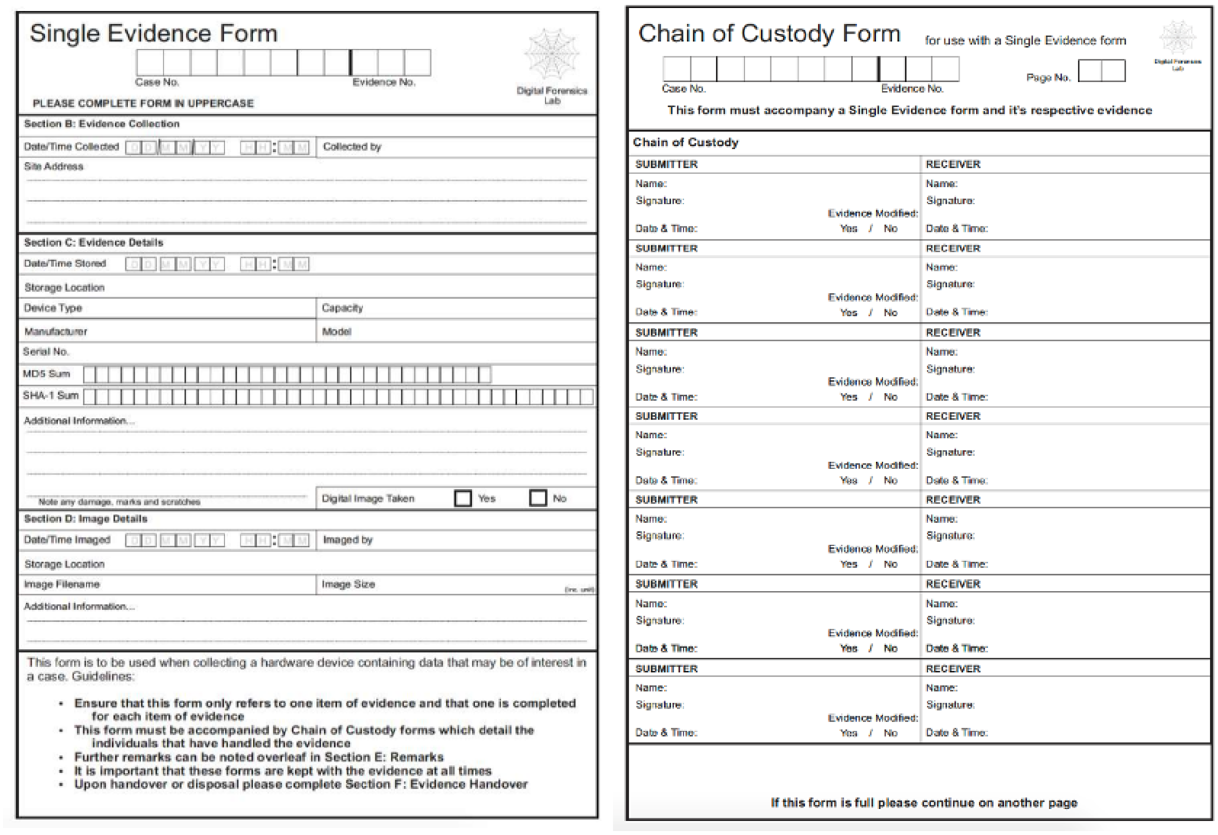
1.4. International Standards

The entities that seek to develop good practice guides in the area of digital forensics are numerous, bu our references are ISO/IEC, NIST, ENISA, SANS and others who seek to disseminate and develop knowledge in the area. Thus, there are some important documents in the development of the functions of the digital forensics expert, namely:

- **RFC 3227:2002** Guide to the acquisition and preservation of digital evidence
- **NIST 800-86** Guide to the integration of forensic techniques in incident response
- **NIST 800-144** Guide to Security and Privacy in the Cloud
- **NIST 800-101** Guide to mobile device forensics
- **ISO/IEC 20000-1:2018** Information technology - Service management
- **ISO/IEC 27001:2013** Definition of an ISMS (Information Security Management System)
- **ISO/IEC 27002:2013** Guide to good practices in information security
- **ISO/IEC 27005:2018** Information security risk management
- **ISO/IEC 27032:2012** Guide to cybersecurity
- **ISO/IEC 27037:2012** Guide to the identification, collection, acquisition and preservation of digital evidence

1.5. Chain of custody

The chain of custody is a document/form that must be filled out by the person who seizes the equipment. This document must always accompany the equipment itself, keeping a dated record of all persons who had custody of the equipment. An example of the chain of custody is presented (Figure 3 – Digital Forensics Lab).



Single Evidence Form

Case No. Evidence No.

Digital Forensics Lab

PLEASE COMPLETE FORM IN UPPERCASE

Section B: Evidence Collection	
Date/Time Collected D D M M Y Y H H : M M	Collected by
Site Address	
Section C: Evidence Details	
Date/Time Stored D D M M Y Y H H : M M	
Storage Location	
Device Type	Capacity
Manufacturer	Model
Serial No.	
MD5 Sum	
SHA-1 Sum	
Additional Information...	
Note any damage, marks and scratches	Digital Image Taken ☐ Yes ☐ No
Section D: Image Details	
Date/Time Imaged D D M M Y Y H H : M M	Imaged by
Storage Location	
Image Filename	Image Size (inc. unit)
Additional Information...	

This form is to be used when collecting a hardware device containing data that may be of interest in a case. Guidelines:

- **Ensure that this form only refers to one item of evidence and that one is completed for each item of evidence**
- **This form must be accompanied by Chain of Custody forms which detail the individuals that have handled the evidence**
- **Further remarks can be noted overleaf in Section E: Remarks**
- **It is important that these forms are kept with the evidence at all times**
- **Upon handover or disposal please complete Section F: Evidence Handover**

Chain of Custody Form for use with a Single Evidence form

Digital Forensics Lab

Case No. Evidence No. Page No.

This form must accompany a Single Evidence form and it's respective evidence

Chain of Custody

SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:
SUBMITTER		**RECEIVER**
Name:		Name:
Signature:	Evidence Modified:	Signature:
Date & Time:	Yes / No	Date & Time:

If this form is full please continue on another page

Figure 3 – Chain of custody Form

Source: https://www.dfir.training/index.php?option=com_jreviews&format=ajax&url=media/download&m=wx99T&1661384494937

1.6. Process models in forensic analysis

Each forensic investigator has their own working method and methodology during the course of a forensic analysis, and there is no standard model for each type of investigation, which is usually guided by the past experience of each investigator.

Over the time, various methodologies that define the need for a sequence of generic steps in a forensic investigation have emerged, usually defined as "evidence collection, preservation or examination, analysis".

Several investigation models have been proposed, also called "Digital Forensics Investigation Frameworks"(Figure 4), these being some of the most popular:

· DFRWS model - Digital Forensic Research Workshop (Palmer et al. 2001)

· ADFM - Abstract digital forensics model (Reith et al. 2002)

· IDIP - Integrated Digital Investigation Process (Carrier et al. 2003)

· EIDIP - Enhanced Integrated Digital Investigation Process (Baryamureeba & Tushabe 2004)

· CFFTPM - Computer Forensics Field Triage Process Model (Rogers et al. 2006)

· SRDFIM - Systematic Digital Forensic Investigation Model (Agarwal et al. 2011)

· IDFPM - Integrated Digital Forensic Process Model (Kohn et al. 2013)

· EDRM - Electronic Discovery Reference Model (https://edrm.net, 2014)

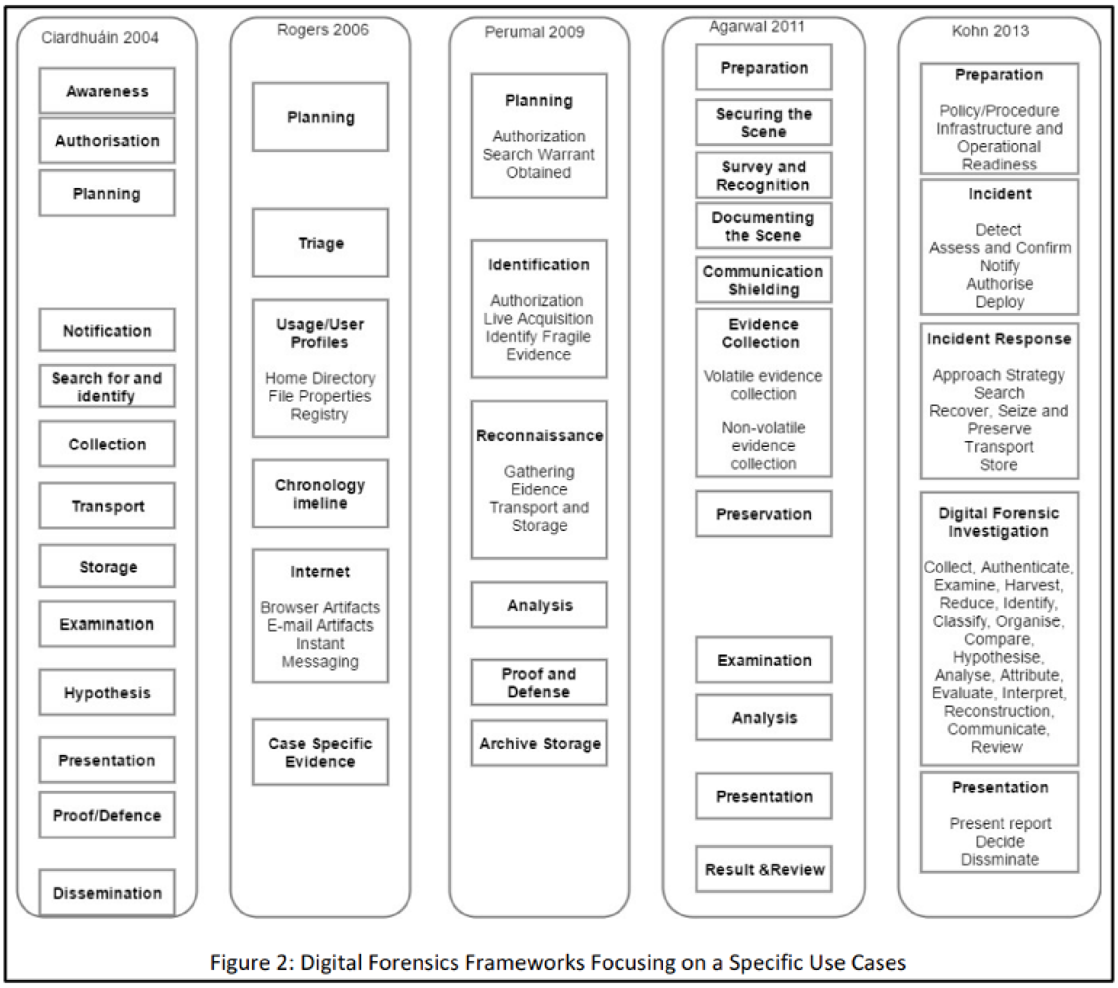

Ciardhuáin 2004
Awareness
Authorisation
Planning
Notification
Search for and identify
Collection
Transport
Storage
Examination
Hypothesis
Presentation
Proof/Defence
Dissemination
Rogers 2006
Planning
Triage
Usage/User Profiles
Home Directory File Properties Registry
Chronology imeline
Internet
Browser Artifacts E-mail Artifacts Instant Messaging
Case Specific Evidence
Perumal 2009
Planning
Authorization Search Warrant Obtained
Identification
Authorization Live Acquisition Identify Fragile Evidence
Reconnaissance
Gathering Eidence Transport and Storage
Analysis
Proof and Defense
Archive Storage
Agarwal 2011
Preparation
Securing the Scene
Survey and Recognition
Documenting the Scene
Communication Shielding
Evidence Collection
Volatile evidence collection
Non-volatile evidence collection
Preservation
Examination
Analysis
Presentation
Result &Review
Kohn 2013
Preparation
Policy/Procedure Infrastructure and Operational Readiness
Incident
Detect Assess and Confirm Notify Authorise Deploy
Incident Response
Approach Strategy Search Recover, Seize and Preserve Transport Store
Digital Forensic Investigation
Collect, Authenticate, Examine, Harvest, Reduce, Identify, Classify, Organise, Compare, Hypothesise, Analyse, Attribute, Evaluate, Interpret, Reconstruction, Communicate, Review
Presentation
Present report Decide Dissminate
Figure 2: Digital Forensics Frameworks Focusing on a Specific Use Cases

Figure 4 - Digital Forensics Investigation Frameworks

Source: Figure 2 of the article https://arxiv.org/ftp/arxiv/papers/1708/1708.01730.pdf

In 2001, a research arising from the Digital Research Workshops proposed a 6-step framework, as presented in Figure 5.

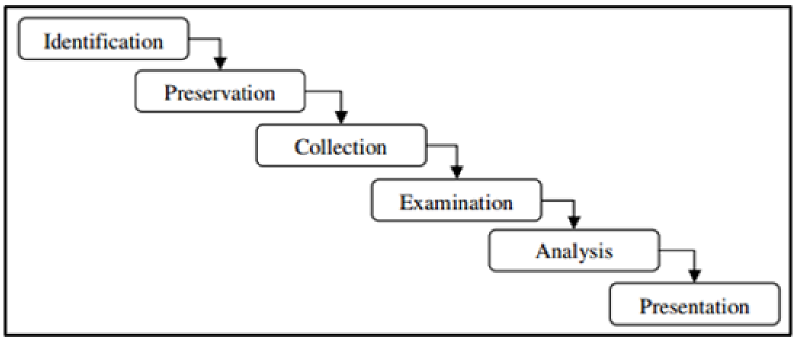


Figure 5 - DFRWS Framework

This is still today one of the main digital forensic investigation methodologies, and the one we will follow. Each of the stages is described below:

Identification – When the researcher needs to identify all relevant information and define the strategy for acquiring that information. The researcher may be dealing with a typical storage device, such as a hard drive, a memory card, or else digital data may need to be collected from network traffic data, volatile data such as memory data, mobile or IoT devices, or any other digital data storage device. At this stage, preparation prior to the use of techniques and tools, is extremely important to ensure the authenticity, integrity and non-repudiation of all evidence in court.

Preservation – This is the stage that seeks to involve tasks such as setting up proper case management and ensuring anacceptable chain of custody in court. This stage is crucial to ensure that data is collected without any external contamination and analysed correctly.

Collection – This step refers to the acquisition of digital evidence, and traditionally this can be done through cloning or forensic imaging of the storage device. The acquisition of volatile data, or other relevant and volatile data, could be decisive for the investigation phase, especially when the data relating to the acquired storage is encrypted. The data collected at this stage is the input data or the source of data for analysis stage.

Examination – This is the phase of searching for the desired data, involving search techniques, recovery of deleted data, data decryption, password cracking, malware analysis, pattern analysis, among others. This phase is interconnected with the analysis phase, since, for example, after the identification of documents it will be necessary to analyse them, taking into account the answer to the requested questions.

Analysis – Analysis of all the data collected. This is the most time-consuming phase, due to the need to do a thorough search and identify all relevant artefacts. It is common in most cases that the data collected comes in the form of unstructured data, requiring specific tools and more time-consuming analysis to idenfitfy potential digital evidence data, where structured data is included, such as records, databases, data files, system files, web pages and others.

Presentation – This is the last phase of the digital forensic analysis process, where a final report with all relevant data to be submitted to the judge. This report should be submitted in hard copy, with all the artefacts considered important. In case of doubts about the information presented in the submitted report, the expert’s testimony in court is required to provide the respective clarifications .

2. Preservation and collection of digital evidence at the crime/incident scene

For the analysis to be performed in the best condition, it will be necessary that the preservation and collection is also performed correctly. In this section we will approach actions at the incident scene in the same way as action at the crime scene, since there are obvious similarities in both, but each of them will necessarily have their specificities and specific characteristics that we will not portray here.

INCIDENT

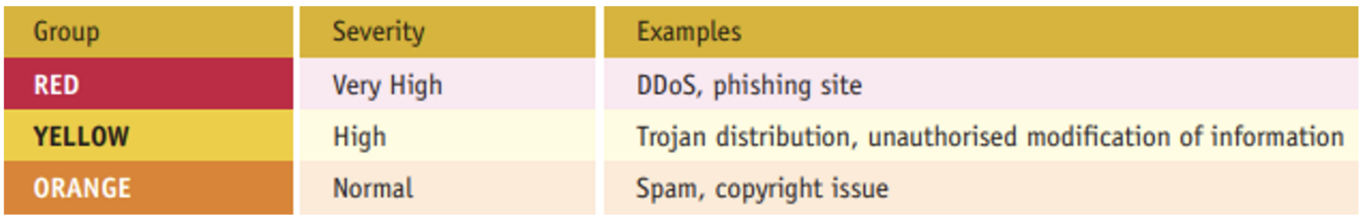
“Computer incident is an interruption or quality breakdown in an Information Technology Service” Source: Manual “ITIL V3 – Service Operation” (OGC, 2007)

For there to be an incident, there must necessarily be a compromise of the availability, authenticity, integrity and/or confidentiality of the data.

It is the Computer Security Incident Response Team (CSIRT) networks that make it possible to collect data regarding computer incidents, for this they have developed a common taxonomy (Source: https://www.redecsirt.pt/files/RNCSIRT_Taxonomia_v3.0.pdf) of incident classification, classifying them through 2 vectors, by type of incident and by type of event.

ENISA has als developed and promoted knowledge on good practices in incident identification and management, publishing regularly on the subject (Read: https://www.enisa.europa.eu/publications/using-taxonomies-in-incident-prevention-detection).

ENISA in 2010 published the document “Incident Management Guide“ (https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management) where it classifies incidents into categories according to their degree of severity, as presented in Figure 6.



Group	Severity	Examples
RED	Very High	DDoS, phishing site
YELLOW	High	Trojan distribution, unauthorised modification of information
ORANGE	Normal	Spam, copyright issue

Figure 6 – Incident classification

Source: Incident Management Guide (ENISA 2010)

2.1. International Incident Response Standards

The entities that seek to develop good practice guides in the area of incident response are numerous, however our references are ISSO/IEC, NIST and ENISA. Thus, there are some important documents in the development of the functions of the digital forensics expert, namely:

- **Incident Management Guide** (ENISA 2010)
- **ISO/IEC 27035:2016** Information security incident management guide for medium and large organisations
- **ISO/IEC 27037:2012** Guide to the identification, collection, acquisition and preservation of digital evidence
- **NIST 800-86** Guide to the integration of forensic techniques in incident response
- **NIST IR 8796** Security Analysis of First Responder mobile devices and wearables
- **ISO/IEC 27001:2013** Definition of an ISMS (Information Security Management System)
- **ISO/IEC 27002:2013** Guide to good practices in information security
- **ISO/IEC 27005:2018** Information security risk management
- **ISO/IEC 27032:2012** Guide to cybersecurity

ISO/IEC 27002 – Information security incident management defines the difference between **Event** and **Incident**, where an event may not always lead to anincident, but anincident always leads to anevent.

2.2. Incident management and mitigation

ISO/IEC 27035 defines 5 steps in incident management and mitigation, namely:

1. Preparation and Planning
2. Detection and recording
3. Assessment and decision
4. Response
5. Lessons learned

1.Preparation and Planning is the phase of identifying all the critical assets of the institution, internal processes for access to information, creation of monitoring systems that allow the identification of incidents, as well as all the responsibilities and procedures in case of anincident.

Preparation

· Preparation of the digital lab

· Defining the team leader

· Defining the team members and responsibilities

· Preparing the briefing / intervention strategy

Briefing

· Intervention strategy?

· Equipment needed to take to the incident site?

· What type of collection/acquisition methods (tools)?

· What is the network activity?

· Volatility of the data collected?

· Could the equipment have been configured to destroy evidence?

· How will we store/transport the digital evidence?

· Related equipment, manuals, etc ?

· Identify possible conflicts of interest?

· Risk assessment

2. Detection and recording is necessarily the phase of identifying events and distinguishing between the event or consecutive events and the possible incident .

Similar to any other crime scene, since the incident may have been intentionally provoked by an internal employee of the organisation. In this way we should take into consideration the previous preparation to act according to the following points:

· Securing the crime scene

· Collecting preliminar information

· Documenting the crime scene

· Collecting and preserving evidence

· Packaging and transporting

· Chain of custody

The maximum possible **collection of information**, appropriate to the type of possible incident, should be carried out in order to enable efficient decision-making. In this way we should take into consideration all types of information that can be acquired, such as:

· Connection type (Wi-Fi/Ethernet)

- Which computers are used to connect to the internet?
- Location of systems and identify whohas access
- Details of removable devices and user properties
- Obtain details of network topology
- Obtain details of other peripherals connected to the computer
- Are there any other questions on this subject that were not answered?

Of this information, we should take into consideration the surrounding information, such as:

- What services are offered by the organisation?
- Who are those affected by the incidents? Have they been informed?
- Are there logical protection measures (antivirus, firewall, IDS, IPS)? Alarms?
- What physical security measures are in place?
- Are there CCTV records
- Identify the number of computers and computers connected to the internet
- Check the latest hardware replacements
- Level of employee access? Recent redundacies?
- Administrative/Administrator access levels?
- Organisation's Security Policies?
- Procedures given to contain the incident?
- List of suspects? Why are they suspects?
- System Logs? Network logs? Anything suspicious?
- System usage after the incident? CMD/Shell commands? Scripts? Tasks? Processes?
- Post-incident analysis procedures?

The incident response team should take into consideration the collection of volatile data, which is possibly critical for quick decision-making regarding all past events. Thus, it is important to act in different ways according to whether the computer is on or off (Figure 7).

How to find the computer?

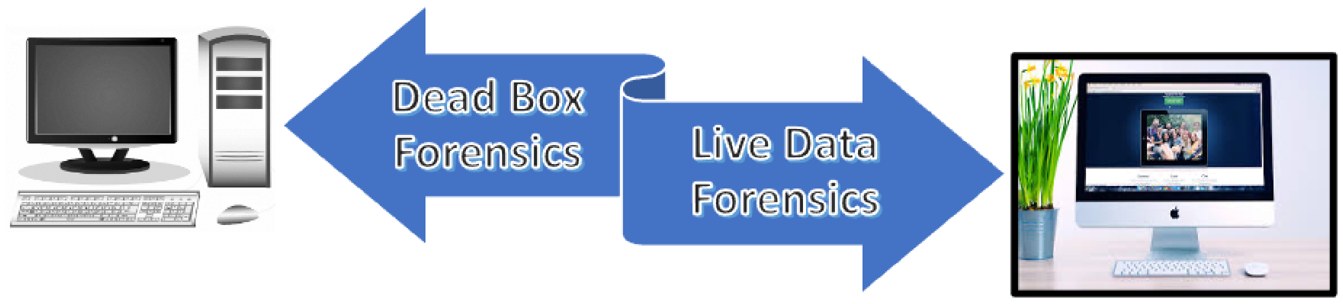
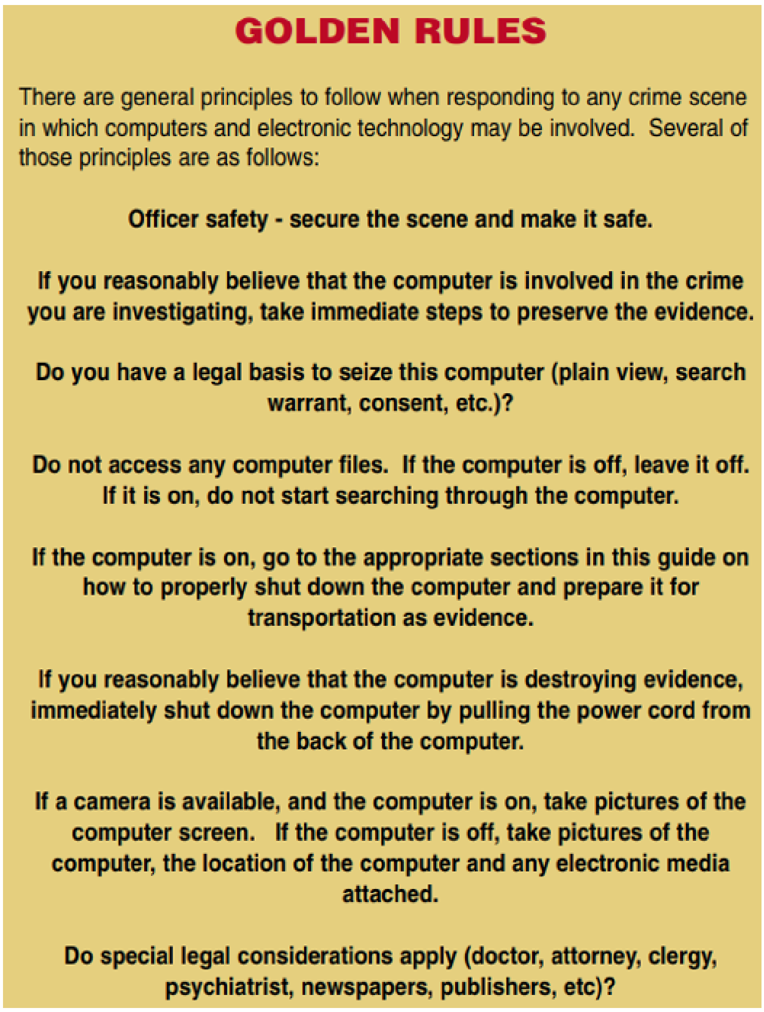


Figure 7 – first action

The first responder must have proper authority and expertise to act

The U.S. Department of Homeland Security has published a short guide for First Responders, entitled "Best Practices for seizing Electronic Evidence", which has what they call golden rules, which is presented in Figure 8.



GOLDEN RULES

There are general principles to follow when responding to any crime scene in which computers and electronic technology may be involved. Several of those principles are as follows:

Officer safety - secure the scene and make it safe.

If you reasonably believe that the computer is involved in the crime you are investigating, take immediate steps to preserve the evidence.

Do you have a legal basis to seize this computer (plain view, search warrant, consent, etc.)?

Do not access any computer files. If the computer is off, leave it off. If it is on, do not start searching through the computer.

If the computer is on, go to the appropriate sections in this guide on how to properly shut down the computer and prepare it for transportation as evidence.

If you reasonably believe that the computer is destroying evidence, immediately shut down the computer by pulling the power cord from the back of the computer.

If a camera is available, and the computer is on, take pictures of the computer screen. If the computer is off, take pictures of the computer, the location of the computer and any electronic media attached.

Do special legal considerations apply (doctor, attorney, clergy, psychiatrist, newspapers, publishers, etc)?

Figure 8 - Best Practices for seizing Electronic Evidence

Source: https://www.crime-scene-investigator.net/SeizingElectronicEvidence.pdf

These rules are currently in force and should be followed when responding to security incidents.

The U.S. Department of Justice – National Institute of Justice, on the other hand, has published a flow chart summarizing the process of approaching devices, called "Collecting Digital Evidence Flow Chart" (Figure 9).

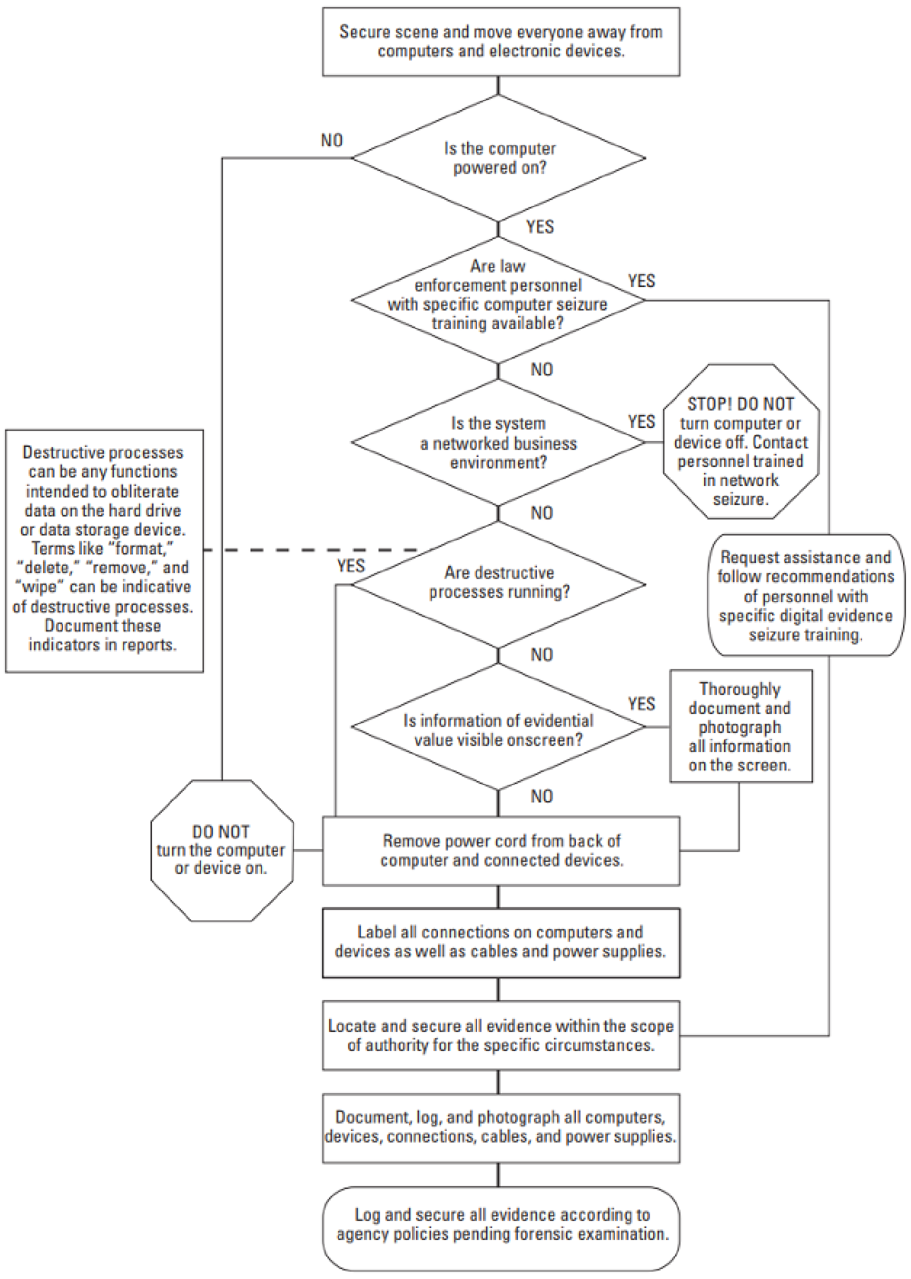

Secure scene and move everyone away from computers and electronic devices.
Is the computer powered on?
NO
YES
Are law enforcement personnel with specific computer seizure training available?
YES
NO
Is the system a networked business environment?
YES
STOP! DO NOT turn computer or device off. Contact personnel trained in network seizure.
NO
Destructive processes can be any functions intended to obliterate data on the hard drive or data storage device. Terms like "format," "delete," "remove," and "wipe" can be indicative of destructive processes. Document these indicators in reports.
Are destructive processes running?
YES
Request assistance and follow recommendations of personnel with specific digital evidence seizure training.
NO
Is information of evidential value visible onscreen?
YES
Thoroughly document and photograph all information on the screen.
NO
DO NOT turn the computer or device on.
Remove power cord from back of computer and connected devices.
Label all connections on computers and devices as well as cables and power supplies.
Locate and secure all evidence within the scope of authority for the specific circumstances.
Document, log, and photograph all computers, devices, connections, cables, and power supplies.
Log and secure all evidence according to agency policies pending forensic examination.

Figure 9 - Collecting Digital Evidence Flow Chart

Source: Collecting Digital Evidence Flow Chart. US Department of Justice - National Institute of Justice (2010).

In this regard, we can determine the following procedures:

1. Ensure physical and electronic security.

2. If the computer is turned off:

· Ensure that it does not turn on (Disconnect power cord/ Remove battery, if any)

· Label/photograph all components and peripherals

· Identify storage devices

· Check BIOS data/time (Without hard disk)

3. If the computer is turned on:

· Disconnect communications (disconnect Network cable/ remove SIM Card)

· Photograph the screen and all its contentes (describe the visible content)

- If there is a need to collect volatile and non-volatile data:
- Perform Live acquisition of the data (according to the order of volatility)
- Check that storage devices are encrypted
- Disconnect the computer from the power supply
- Labelling/photographing all components and peripherals
- Identify the storage devices
- Check BIOS date/time (Without hard disk)

4. If a mobile device (smartphone/Tablet) is envolved:

- If the device is switched off, do not switch on
- If the device is on :
- Photograph the display (if available)
- Pu tinto flight mode
- Always ensure that the device connected to the battery has sufficient power supply to keep it active until analysis
- Use a Faraday bag/sack (Figure 10)

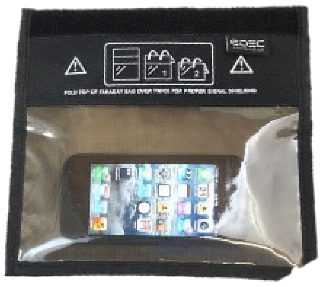


Figure 10 – Faraday bag

Label/photograph all components

- Collect all additional storage devices (memory cards, SIM cards, etc.)
- Document all steps involved in the mobile device seizure
- Question for access codes (PIN/Pattern), SIM card PIN and PUK codes (check device carrying pouches – Figure 11)

How to store and transport?

- Wear gloves

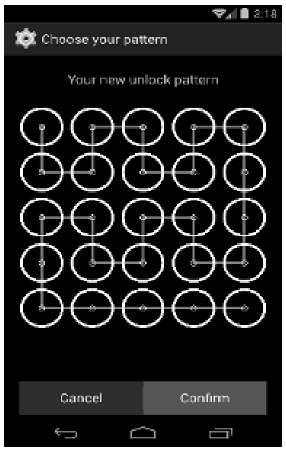
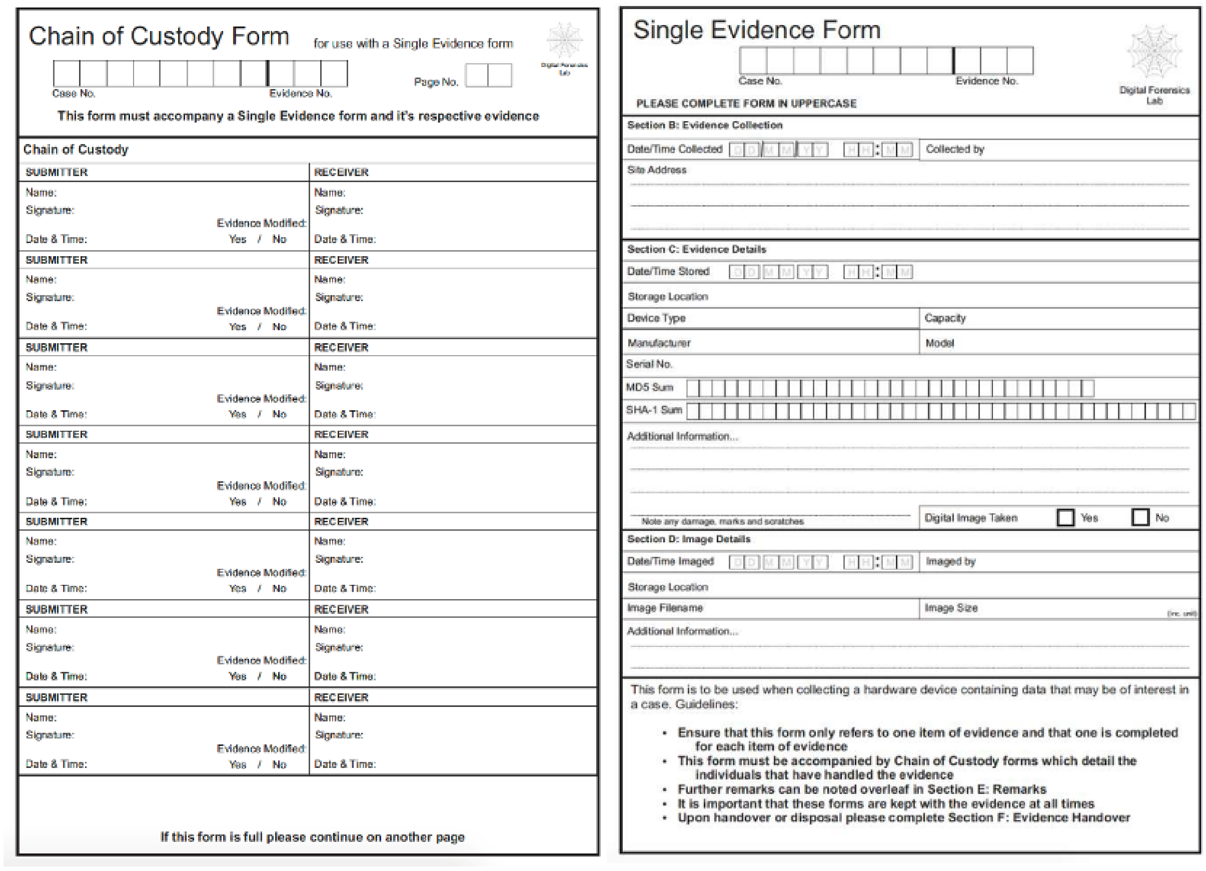


Figure 11 – Standard Code

Do not cover identifying information

- Digital evidence with openings and moving components should be sealed with tamper-evident seals or tamper-evident tape
- Devices should be stored in static dissipation and faraday bag
- Document
- Document in a proper report and signed by all involved
- Fill out the equipment's Chain of Custody (Figure 12)



Chain of Custody Form for use with a Single Evidence form

Case No. Evidence No. Page No.

Digital Forensics Lab

This form must accompany a Single Evidence form and it's respective evidence

Chain of Custody

SUBMITTER	RECEIVER
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:
SUBMITTER	**RECEIVER**
Name: Signature: Evidence Modified: Date & Time: Yes / No	Name: Signature: Date & Time:

If this form is full please continue on another page

Single Evidence Form

Case No. Evidence No.

Digital Forensics Lab

PLEASE COMPLETE FORM IN UPPERCASE

Section B: Evidence Collection

Date/Time Collected D D M M Y Y H H : M M	Collected by
Site Address	

Section C: Evidence Details

Date/Time Stored D D M M Y Y H H : M M	
Storage Location	
Device Type	Capacity
Manufacturer	Model
Serial No.	
MD5 Sum	
SHA-1 Sum	
Additional Information... Note any damage, marks and scratches	Digital Image Taken ☐ Yes ☐ No

Section D: Image Details

Date/Time Imaged D D M M Y Y H H : M M	Imaged by
Storage Location	
Image Filename	Image Size (inc. unit)
Additional Information...	

This form is to be used when collecting a hardware device containing data that may be of interest in a case. Guidelines:

- Ensure that this form only refers to one item of evidence and that one is completed for each item of evidence
- This form must be accompanied by Chain of Custody forms which detail the individuals that have handled the evidence
- Further remarks can be noted overleaf in Section E: Remarks
- It is important that these forms are kept with the evidence at all times
- Upon handover or disposal please complete Section F: Evidence Handover

Figure 12 – Chain of custody

Source: https://www.dfir.training/index.php?option=com_jreviews&format=ajax&url=media/download&m=wx99T&1661384494937

3.Assessment and decision is the phase after the knowledge of the existence of a possible incident, gathering all useful information about what happened, leading to a decision on the confirmaton of a IT security incident. With this confirmation, a quick response must be made in order to mitigate and solve it .

4.Response is the phase where the incident is classified, in accordance with that defined in phase 1, categorising the incident and classifying its severity, carrying out the necessary procedures for its mitigation and resolution, using crisis management mechanisms and incident reporting to the competent authorities, in the Portuguese case are the Judicial Police (PJ), the National Cyber Security Centre (CNCS) and the National Data Protection Commission (CNPD).

5.Lessons Learned is one of the most necessary phases, as it enables measures to be implemented so that a similar incident does not happen again, as well as using the knowledge generated to improve the organisation's security system.

See: https://www.cncs.gov.pt/certpt/coordenacao-da-resposta-a-incidents/

2.3. Relationship between the incident resolution process and forensic computing

These procedures relate to digital forensics (Figure 13) whenever anincident is detected and we want to know more about it, in an attempt to bring the perpetrator to justice. Thus, in the incident management process, we have the same investigation process, indicated in section 1.6, regarding the 6 steps of the analysis framework.

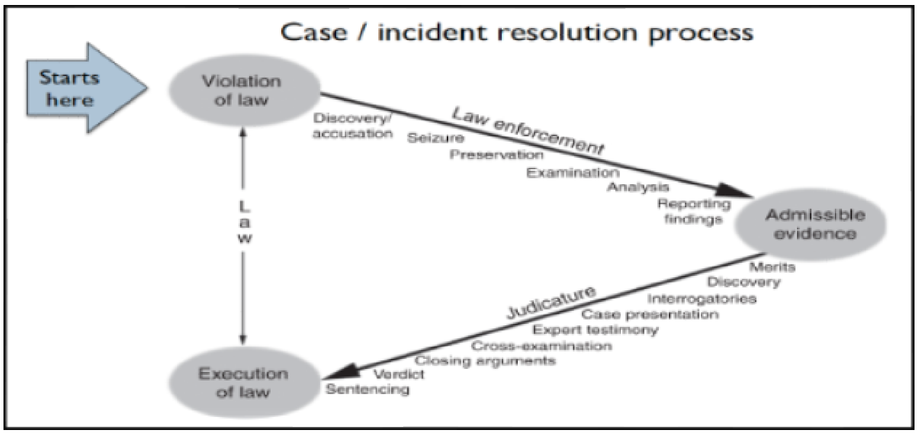


Figure 13 – Incident resolution process

Source: http://www.c-jump.com/bcc/t155t/Week03a/W24_0030_overview_of_caseinci.htm

3. Digital evidence acquisition procedures

Computer forensics is the process of acquiring, analysing and preserving digital evidences, using standardised procedures that allow the integrity of the evidence image.

It is therefore necessary to ensure that the forensic acquisition of storage devices is carried out in accordance with internationally accepted best practices, one of them being the prior sterilisation of the hard disk that will receive the copy of the data, as described in the following point.

3.1. Sterilisation procedure

The sterilisation procedures is intended to ensure that our target device is ready to receive the original information. Sterilisation aims to write all bits of our collection disk with the value 0 (zero), thus ensuring that no previous information is presente on it.

After sterilisation it is always necessary to validate, checking that the sterilisation has occured in the correct way. After this validation we can proceed with formatting the disk to an appropriate file system.

There is a lot of software that allows you to perform this procedures of sterilising a disk and formatting it to the desired file system. Here we will demonstrate the process using the tools present in most Linux distributions, lsblk,fdisk and dc3dd.

The first step is to identify the disc to be sterilized. It is very important that this identification is univocal and confirmed as many times as necessary. A sterilized disc cannot be recovered.

3.1.1. Device identification

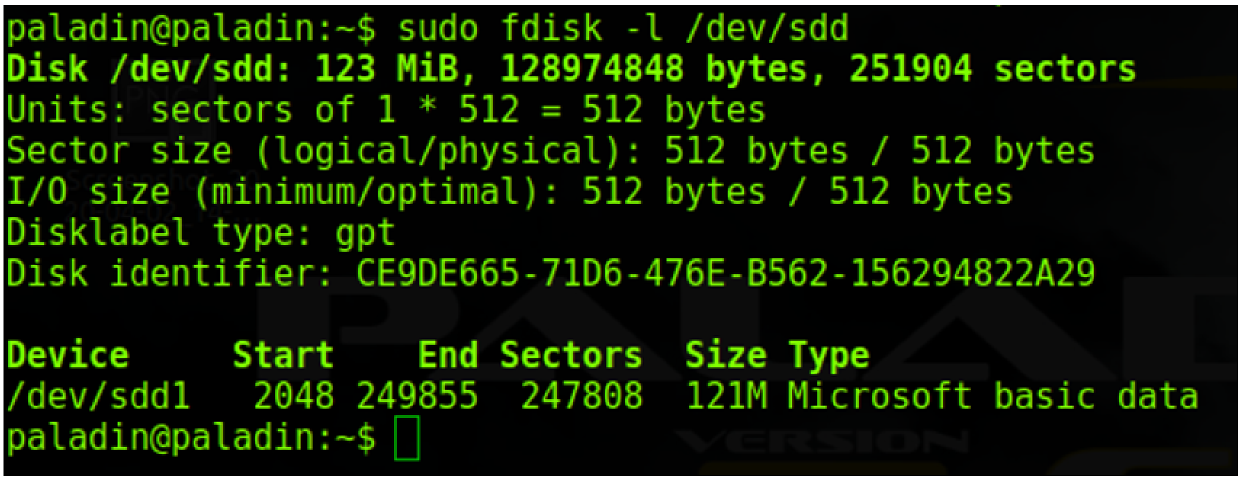
The device identification is done through a series of commands. First of all it is necessary to determine which volumes installed on the computer to be used. A good practice is to use a computer with only the disk to be sterilized connected, starting the operating system from a liveCD or a USB stick. This reduces the possbility of errors in the identification of the disk:

```
$ lsblk | grep sd*
```

This command will list all the storage devices recognised by the operating system. All disks will be displayed as well as their size and partitions. This command only helps us to know what is the name of our disk in the computer.

In case of doubt we can still use the command (Figure 14)

*fdisk -l /dev/sd**



```
paladin@paladin:~$ sudo fdisk -l /dev/sdd
Disk /dev/sdd: 123 MiB, 128974848 bytes, 251904 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: CE9DE665-71D6-476E-B562-156294822A29

Device      Start    End Sectors  Size Type
/dev/sdd1    2048 249855  247808  121M Microsoft basic data
paladin@paladin:~$ 
```

Figure 14 – identification of the device

This command gives us more information about the desired disk.

3.1.2. Sterilisation of the target disk

The sterilization process is nothing more than writing the whole disk with the value 0 (zero), that is, force all the bits of the hard disk to acquire the value zero.

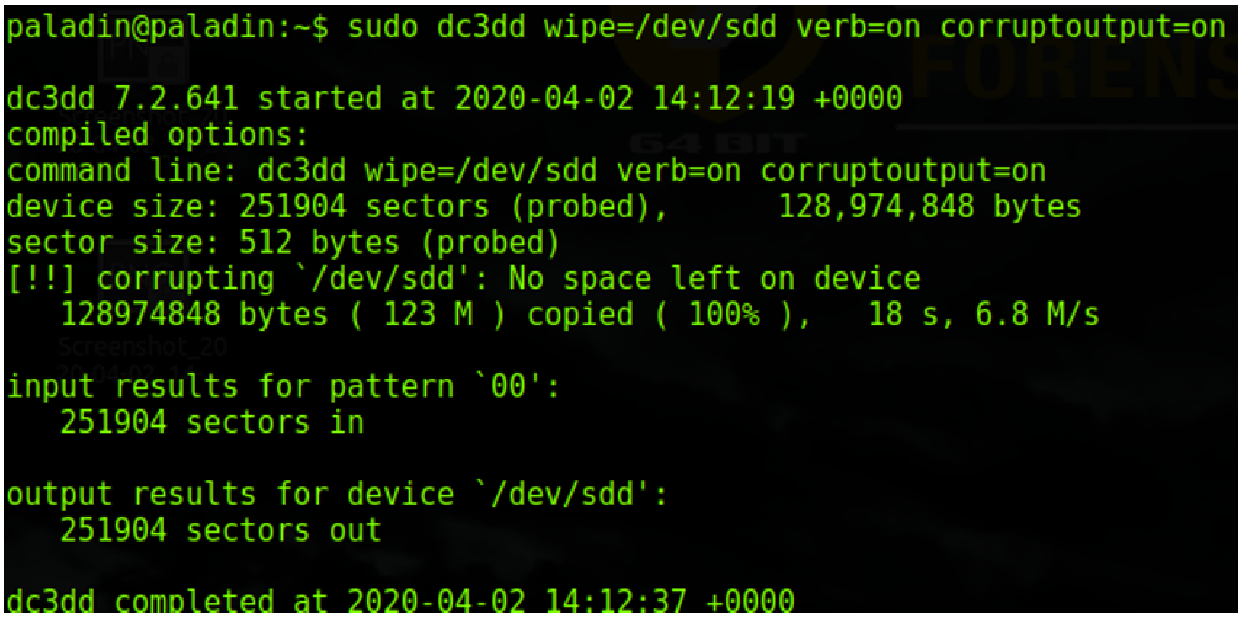
For this task you can use tools like Live-CD DBAN (www.dban.org) or in Windows the Eraser (eraser.heidi.ie).

In Linux wen can use the commands (Figure 15)

```
dc3dd wipe=/dev/sdd verb=on corruptoutput=on
```

or

```
dcfldd if=/dev/zero of=/dev/sdb bs=8k conv=noerror,sync
```



```
paladin@paladin:~$ sudo dc3dd wipe=/dev/sdd verb=on corruptoutput=on

dc3dd 7.2.641 started at 2020-04-02 14:12:19 +0000
compiled options:
command line: dc3dd wipe=/dev/sdd verb=on corruptoutput=on
device size: 251904 sectors (probed),      128,974,848 bytes
sector size: 512 bytes (probed)
[!!] corrupting `/dev/sdd': No space left on device
   128974848 bytes ( 123 M ) copied ( 100% ),   18 s, 6.8 M/s

input results for pattern `00':
   251904 sectors in

output results for device `/dev/sdd':
   251904 sectors out

dc3dd completed at 2020-04-02 14:12:37 +0000
```

Figure 15 – Sterilisation of the target disk

This command performs a write of all the bits of a hard disk, so it will take longer the bigger the hard disk. In our example a device of only 123mb took 18 seconds to be written, however a hard disk of 1 TB, may take more than 8 hours. It is also important to note that depending on the hard disk technology, this time may be higher or lower, depending on its writing speed.

In Microsoft Windows, the sterilization of the destination disk, can be performed through the diskpart command.

Performing the identification of the destination disk by:

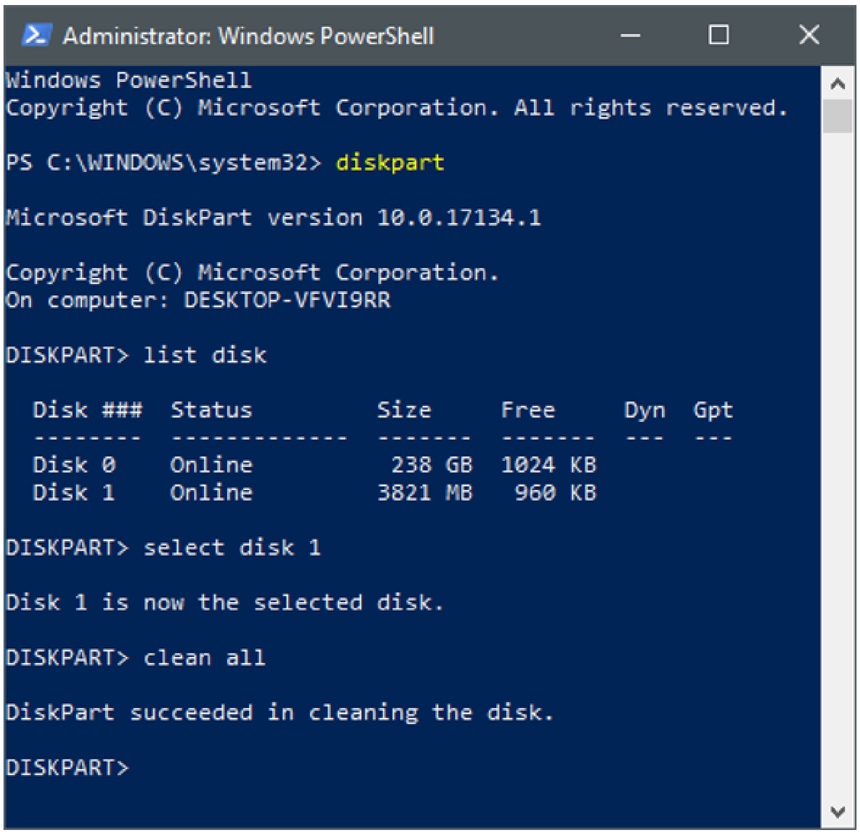
LIST DISK (device identification)

LIST VOLUME (volume identification)

SELECT DISK 1 (select the disk to be sterilised)

Performing the sterilisation (Figure 16)

CLEAN ALL



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> diskpart

Microsoft DiskPart version 10.0.17134.1

Copyright (C) Microsoft Corporation.
On computer: DESKTOP-VFVI9RR

DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online          238 GB  1024 KB
  Disk 1    Online         3821 MB   960 KB

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> clean all

DiskPart succeeded in cleaning the disk.

DISKPART>
```

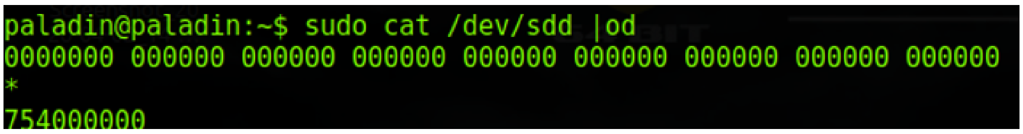
Figure 16 – Destination disk sterilisation under Windows

3.1.3. Verification of the Sterilisation

Finally, it is important to verify that the writing of the hard disk was effective, for this we execute the command:

cat /dev/sdb |od

If the write was successful, the output of the command will be 0000000, which indicates that the hard disk is written with zeros only (Figure 17).



```
paladin@paladin:~$ sudo cat /dev/sdd |od
0000000 000000 000000 000000 000000 000000 000000 000000 000000
*
754000000
```

Figure 17 – verification of the sterilisation

The “cat” command will display the device’s contents, while the “|od” argument will convert this content to octal base, so only zeros will be displayed when the sterilization was successful.

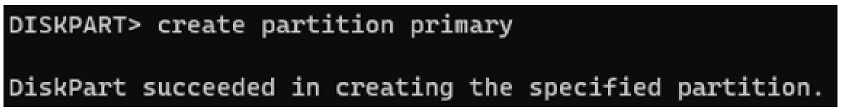
Apart from the procedures presented, there are other methods and different commands that can be used, such as displaying in hexadecimal format, or others.

3.1.4. Formatting

After the sterilisation, it is necessary to format the disk, enabling it to receive data.

This formatment can still be performed using Diskpart, as follows:

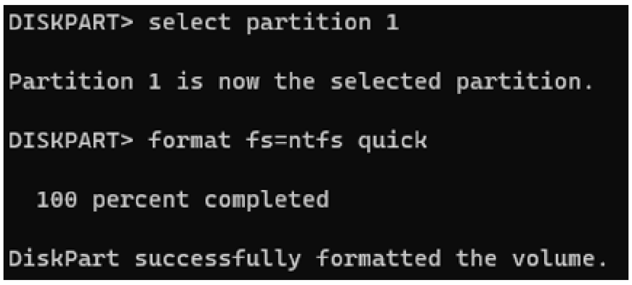
Creating the primary partition (Figure 18)



```
DISKPART> create partition primary

DiskPart succeeded in creating the specified partition.
```

Figure 18 – Creation of the primary partition

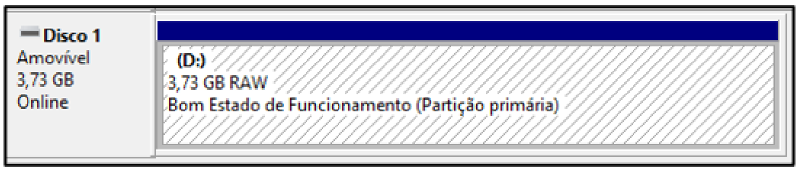
NTFS formatting (Figure 19)



```
DISKPART> select partition 1

Partition 1 is now the selected partition.

DISKPART> format fs=ntfs quick

  100 percent completed

DiskPart successfully formatted the volume.
```

Figure 19 - Formatment

From the graphical environment, the Disk Management application can be used through the DISKMGMT.MSC command (Figure 20).

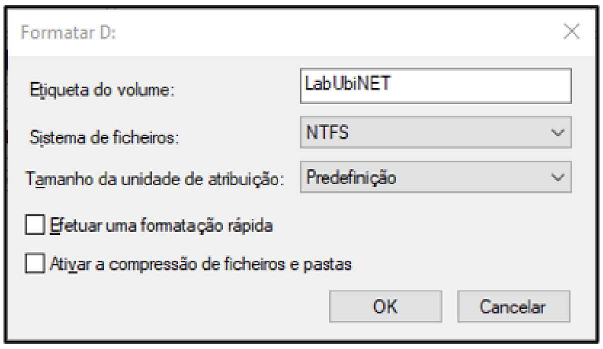


Disco 1
Amovível
3,73 GB
Online

(D:)
3,73 GB RAW
Bom Estado de Funcionamento (Partição primária)

Figure 20 - Disk Management

Right-click on the desired disk and select “Format”, then indicate the name and the desired file system. Fast formatting can be selected, since the disk has been previously sterilized (Figure 21).


Formatar D:

Etiqueta do volume: LabUbiNET

Sistema de ficheiros: NTFS

Tamanho da unidade de atribuição: Predefinição

☐ Efetuar uma formatação rápida

☐ Ativar a compressão de ficheiros e pastas

Figure 21 – Formatting using Disk Management

3.2. Identification of data storage devices

Storage devices have undergone a rapid evolution in the recent years, requiring the forensic analyst to be attentive and research each equipment under analysis, seeking to know about all the data storage devices that the equipment supports. In this way, the analyst can seek to identify all such equipment (Figure 22).

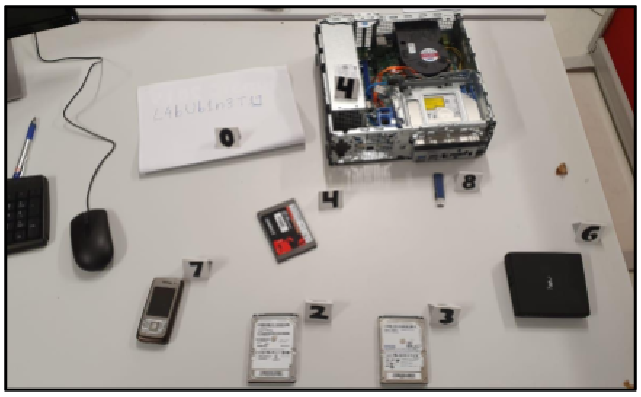


Figure 22 – Identification of storage devices

3.3. Photographic reportage

After identifying the equipment and respective data storage devices, it is important to record their current condition. To do so, assign an internal designation to each piece of equipment and photograph the devices at different angles (Figure 23), using a metric scale and paying special attention to any damage they may have. This is information that may be important in court.

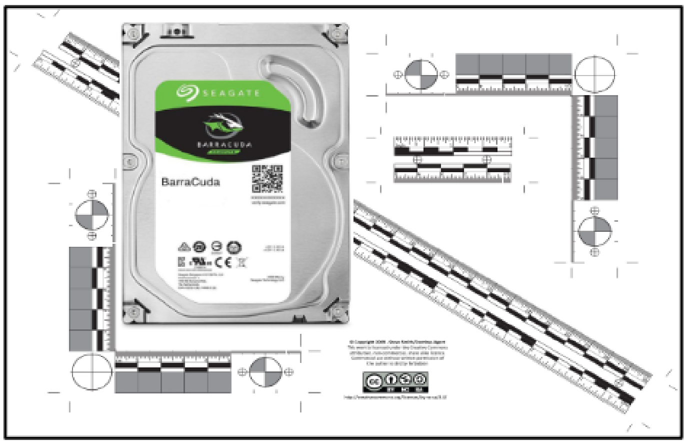


Figure 23 – Photographic Reportage

3.4. Forensic Scope Distributions

Taking into account the techniques and procedures for the acquisition and analysis of data storage devices, it is relevant to have knowledge of forensic Linux distributions. These have a set of tools which allow the acquisition and analysis of information, taking into account the best practices. These are generally Live distributions, which do not need to be installed on the computer, but which allow you to connect the disks without worrying about blocking writing, as they come natively configured without automatically mounting them on the system.

The distributions we indicate are as follows:

- **CAINE (Computer Aided INvestigative Environment Live CD/DVD)**
- **DFF (Digital Forensics Framework)**
- **SANS SIFT (Sans Investigative Forensics Toolkit)**
- **Paladin Edge (Sumuri)**

3.5. Acquisition techniques

The acquisition procedures is determinant to guarantee the integrity of the digital evidence and to facilitate its analysis process, being one of the techniques used in the scope of the good practices in digital forensics, which guarantees the admissibility of the evidence extracted in court.

We call acquisition the bit-by-bit binary copy of data storage devices, and there is also the so-called forensic copy or duplication (Figure 24).

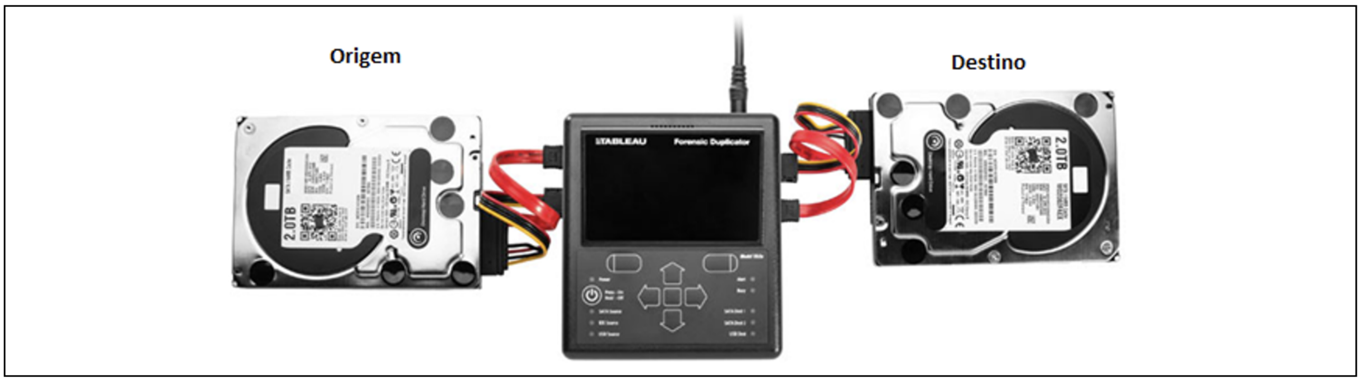


Figure 24 – Storage Device Duplicator

Source: https://security.opentext.com/tableau/hardware/details/td2u

In any of the types of forensic collection used, it is important to bear in mind that the destination disk should have greater capacity than the source one.

3.5.1. Write Blocker

In the acquisition procedures, a hardware device (Figure 25) should be used to block writing to the source disk. In this way the integrity of the data on that disk is guaranteed, protecting the disk against inadvertent changes, such as from the operating system or antivirus, as well as the validation of the data copied to the destination disk.

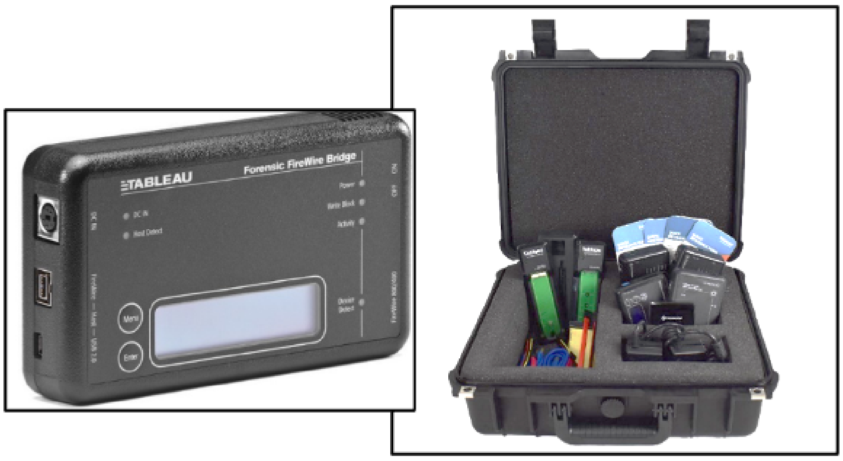


Figure 25 – Write Blocker

Source: www2.guidancesoftware.com

If you don't have access to a hardware write blocker, you can use software write blockers. These require greater care in validating their proper operation, since they depend on the operating system we are using. Some examples are presented in Figure 26.

DFF
DIGITAL FORENSICS FRAMEWORK
EnCase® Forensic
ForensicBlock
C.A.IN.E.
Computer Aided Investigative Environment
ForensicSoft™
www.forensicsoft.com
USB Write Blocker for ALL Windows
Forensic - Write Blocker for ALL Windows version 1.3
Brought to you by: securitemultise

Figure 26 – Forensic software with write-blocking

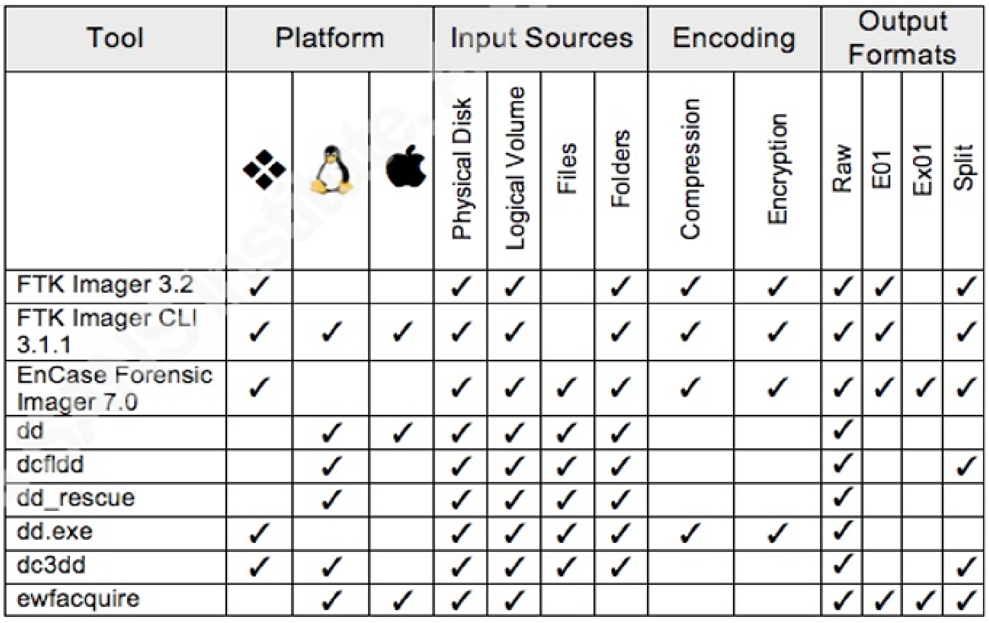
The write blocking on the Microsoft Windows operating system can be done by creating the registry key

“HKLM\SYSTEM\ControlSet001\Control\StorageDevicePolicies\WriteProtect”, as described below:

- ”regedit” in administrator mode and go to the following path:
“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control”
- Create a new key in “Control” with the name: **“StorageDevicePolicies”**
- Add a new value of type “DWORD (32-bit)”, with the name: **“WriteProtect”**
- Change its information from “**0**” to **“1”**
- Test the lock with several storage devices

3.5.2. Comparison of procurement applications

It is possible to perform the forensic acquisition procedures in several operating systems, as there are multiple applications with the capacity to perform this task, an example of which is Figure 27 comparative of these same applications.



Tool	Platform			Input Sources				Encoding		Output Formats			
	❖	🐧		Physical Disk	Logical Volume	Files	Folders	Compression	Encryption	Raw	E01	Ex01	Split
FTK Imager 3.2	✓			✓	✓		✓	✓	✓	✓	✓		✓
FTK Imager CLI 3.1.1	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓
EnCase Forensic Imager 7.0	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
dd		✓	✓	✓	✓	✓	✓			✓			
dcfldd		✓		✓	✓	✓	✓			✓			✓
dd_rescue		✓		✓	✓	✓	✓			✓			
dd.exe	✓			✓	✓	✓	✓	✓	✓	✓			
dc3dd	✓	✓		✓	✓	✓	✓			✓			✓
ewfacquire		✓	✓	✓	✓					✓	✓	✓	✓

Figure 27 – Comparison of procurement applications

Source: Vandeven, Sally, The SANS Institute, 2014

Read: https://www.raedts.biz/forensics/forensic-imaging-tools-compared-tested/

3.5.3. Linux Acquisition

The acquisition through the Linux operating system, among other applications, can be done through dcfldd or dc3dd, applications derived from the well-known dd.

The device and its partition table must be identified

mmls /dev/sdb

Acquisition through dclfdd:

dcfldd if=/dev/sdb hash=md5,sha256 hashwindow=10G md5log=md5.txt sha256log=sha256.txt hashconv=after bs=512 conv=noerror,sync split=10G of=diskimage.dd

Or acquisition through dc3dd:

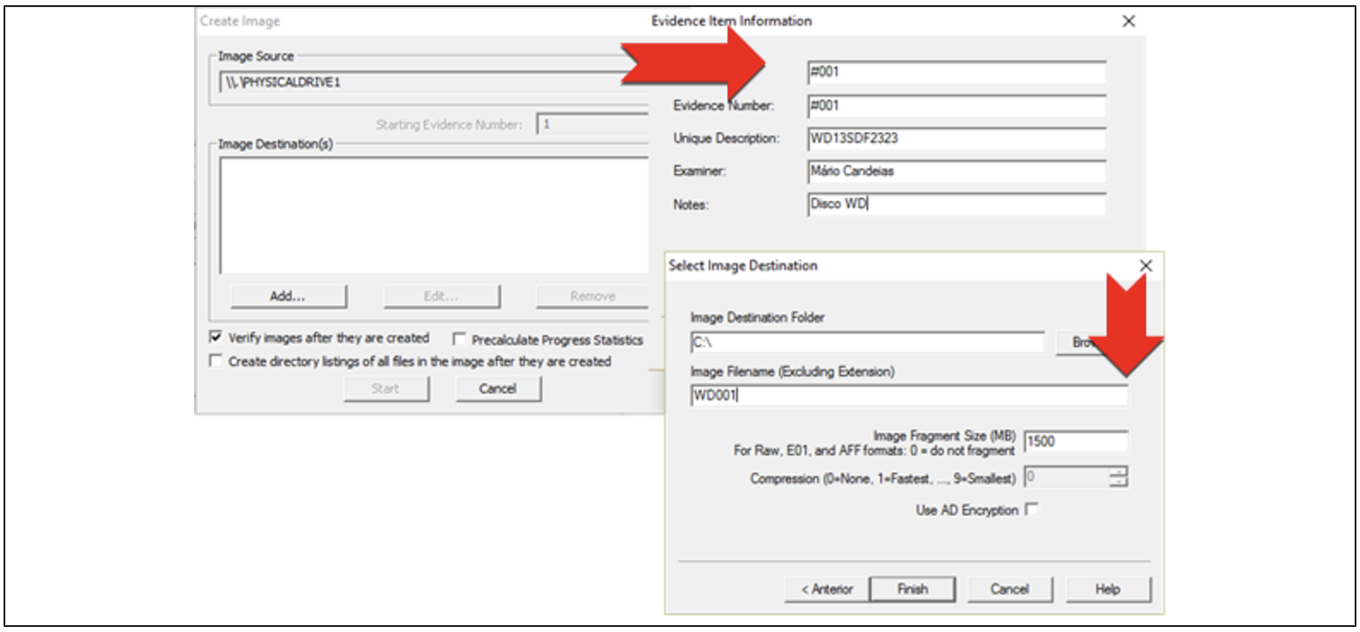
dc3dd if=/dev/sdb hash=md5,sha256 hlog=hash_log.log log= diskimage.log rec=off of=diskimage.dd

The log files should be checked, identifying the correspondence of the hashes of the contents of the source disk with the contents of the forensic image created in this procedure. It is also important to verify the unreadable content, or bad sectors, which in both procedures will remain without any written value, leaving this space with a value of 0 (zero).

3.5.4. Windows Acquisition

The acquisition through the MS Windows operating system, among other applications, can be done through the well-known free application FTK Imager from the company Acess Data.

You should select the Create Disk Image / Physical Drive option in the FTK Imager, as shown in Figure 28.


Create Image
Image Source
\\.\PHYSICALDRIVE1
Starting Evidence Number: 1
Image Destination(s)
Add...
Edit...
Remove
Verify images after they are created
Precalculate Progress Statistics
Create directory listings of all files in the image after they are created
Start
Cancel
Evidence Item Information
Select Image Destination
Bro
< Anterior
Finish
Cancel
Help

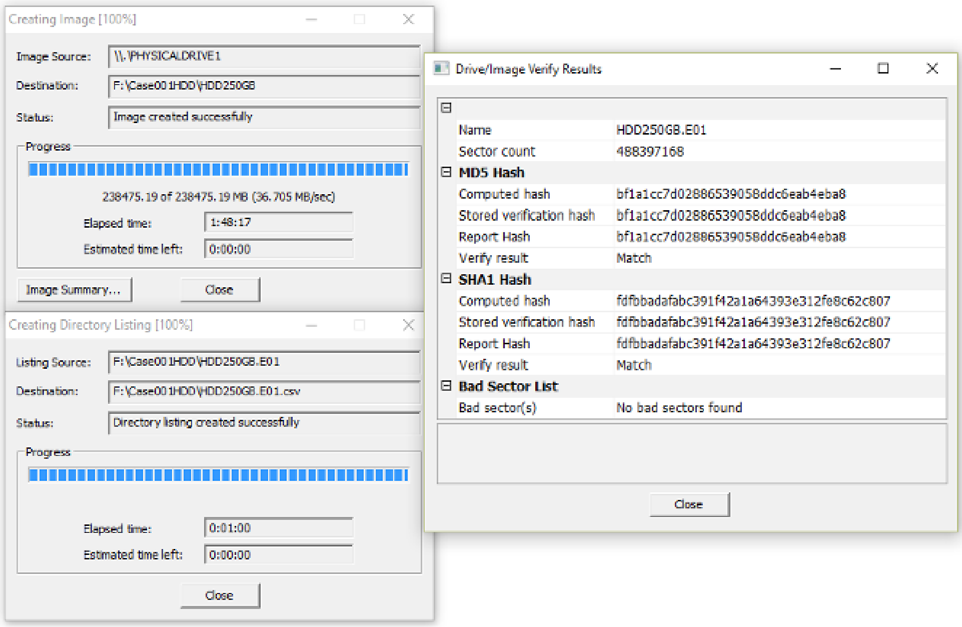
Evidence Item Information: #001
Evidence Number: #001
Unique Description: WD13SDF2323
Examiner: Mário Candeias
Notes: Disco WD

Image Destination Folder: C:\
Image Filename (Excluding Extension): WD001
Image Fragment Size (MB) For Raw, E01, and AFF formats: 0 = do not fragment: 1500
Compression (0=None, 1=Fastest, ..., 9=Smallest): 0
Use AD Encryption

Figure 28 – Acquisition procedure with the AccessData FTK Imager

Before the beginning of the acquisition procedures, it is possible to select the option to create a list of all the files, which will be created after the acquisition and will be saved in the ".csv" format.

After the start of the acquisition, a validation window will be displayed, with the validation of the hash correspondence, as well as the information of the unread sectors, also called bad sectors (Figure 29).



Creating Image [100%]

Image Source:	\\.\PHYSICALDRIVE1
Destination:	F:\Case001HDD\HDD250GB
Status:	Image created successfully

Progress

238475.19 of 238475.19 MB (36.705 MB/sec)

Elapsed time:	1:48:17
Estimated time left:	0:00:00

Creating Directory Listing [100%]

Listing Source:	F:\Case001HDD\HDD250GB.E01
Destination:	F:\Case001HDD\HDD250GB.E01.csv
Status:	Directory listing created successfully

Progress

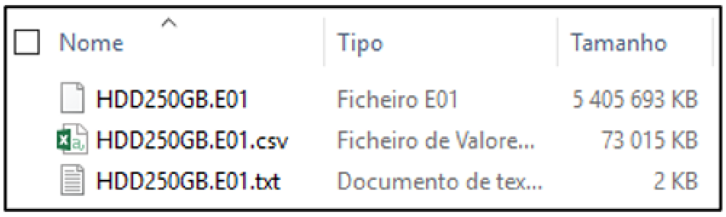
Elapsed time:	0:01:00
Estimated time left:	0:00:00

Drive/Image Verify Results

Name	HDD250GB.E01
Sector count	488397168
MD5 Hash	
Computed hash	bf1a1cc7d02886539058ddc6eab4eba8
Stored verification hash	bf1a1cc7d02886539058ddc6eab4eba8
Report Hash	bf1a1cc7d02886539058ddc6eab4eba8
Verify result	Match
SHA1 Hash	
Computed hash	fdfbbadafabc391f42a1a64393e312fe8c62c807
Stored verification hash	fdfbbadafabc391f42a1a64393e312fe8c62c807
Report Hash	fdfbbadafabc391f42a1a64393e312fe8c62c807
Verify result	Match
Bad Sector List	
Bad sector(s)	No bad sectors found

Figure 29 – Acquisition result with AccessData FTK Imager

We will also have that same information in a text file, stored in the same location as the forensic image file (Figure 30). The ".csv" file will also be in the same location, if we have selected the create the file list.



Nome	Tipo	Tamanho
HDD250GB.E01	Ficheiro E01	5 405 693 KB
HDD250GB.E01.csv	Ficheiro de Valore...	73 015 KB
HDD250GB.E01.txt	Documento de tex...	2 KB

Figure 30 – Files resulting from the acquisition with AccessData FTKImager

4. Acquisition and analysis of volatile information

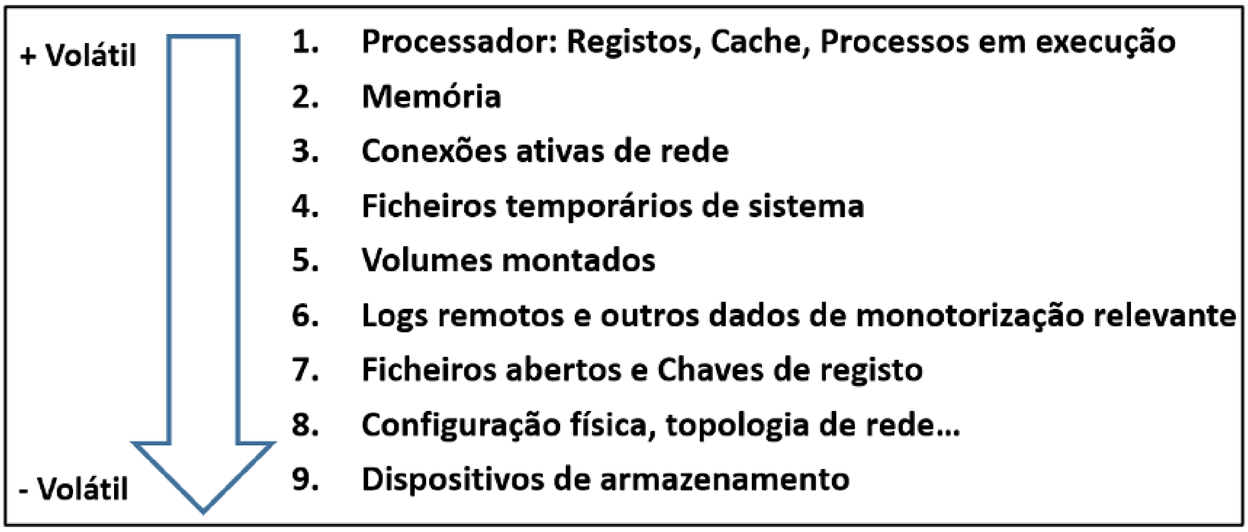
Volatile information is information that is lost when a system is shut down or loses power. Volatile information generally exists in physical memory, or RAM, and consists of information about processes, network connections, open files, clipboard, and similar. This information describes the state of the system at a given time.

When performing a live-data forensics analysis, one of the first things researchers should collect is the contents of RAM. By collecting this information first, the impact of their data collection on the RAM content is minimised, however this capture may cause system instability or even lead to a Blue Screen of Death (BSoD), leading several authors to indicate that this procedures should be performed after the collection of other volatile information, being prioritised according to each situation.

Some of the specific types of volatile information that should be collected:

- RAM memory
- System date and time
- Network information
- Logged in users
- Open files
- Network connections
- Information about running processes
- Mapping between processes and ports
- Network status
- Clipboard content
- Information about services and drivers
- History of executed commands
- Mapped drives
- Shares
- Passwords and cryptographic keys

Of these, it is necessary to identify for each specific case, which information is more volatile and will be more important to obtain first. Here a possible sequence of information in order of volatility (Figure 31).



1. Processador: Registos, Cache, Processos em execução
2. Memória
3. Conexões ativas de rede
4. Ficheiros temporários de sistema
5. Volumes montados
6. Logs remotos e outros dados de monotorização relevante
7. Ficheiros abertos e Chaves de registo
8. Configuração física, topologia de rede…
9. Dispositivos de armazenamento

Figure 31 – Possible sequence of information in order of volatility

4.1. Process of capturing volatile information

First of all it is necessary to ensure the stability of our working machine. One of the problems that can be experienced is the automatic restart of the machine, caused by automatic updates of the operating system. It is therefore advisable to deactivate, not the operating system updates, but their ability to force a restart of the machine. This restart could, for example, interrupt a disk acquisition or an image analysis.

Another good practice is to deny write access to removable disks, thus preventing the source disks from being altered.

A **volatile data** is any data that can be lost when the system is shut down, such as a record of a connection to a website that is still present in RAM or the system clipboard. The collection of this data must occur while the system is running.

Live-Data Forensics, is the technique used to collect data that is volatile and can be lost if the device loses power.

Memory DUMP is the procedure of storing in a file, all data present at a given moment in the physical memory of the computer.

When collecting this type of data, it is necessary to take into account the order of volatility of the data, adapting the collection to the category of data that is of most interest. If our goal is to identify the sending of an email to a certain address, it would not make much sense to give priority to the identification of processes instead of collecting passwords, or access credentials, that allow access to the email address.

It is very important that live data collection is properly documented, preferable by creating a collection team of at least two people, to ensure that the collection procedures are carried out by one person while the other documents the collection process used.

It is also important to guarantee the minimum alteration to the system to be analysed and if any alteration needs to be made, it should be registered in a report for future memory.

Recommendations on the use of scripts:

· Use of environment variables
(ex.: cmd: %COMPUTERNAME% / PS: $env:Computername)

· Running in Administrator mode

4.1.1. Native System Tools

Considering the minimal digital footprint on the device, we should whenever possible use forensic tools to collect useful information that allows a more efficient analysis. An example of this is information related to full-disk encryption. However, the MS Windows operating system allows the execution of commands and scripts with native tools, being an excellent possibility to obtain the necessary information with a minimal digital footprint.

Command Line	The command line of the MS Windows system is natively one of the most used in the collection of system information, enabling the execution of numerous programs for this purpose	C:\>
Windows Management Instrumentation (WMI)	Windows Management Instrumentation allows access to the operating system via Windows Management Instrumentation Command-line and is anexcellent way of obtaining information about the operating system.	WMI
Windows Batch File	It is a script file which allows you to group a set of commands together, line by line. It allows the use of repetiton structures, conditional structures, the use of variables, typical of a scripting language.	
Powershell	PowerShell is currently the scripting language, initially developed for MS Windows systems, with its open source code and cross-platform support being made available in 2016. With its own shell, PowerShell was developed to allow the execution of cmdlets, also allowing the execution of other shells.	

4.1.2. External Tools

As recourse is made to external tools, care must be taken to thoroughly test each of these tools in order to know exactly what they do in the system. Any external tools used, the date/time of use should be noted, as well as describing the intention of use.

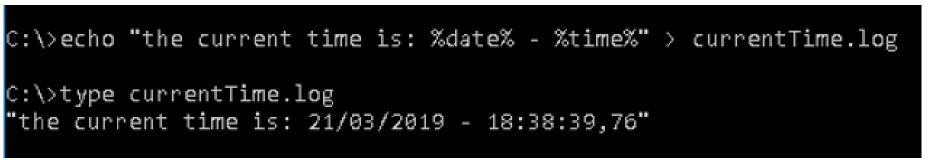
Windows Sysinternals	Windows Sysinternals represents a set of tools originally created by Mark Russinovich, intended to help system administrators manage and monitor Windows systems. https://docs.microsoft.com/en-us/sysinternals/	Windows Sysinternals
Nirsoft	NirSoft represents a set of tools created by Nir Sofer, from which we highlight those which he categorised as forensic. http://www.nirsoft.net/	NirSoft Tools and Utilities
Mitec	Mitec is also a site that offers a set of interesting tools for information gathering and analysis, such as its MiTeC System Information X and Windows Registry Recovery https://www.mitec.cz/	MITEC
Zimmerman	Eric Zimmerman has developed a set of free-to-use tools intended to assist in incident response and forensic analysis.	**Eric Zimmerman Tools**

4.1.3. Date, Time and other System informations

This element should be the first to be collected when performing an investigation. The system date makes it possible to contextualise the information collected later and allows the researcher to build a timeline of events that have occurred, not only in the system under analysis, but through correlation between information from other systems. Another important piece of data is the time elapsed since the last boot (uptime).

Some tools can help researchers in these tasks, such as MiTeC - System Information X[1] and WinAudit[2].

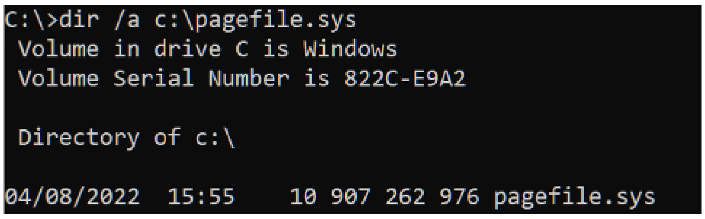
Collection of the date/time of the system being serviced (Figure 32).



```
C:\>echo "the current time is: %date% - %time%" > currentTime.log

C:\>type currentTime.log
"the current time is: 21/03/2019 - 18:38:39,76"
```

Figure 32 – Obtaining the system's date and time.

Collecting the date/time of the last system start-up (Figure 33).



```
C:\>dir /a c:\pagefile.sys
 Volume in drive C is Windows
 Volume Serial Number is 822C-E9A2

 Directory of c:\

04/08/2022  15:55    10 907 262 976 pagefile.sys
```

Figure 33 – Obtaining the date and time of the last system boot

Commands useful for obtaining data from the system:

· Windows version: **ver**

· Environment variables: **set**

· System informations: **systeminfo /fo list >> C:\tmp\info.txt**

· Consult the registry: **reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v ProductName**

· Consult from WMI: **wmic os get name, version**

· Start and shutdown of the system: **TurnedOnTimesView.exe** (Fonte: Nirsoft.net)

Commands useful for obtaining data about System Users

· Users:

o **Net User** [username]

o **Userprofilesview.exe /shtml "f:\temp\profiles.html" /sort "User Name"** (Source: Nirsoft.net)

· Logged Users:

o **PSLoggedOn.exe** (Source: SysInternals)

o **LogonSessions.exe** (Source: SysInternals)

4.1.4. Processes and Applications

It is central to enumerate the processes running on a potentially compromised System. A process is a section or instance of an application that is running. Looking at the running processes in Task Manager gives some information, however much more information can be obtained than can be seen there.

Some of the types of information about the running processes that can be obtained:

· The absolute path of the executable file

· The command used to launch the process

· The amount of time the process is running

- Which user started the process and its level of permissions in the system
- Modules that the process has loaded
- Content of the memory allocated to the process

Examples of programs and commands for acquiring infomation about the **processes running** on the system:

- **Psinfo.exe -h -s -d /accepteula** (Source: SysInternals)
- ***PsList.exe*** (Source: sysinternals)
- ***CurrProcess.exe*** (Source: Nirsoft.net)
- ***tasklist /v***
- ***Wmic process get name, processid, priority, threadcount, privatepagecount***

Examples of programs and commands for acquiring information about services, scheduled tasks and system events:

- [Services] **PsService.exe** (SysInternals)
- [Services] **net start**
- [scheduled tasks] **schtasks**
- [events] **PsLogList.exe** (SysInternals)
- [events] **EventLogSourcesView.exe** (Nirsoft)
- [events] **wevtutil**

4.1.5. Memory

Clipboard is an area in memory where data can be stored for future use. Most Windows applications provide this functionality through the Edit menu and the Copy, Paste or Cut choices. This feature is useful for moving data between applications or documents. Often data stays on the Clipboard for days without the user realizing it.

To collect the data stored in this area of the memory the following application can be used **InsideClipboard.exe** (Nirsoft.net).

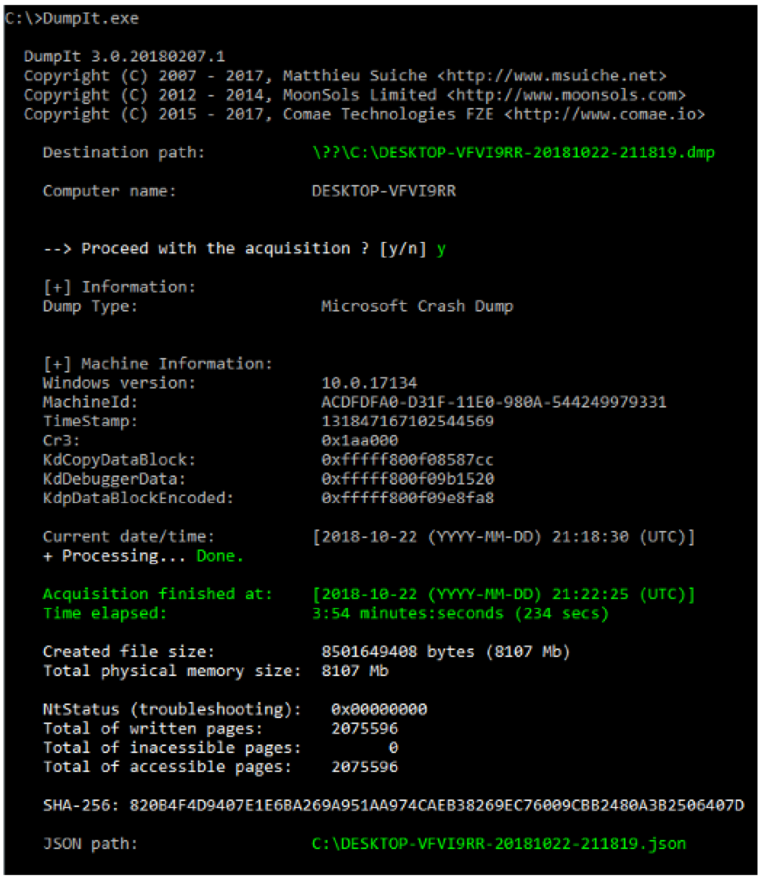
Malware analysts search in memory when dealing with obfuscated malware, because when it is executed it is deciphered into the same memory. In addition Rootkits hide processes, files, registry keys and even network connections. It is possible to check what is hidden from the user's view by analysing RAM memory. This data is very useful for contextualising identified data in future analysis.

4.1.6. Memory Acquisition

The Memory Dump process (Figure 34) is also widely used to diagnose bugs in programs, as these dumps are normally created when there is an error and programs unexpectedly stop working.

These memory dumps are done in binary, octal or hexadecimal format. An investigation can be made using programs, such as:

- DumpIT (moonsols)
- AccessData FTK Imager
- Belkasoft Live RAM Capturer



```
C:\>DumpIt.exe

  DumpIt 3.0.20180207.1
  Copyright (C) 2007 - 2017, Matthieu Suiche <http://www.msuiche.net>
  Copyright (C) 2012 - 2014, MoonSols Limited <http://www.moonsols.com>
  Copyright (C) 2015 - 2017, Comae Technologies FZE <http://www.comae.io>

    Destination path:           \??\C:\DESKTOP-VFVI9RR-20181022-211819.dmp

    Computer name:              DESKTOP-VFVI9RR


    --> Proceed with the acquisition ? [y/n] y

    [+] Information:
    Dump Type:                   Microsoft Crash Dump


    [+] Machine Information:
    Windows version:             10.0.17134
    MachineId:                   ACDFDFA0-D31F-11E0-980A-544249979331
    TimeStamp:                   131847167102544569
    Cr3:                         0x1aa000
    KdCopyDataBlock:             0xfffff800f08587cc
    KdDebuggerData:              0xfffff800f09b1520
    KdpDataBlockEncoded:         0xfffff800f09e8fa8

    Current date/time:          [2018-10-22 (YYYY-MM-DD) 21:18:30 (UTC)]
    + Processing... Done.

    Acquisition finished at:    [2018-10-22 (YYYY-MM-DD) 21:22:25 (UTC)]
    Time elapsed:               3:54 minutes:seconds (234 secs)

    Created file size:           8501649408 bytes (8107 Mb)
    Total physical memory size:  8107 Mb

    NtStatus (troubleshooting):   0x00000000
    Total of written pages:       2075596
    Total of inacessible pages:         0
    Total of accessible pages:    2075596

    SHA-256: 820B4F4D9407E1E6BA269A951AA974CAEB38269EC76009CBB2480A3B2506407D

    JSON path:                  C:\DESKTOP-VFVI9RR-20181022-211819.json
```

Figure 34 – Example of how DumpIT works

There are other files[3], to support the main memory, which must be collected, such as **pagefile.sys**, used by Windows as “virtual memory”. Whenever the system needs to use more memory than is available in RAM. Or **hiberfil.sys** is used to store data from memory when the computer goes into hibernation.

For memory collection in a Linux environment, the *dcfldd* or *insmod programs can be used*.

- dcfldd if=/dev/fmem of=memory.dump
- insmod lime-XX.ko "path="memory.dump" format=raw"

4.1.7. Network information

Collecting volatile information about the computer’s network status: active connections, open ports, routing information and configuration, cache, ARP….

As soon as an incident is reported, the investigator must collect information about the status of network connections to the affected system.

These connections can expire and the information can disappear over time. Looking at this data can help determine if an attacker is still logged into the system, if there are malware-related connections, if there is a process trying to find other machines on the network to propagate that malware, or to send log information to a malicious server.

Collecting this information can provide important clues and add context to other informations collected.

Examples of commands for gathering network information:

- Network card information: **ipconfig /all**
- DNS cache: **ipconfig /displaydns**
- Active network connections: **Netstat -a**
- ARP cache: **Arp -a**
- **Netsh int ipv6 show neigh**

- Wifi events: **Netsh wlan show all**
- Wireless Networks: **WifiInfoView.exe** (Source: Nirsoft.net)
- Routing Table: **Route print**
- Cache connections: **Netstat -c**
- List Cache sessions: **Netstat -s**
- **Net accounts**
- Resource sharing: **Net share**
- Query the Server about DNS: **Nslookup -d**
- List current connections: **Rasdial**
- List Profiles: **Netsh wlan show profiles**

---

[1] www.mitec.cz/msi.html

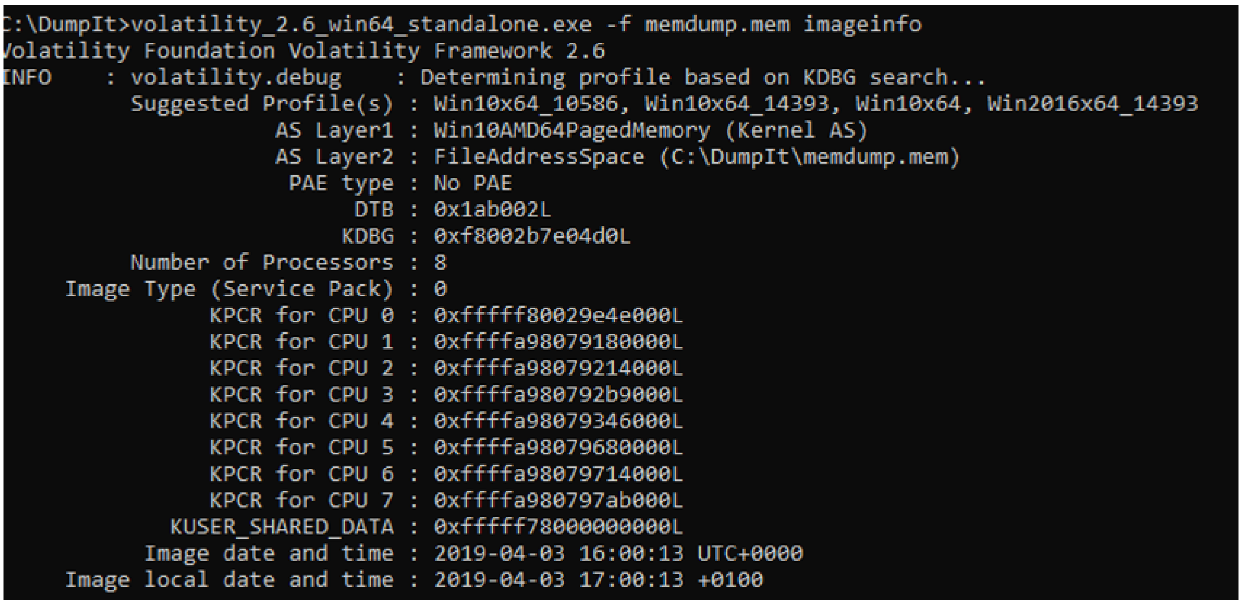
[2] www.parmavex.co.uk

[3] https://www.hackingarticles.in/forensics-analysis-of-pagefile-and-hibersys-file-in-physical-memory/

4.2. Memory Acquisition Analysis

There are some memory dump analysis tools that are based only on the contents of RAM. These contents may be incomplete, since parts of the memory are stored on disk when it is not enough to store all the data. To overcome this problem, Nicholas Paul Maclean published his thesis, "Acquisition and Analysis of Windows Memory", how memory management works on Windows systems and provided an open-source tool called vtop to fully reconstruct the virtual memory space of a process.

For the analysis of the Memory Dump we can use the program Volatility, where it is possible to perform tasks such as obtaining high-level information about the image, where the operating system identification (Figure 35), service pack, hardware, architecture, memory address and the time of the Dump is deduced.



```
:\DumpIt>volatility_2.6_win64_standalone.exe -f memdump.mem imageinfo
olatility Foundation Volatility Framework 2.6
NFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_10586, Win10x64_14393, Win10x64, Win2016x64_14393
                     AS Layer1 : Win10AMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\DumpIt\memdump.mem)
                      PAE type : No PAE
                           DTB : 0x1ab002L
                          KDBG : 0xf8002b7e04d0L
          Number of Processors : 8
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff80029e4e000L
                KPCR for CPU 1 : 0xffffa98079180000L
                KPCR for CPU 2 : 0xffffa98079214000L
                KPCR for CPU 3 : 0xffffa980792b9000L
                KPCR for CPU 4 : 0xffffa98079346000L
                KPCR for CPU 5 : 0xffffa98079680000L
                KPCR for CPU 6 : 0xffffa98079714000L
                KPCR for CPU 7 : 0xffffa980797ab000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2019-04-03 16:00:13 UTC+0000
     Image local date and time : 2019-04-03 17:00:13 +0100
```

Figure 35 – Getting information about a memory dump

Commands useful for performing a record dump:

Export to text:

C:\Regdmp.exe > e:\registryDump.txt

Find Expressions in the exported file:

C:\Find/i "URL" registryDump.txt

Copy of the registry files in use:

C:\RawCopy.exe C:\WINDOWS\system32\config\SYSTEM E:\output -AllAttr

Other useful commands:

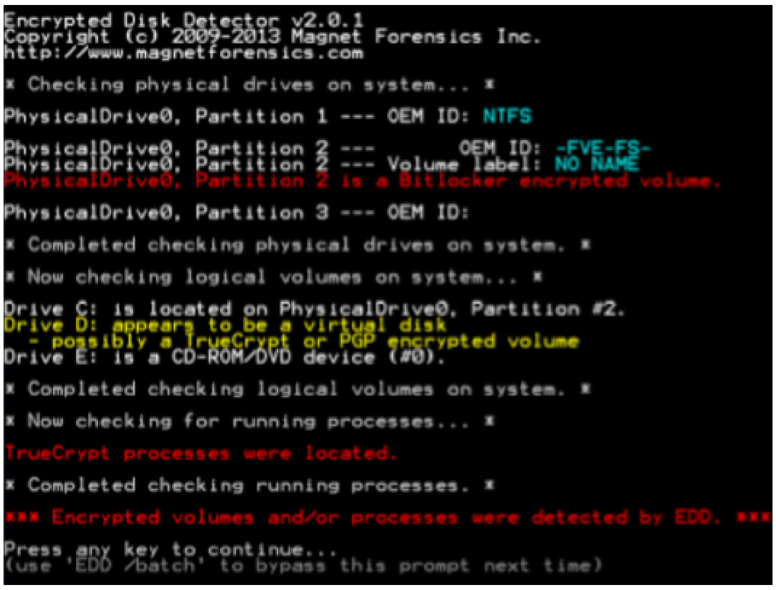
Get a screenshot of the desktop:

C:\nircmd.exe savescreenshot screen1.png (Nirsoft.net)

Checking that the disk is protected with Encryption (Figure 36)

C:\EDD.exe /accepteula /Batch > e:\EncryptedDiskDetector.txt

C:\Manage-bde –protectors c: -get



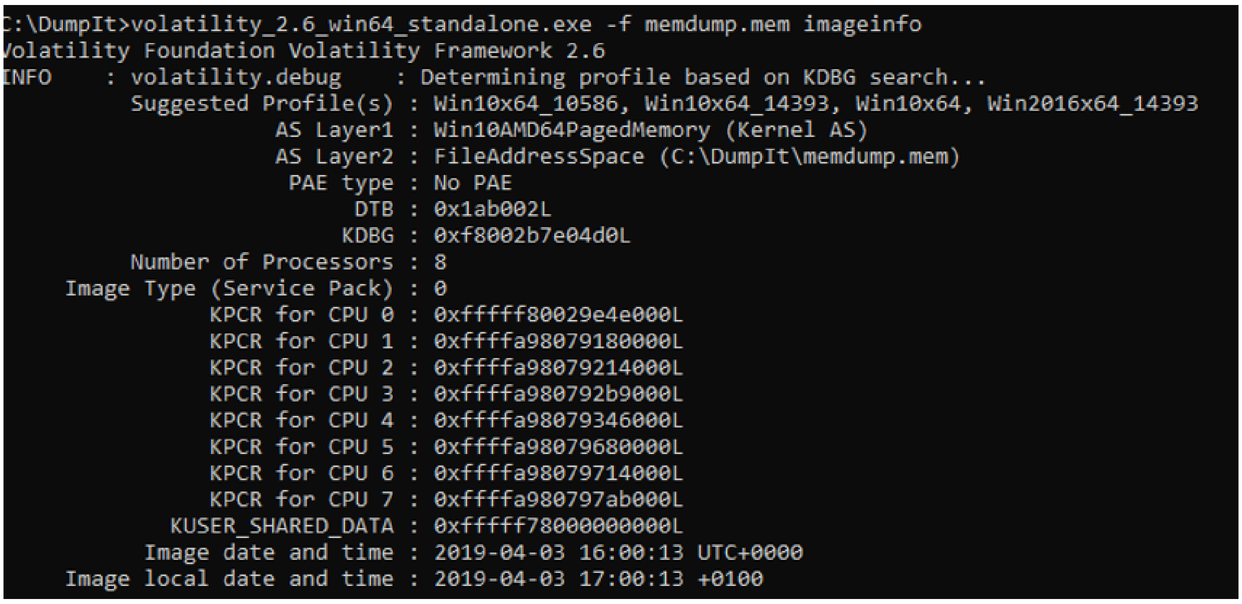
```
Encrypted Disk Detector v2.0.1
Copyright (c) 2009-2013 Magnet Forensics Inc.
http://www.magnetforensics.com

* Checking physical drives on system... *

PhysicalDrive0, Partition 1 --- OEM ID: NTFS

PhysicalDrive0, Partition 2 ---       OEM ID: -FVE-FS-
PhysicalDrive0, Partition 2 --- Volume label: NO NAME
PhysicalDrive0, Partition 2 is a Bitlocker encrypted volume.

PhysicalDrive0, Partition 3 --- OEM ID:

* Completed checking physical drives on system. *

* Now checking logical volumes on system... *

Drive C: is located on PhysicalDrive0, Partition #2.
Drive D: appears to be a virtual disk
   - possibly a TrueCrypt or PGP encrypted volume
Drive E: is a CD-ROM/DVD device (#0).

* Completed checking logical volumes on system. *

* Now checking for running processes... *

TrueCrypt processes were located.

* Completed checking running processes. *

*** Encrypted volumes and/or processes were detected by EDD. ***

Press any key to continue...
(use 'EDD /batch' to bypass this prompt next time)
```

Figure 36 – Identifying an encrypted disk

4.2.1. Sintax of the Volatility program

The first version of the Volatility Framework was released at a Black Hat conference. The software is based on years of academic research in advanced memory analysis and forensics. Volatility now allows researchers to analyze what state the machine was in at the time the acquisition was made, based on data gathered from volatile memory.

The Volatility Framework is based on the Python programming language, being developed in Python 2 its most mature version, being this the one we will address in this topic. With the appearance of Python 3, there was also a need to update the Volatility version, taking advantage of the new Python version and providing it with more automation. In version 2 of the Volatility Framework, the first step in performing a memory analysis is to identify the type of Operating System. To do this we can use the imageinfo command of the Volatility program (Figure 37). This command is useful for obtaining high level information about the image, it indicates the probable operating system identification (profile), service pack, hardware architecture, memory address and the time of the Dump.



```
C:\DumpIt>volatility_2.6_win64_standalone.exe -f memdump.mem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_10586, Win10x64_14393, Win10x64, Win2016x64_14393
                     AS Layer1 : Win10AMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (C:\DumpIt\memdump.mem)
                      PAE type : No PAE
                           DTB : 0x1ab002L
                          KDBG : 0xf8002b7e04d0L
          Number of Processors : 8
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff80029e4e000L
                KPCR for CPU 1 : 0xffffa98079180000L
                KPCR for CPU 2 : 0xffffa98079214000L
                KPCR for CPU 3 : 0xffffa980792b9000L
                KPCR for CPU 4 : 0xffffa98079346000L
                KPCR for CPU 5 : 0xffffa98079680000L
                KPCR for CPU 6 : 0xffffa98079714000L
                KPCR for CPU 7 : 0xffffa980797ab000L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2019-04-03 16:00:13 UTC+0000
     Image local date and time : 2019-04-03 17:00:13 +0100
```

Figure 37 - Volatility – Example output of the imageinfo command

Later we must pass the contents of memory to text files so that an analysis of its contents is possible. Volatility provides a number of plugins for this purpose.

Sintax: **volatility -f <nome_da_imagem> -profile=<tipo_de_OS> <plugin> > <output>**

- **-f:** File resulting from the system acquisition
- **-profile:** instru*ction to use the operating system profile (previously identified)*
- **plugin:** plugin to be executed

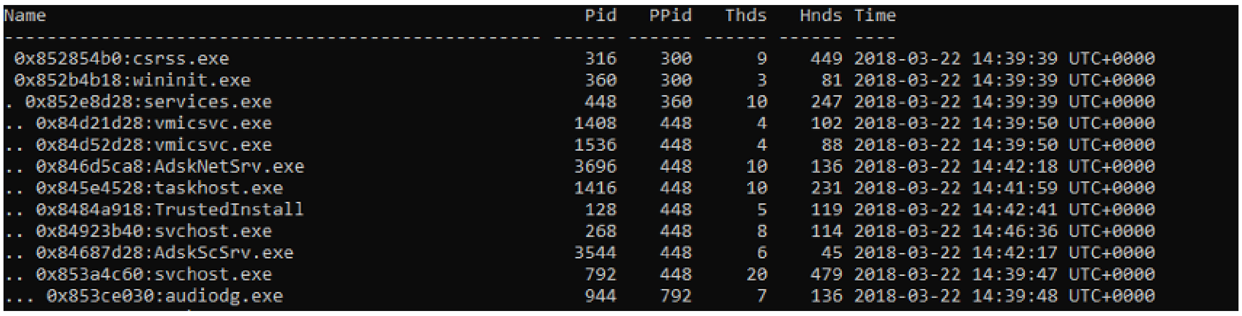
- **output:** file to export the results

4.2.2. Volatility Plugins - Extraction

The plugins that volatility uses are specific to the identification of the respective information in the RAM dump content. We will cover some of these pulgins here.

Pslist List Running Processes

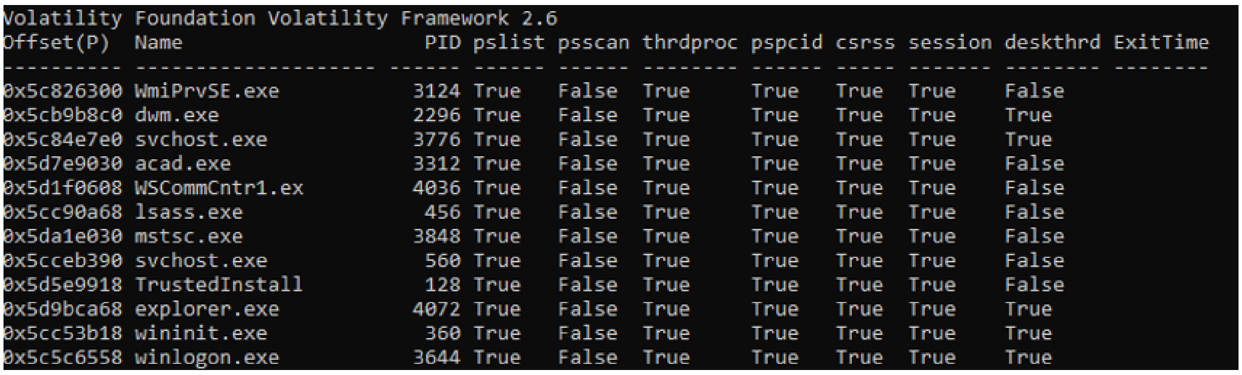
Pstree Display processes, differentiating them in their origin (Figure 38)



```
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x852854b0:csrss.exe                                 316    300      9    449 2018-03-22 14:39:39 UTC+0000
 0x852b4b18:wininit.exe                               360    300      3     81 2018-03-22 14:39:39 UTC+0000
. 0x852e8d28:services.exe                             448    360     10    247 2018-03-22 14:39:39 UTC+0000
.. 0x84d21d28:vmicsvc.exe                            1408    448      4    102 2018-03-22 14:39:50 UTC+0000
.. 0x84d52d28:vmicsvc.exe                            1536    448      4     88 2018-03-22 14:39:50 UTC+0000
.. 0x846d5ca8:AdskNetSrv.exe                         3696    448     10    136 2018-03-22 14:42:18 UTC+0000
.. 0x845e4528:taskhost.exe                           1416    448     10    231 2018-03-22 14:41:59 UTC+0000
.. 0x8484a918:TrustedInstall                          128    448      5    119 2018-03-22 14:42:41 UTC+0000
.. 0x84923b40:svchost.exe                             268    448      8    114 2018-03-22 14:46:36 UTC+0000
.. 0x84687d28:AdskScSrv.exe                          3544    448      6     45 2018-03-22 14:42:17 UTC+0000
.. 0x853a4c60:svchost.exe                             792    448     20    479 2018-03-22 14:39:47 UTC+0000
... 0x853ce030:audiodg.exe                            944    792      7    136 2018-03-22 14:39:48 UTC+0000
```

Figure 38 - volatility pstree Plugin

Psxview Compare processes (Figure 39)

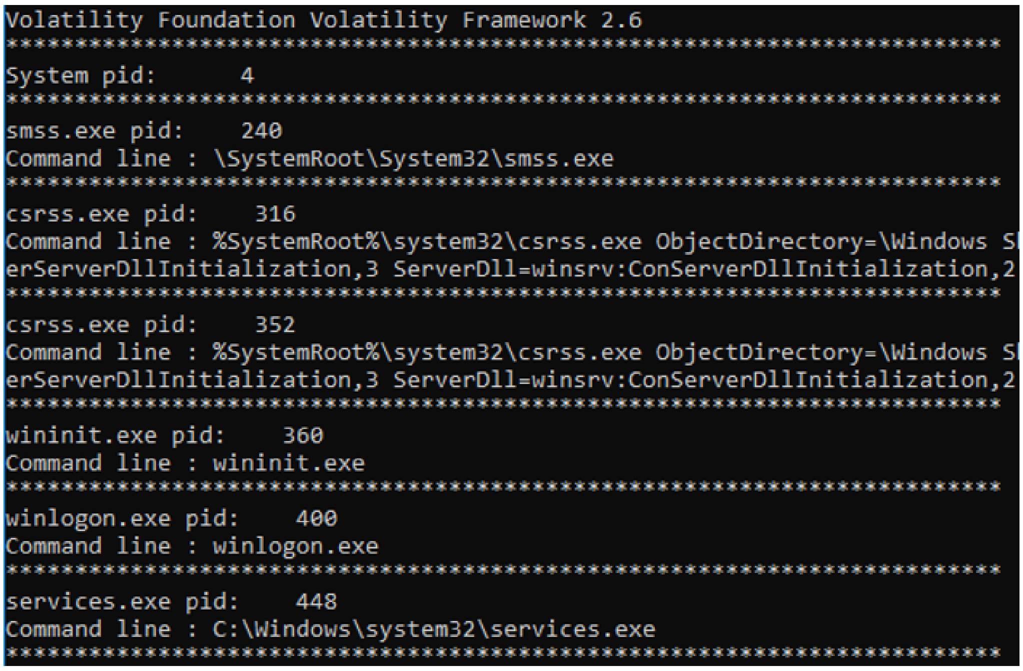


```
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x5c826300 WmiPrvSE.exe           3124 True   False  True     True   True  True    False
0x5cb9b8c0 dwm.exe                2296 True   False  True     True   True  True    True
0x5c84e7e0 svchost.exe            3776 True   False  True     True   True  True    True
0x5d7e9030 acad.exe               3312 True   False  True     True   True  True    False
0x5d1f0608 WSCommCntr1.ex         4036 True   False  True     True   True  True    False
0x5cc90a68 lsass.exe               456 True   False  True     True   True  True    False
0x5da1e030 mstsc.exe              3848 True   False  True     True   True  True    False
0x5cceb390 svchost.exe             560 True   False  True     True   True  True    False
0x5d5e9918 TrustedInstall          128 True   False  True     True   True  True    False
0x5d9bca68 explorer.exe           4072 True   False  True     True   True  True    True
0x5cc53b18 wininit.exe             360 True   False  True     True   True  True    True
0x5c5c6558 winlogon.exe           3644 True   False  True     True   True  True    True
```

Figure 39 – Volatility psxview plugin

Netscan Display network connections

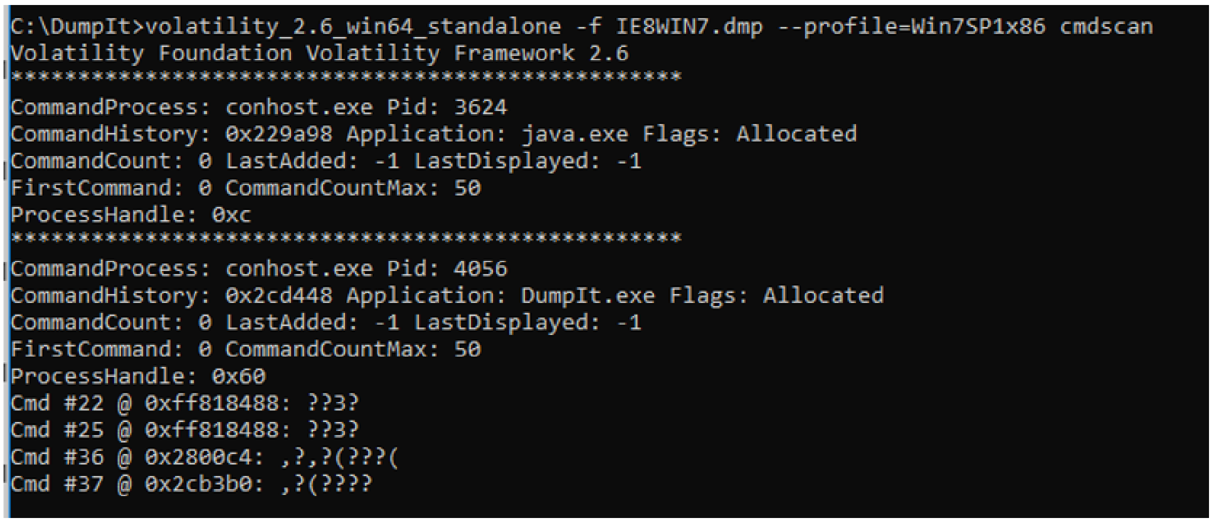
Cmdline Cmdline Compare processes (Figure 40)



```
Volatility Foundation Volatility Framework 2.6
************************************************************************
System pid:      4
************************************************************************
smss.exe pid:    240
Command line : \SystemRoot\System32\smss.exe
************************************************************************
csrss.exe pid:    316
Command line : %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows S
erServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2
************************************************************************
csrss.exe pid:    352
Command line : %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows S
erServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2
************************************************************************
wininit.exe pid:    360
Command line : wininit.exe
************************************************************************
winlogon.exe pid:    400
Command line : winlogon.exe
************************************************************************
services.exe pid:    448
Command line : C:\Windows\system32\services.exe
************************************************************************
```

Figure 40 – Volatility cmdline plugin

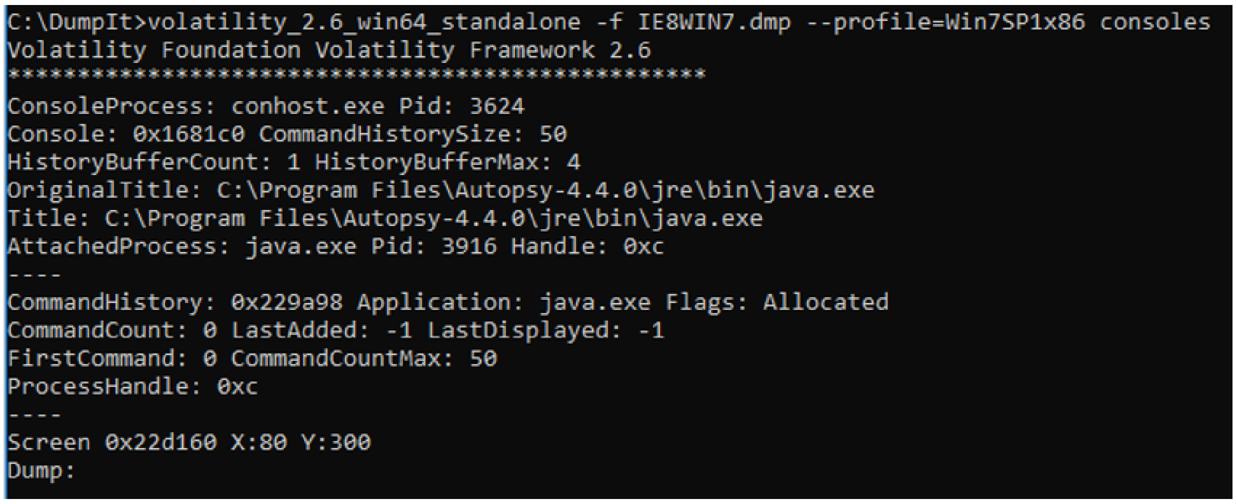
Cmdscan Compare processes (Figure 41)



```
C:\DumpIt>volatility_2.6_win64_standalone -f IE8WIN7.dmp --profile=Win7SP1x86 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: conhost.exe Pid: 3624
CommandHistory: 0x229a98 Application: java.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0xc
**************************************************
CommandProcess: conhost.exe Pid: 4056
CommandHistory: 0x2cd448 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #22 @ 0xff818488: ??3?
Cmd #25 @ 0xff818488: ??3?
Cmd #36 @ 0x2800c4: ,?,?(???(
Cmd #37 @ 0x2cb3b0: ,?(????
```

Figure 41 – Volatility cmdscan plugin

Consoles. Compare processes (Figure 42)



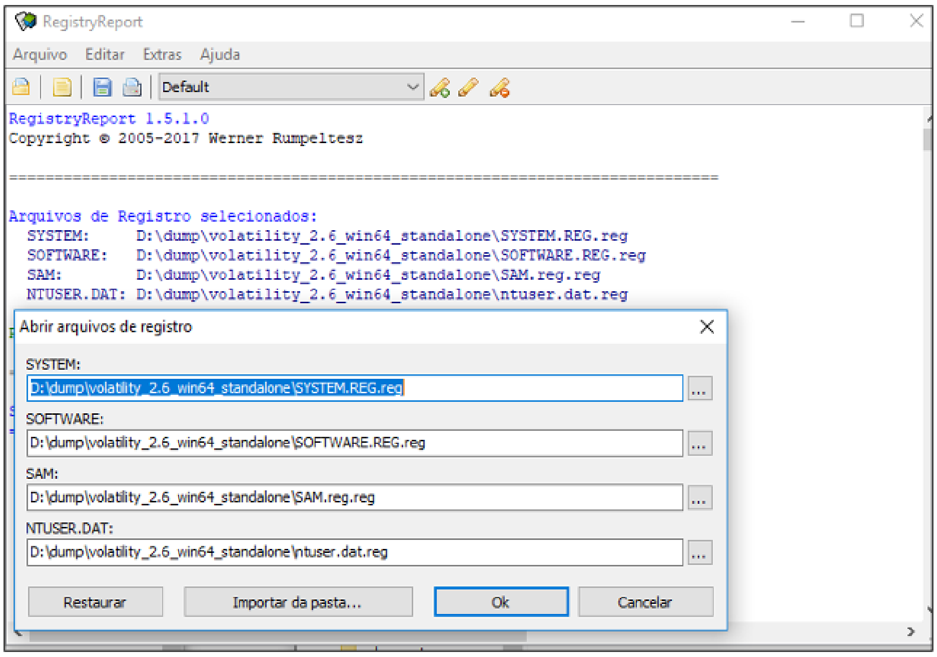
```
C:\DumpIt>volatility_2.6_win64_standalone -f IE8WIN7.dmp --profile=Win7SP1x86 consoles
Volatility Foundation Volatility Framework 2.6
**************************************************
ConsoleProcess: conhost.exe Pid: 3624
Console: 0x1681c0 CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: C:\Program Files\Autopsy-4.4.0\jre\bin\java.exe
Title: C:\Program Files\Autopsy-4.4.0\jre\bin\java.exe
AttachedProcess: java.exe Pid: 3916 Handle: 0xc
----
CommandHistory: 0x229a98 Application: java.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0xc
----
Screen 0x22d160 X:80 Y:300
Dump:
```

Figure 42 – Volatility consoles plugin

Dumpregistry Extract log files

4.2.3. Plugins Volatility – Analysis

With the log files that have been extracted from the memory, it is possible to analyze them with the same tools as the log files extracted from the operating system. An example of this is RegRipper, volatility itself, or RegistryReport, shown in Figure 43.



```
RegistryReport 1.5.1.0
Copyright © 2005-2017 Werner Rumpeltesz

================================================================================

Arquivos de Registro selecionados:
  SYSTEM:      D:\dump\volatility_2.6_win64_standalone\SYSTEM.REG.reg
  SOFTWARE:    D:\dump\volatility_2.6_win64_standalone\SOFTWARE.REG.reg
  SAM:         D:\dump\volatility_2.6_win64_standalone\SAM.reg.reg
  NTUSER.DAT:  D:\dump\volatility_2.6_win64_standalone\ntuser.dat.reg
```

Abrir arquivos de registro

SYSTEM: D:\dump\volatility_2.6_win64_standalone\SYSTEM.REG.reg

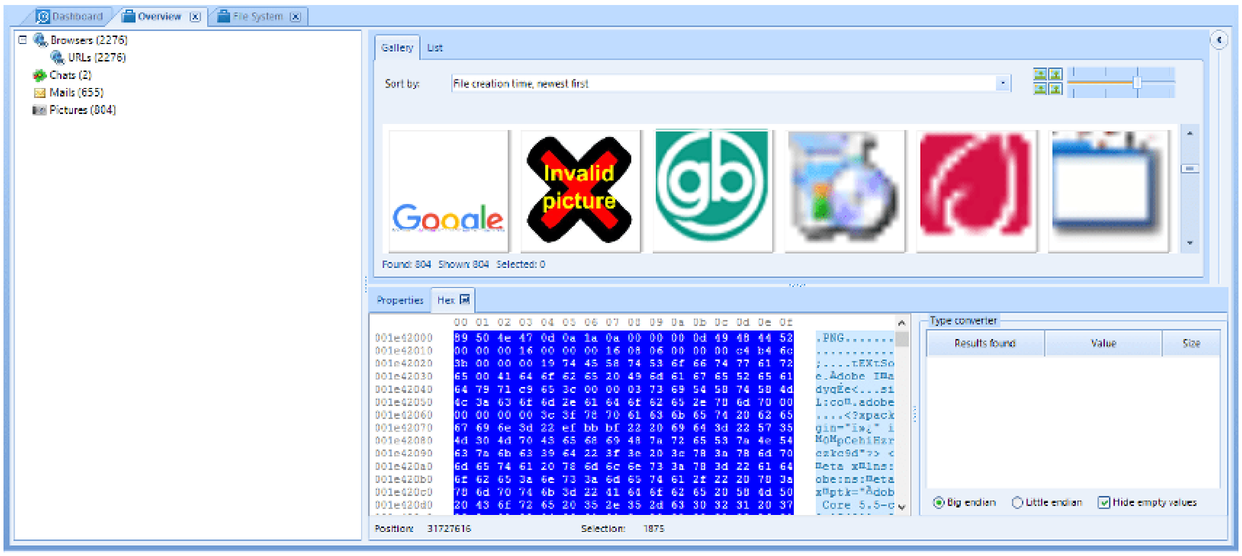
SOFTWARE: D:\dump\volatility_2.6_win64_standalone\SOFTWARE.REG.reg

SAM: D:\dump\volatility_2.6_win64_standalone\SAM.reg.reg

NTUSER.DAT: D:\dump\volatility_2.6_win64_standalone\ntuser.dat.reg

Figure 43 – RegistryReport file analysis

In the analysis of the memory it is also possible to obtain files that have been processed. There are some programs with the possibility of identifying and extracting files from the memory, as shown in the following figure with the Belkasoft software, where it is possible to verify that it identified browsing addresses in browsers, chat conversation data, email files and image files (Figure 44).



Sort by: File creation time, newest first

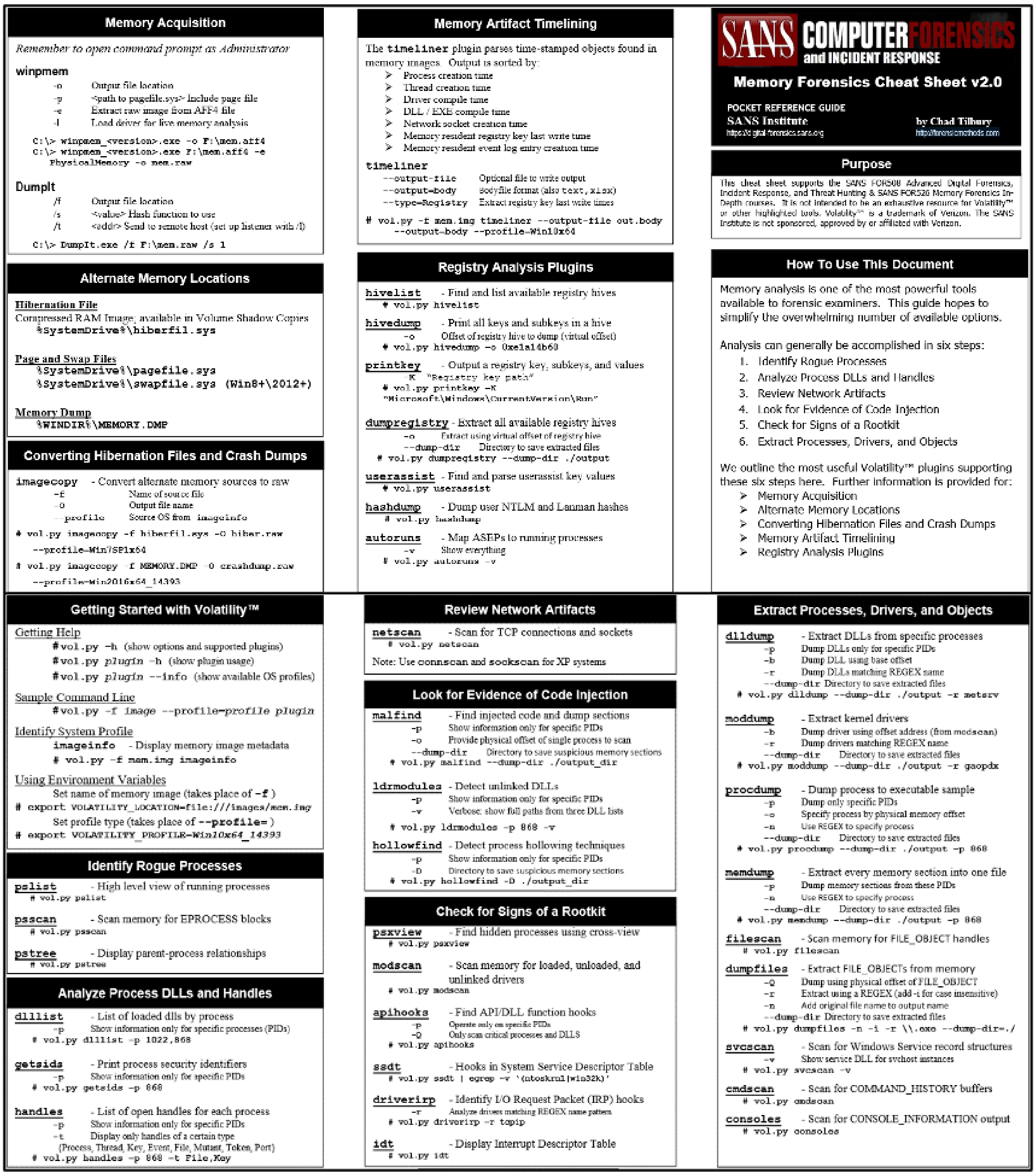
Found: 804 Shown: 804 Selected: 0

```
          00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
001e42000 89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52   .PNG........
001e42010 00 00 00 16 00 00 00 16 08 06 00 00 00 c4 b4 6c   ............
001e42020 3b 00 00 00 19 74 45 58 74 53 6f 66 74 77 61 72   ;....tEXtSo
001e42030 65 00 41 64 6f 62 65 20 49 6d 61 67 65 52 65 61   e.Adobe Ima
001e42040 64 79 71 c9 65 3c 00 00 03 73 69 54 58 74 58 4d   dyqÉe<...si
001e42050 4c 3a 63 6f 6d 2e 61 64 6f 62 65 2e 78 6d 70 00   L:com.adobe
001e42060 00 00 00 00 3c 3f 78 70 61 63 6b 65 74 20 62 65   ....<?xpack
001e42070 67 69 6e 3d 22 ef bb bf 22 20 69 64 3d 22 57 35   gin="ï»¿" i
001e42080 4d 30 4d 70 43 65 68 69 48 7a 72 65 53 7a 4e 54   M0MpCehiHzr
001e42090 63 7a 6b 63 39 64 22 3f 3e 20 3c 78 3a 78 6d 70   czkc9d"?> <
001e420a0 6d 65 74 61 20 78 6d 6c 6e 73 3a 78 3d 22 61 64   meta xmlns:
001e420b0 6f 62 65 3a 6e 73 3a 6d 65 74 61 2f 22 20 78 3a   obe:ns:meta
001e420c0 78 6d 70 74 6b 3d 22 41 64 6f 62 65 20 58 4d 50   xmptk="Adob
001e420d0 20 43 6f 72 65 20 35 2e 35 2d 63 30 32 31 20 37    Core 5.5-c
```

Position: 31727616 Selection: 1875

Figure 44 – File analysis with Belkasoft

SANS has published a Poster (Figure 45) alluding to memory analysis with Volatility, which summarises many of the plugins useful in this type of analysis.



Memory Acquisition

Remember to open command prompt as Administrator

winpmem

```
    -o      Output file location
    -p      <path to pagefile.sys> Include page file
    -e      Extract raw image from AFF4 file
    -l      Load driver for live memory analysis

  C:\> winpmem_<version>.exe -o F:\mem.aff4
  C:\> winpmem_<version>.exe F:\mem.aff4 -e
     PhysicalMemory -o mem.raw
```

DumpIt

```
    /f      Output file location
    /s      <value> Hash function to use
    /t      <addr> Send to remote host (set up listener with /l)

  C:\> DumpIt.exe /f F:\mem.raw /s 1
```

Alternate Memory Locations

<u>Hibernation File</u>
Compressed RAM Image; available in Volume Shadow Copies
`%SystemDrive%\hiberfil.sys`

<u>Page and Swap Files</u>
`%SystemDrive%\pagefile.sys`
`%SystemDrive%\swapfile.sys (Win8+\2012+)`

<u>Memory Dump</u>
`%WINDIR%\MEMORY.DMP`

Converting Hibernation Files and Crash Dumps

```
imagecopy    - Convert alternate memory sources to raw
    -f            Name of source file
    -O            Output file name
    --profile     Source OS from imageinfo
# vol.py imagecopy -f hiberfil.sys -O hiber.raw
   --profile=Win7SP1x64
# vol.py imagecopy -f MEMORY.DMP -O crashdump.raw
   --profile=Win2016x64_14393
```

Getting Started with Volatility™

<u>Getting Help</u>

```
    #vol.py -h  (show options and supported plugins)
    #vol.py plugin -h  (show plugin usage)
    #vol.py plugin --info  (show available OS profiles)
```

<u>Sample Command Line</u>

```
    #vol.py -f image --profile=profile plugin
```

<u>Identify System Profile</u>

```
    imageinfo   - Display memory image metadata
    # vol.py -f mem.img imageinfo
```

<u>Using Environment Variables</u>

```
    Set name of memory image (takes place of -f )
# export VOLATILITY_LOCATION=file:///images/mem.img
    Set profile type (takes place of --profile= )
# export VOLATILITY_PROFILE=Win10x64_14393
```

Identify Rogue Processes

```
pslist       - High level view of running processes
  # vol.py pslist

psscan       - Scan memory for EPROCESS blocks
  # vol.py psscan

pstree       - Display parent-process relationships
  # vol.py pstree
```

Analyze Process DLLs and Handles

```
dlllist      - List of loaded dlls by process
      -p       Show information only for specific processes (PIDs)
  # vol.py dlllist -p 1022,868

getsids      - Print process security identifiers
      -p       Show information only for specific PIDs
  # vol.py getsids -p 868

handles      - List of open handles for each process
      -p       Show information only for specific PIDs
      -t       Display only handles of a certain type
               {Process, Thread, Key, Event, File, Mutant, Token, Port}
  # vol.py handles -p 868 -t File,Key
```

Memory Artifact Timelining

The **timeliner** plugin parses time-stamped objects found in memory images. Output is sorted by:

- Process creation time
- Thread creation time
- Driver compile time
- DLL / EXE compile time
- Network socket creation time
- Memory resident registry key last write time
- Memory resident event log entry creation time

```
timeliner
   --output-file      Optional file to write output
   --output=body      Bodyfile format (also text, xlsx)
   --type=Registry    Extract registry key last write times

# vol.py -f mem.img timeliner --output-file out.body
     --output=body --profile=Win10x64
```

Registry Analysis Plugins

```
hivelist     - Find and list available registry hives
  # vol.py hivelist

hivedump     - Print all keys and subkeys in a hive
      -o       Offset of registry hive to dump (virtual offset)
  # vol.py hivedump -o 0xe1a14b60

printkey     - Output a registry key, subkeys, and values
      -K  "Registry key path"
  # vol.py printkey -K
  "Microsoft\Windows\CurrentVersion\Run"

dumpregistry - Extract all available registry hives
      -o       Extract using virtual offset of registry hive
      --dump-dir    Directory to save extracted files
 # vol.py dumpregistry --dump-dir ./output

userassist   - Find and parse userassist key values
  # vol.py userassist

hashdump     - Dump user NTLM and Lanman hashes
  # vol.py hashdump

autoruns     - Map ASEPs to running processes
      -v       Show everything
  # vol.py autoruns -v
```

Review Network Artifacts

```
netscan      - Scan for TCP connections and sockets
  # vol.py netscan
```

Note: Use **connscan** and **sockscan** for XP systems

Look for Evidence of Code Injection

```
malfind      - Find injected code and dump sections
      -p       Show information only for specific PIDs
      -o       Provide physical offset of single process to scan
      --dump-dir    Directory to save suspicious memory sections
  # vol.py malfind --dump-dir ./output_dir

ldrmodules   - Detect unlinked DLLs
      -p       Show information only for specific PIDs
      -v       Verbose: show full paths from three DLL lists
  # vol.py ldrmodules -p 868 -v

hollowfind   - Detect process hollowing techniques
      -p       Show information only for specific PIDs
      -D       Directory to save suspicious memory sections
  # vol.py hollowfind -D ./output_dir
```

Check for Signs of a Rootkit

```
psxview      - Find hidden processes using cross-view
  # vol.py psxview

modscan      - Scan memory for loaded, unloaded, and
               unlinked drivers
  # vol.py modscan

apihooks     - Find API/DLL function hooks
      -p       Operate only on specific PIDs
      -Q       Only scan critical processes and DLLS
  # vol.py apihooks

ssdt         - Hooks in System Service Descriptor Table
  # vol.py ssdt | egrep -v '(ntoskrnl|win32k)'

driverirp    - Identify I/O Request Packet (IRP) hooks
      -r       Analyze drivers matching REGEX name pattern
  # vol.py driverirp -r tcpip

idt          - Display Interrupt Descriptor Table
  # vol.py idt
```

SANS COMPUTER FORENSICS and INCIDENT RESPONSE

Memory Forensics Cheat Sheet v2.0

POCKET REFERENCE GUIDE
SANS Institute
https://digital-forensics.sans.org

by Chad Tilbury
http://forensicmethods.com

Purpose

This cheat sheet supports the SANS FOR508 Advanced Digital Forensics, Incident Response, and Threat Hunting & SANS FOR526 Memory Forensics In-Depth courses. It is not intended to be an exhaustive resource for Volatility™ or other highlighted tools. Volatility™ is a trademark of Verizon. The SANS Institute is not sponsored, approved by or affiliated with Verizon.

How To Use This Document

Memory analysis is one of the most powerful tools available to forensic examiners. This guide hopes to simplify the overwhelming number of available options.

Analysis can generally be accomplished in six steps:

1. Identify Rogue Processes
2. Analyze Process DLLs and Handles
3. Review Network Artifacts
4. Look for Evidence of Code Injection
5. Check for Signs of a Rootkit
6. Extract Processes, Drivers, and Objects

We outline the most useful Volatility™ plugins supporting these six steps here. Further information is provided for:

- Memory Acquisition
- Alternate Memory Locations
- Converting Hibernation Files and Crash Dumps
- Memory Artifact Timelining
- Registry Analysis Plugins

Extract Processes, Drivers, and Objects

```
dlldump      - Extract DLLs from specific processes
      -p       Dump DLLs only for specific PIDs
      -b       Dump DLL using base offset
      -r       Dump DLLs matching REGEX name
      --dump-dir Directory to save extracted files
 # vol.py dlldump --dump-dir ./output -r metsrv

moddump      - Extract kernel drivers
      -b       Dump driver using offset address (from modscan)
      -r       Dump drivers matching REGEX name
      --dump-dir    Directory to save extracted files
 # vol.py moddump --dump-dir ./output -r gaopdx

procdump     - Dump process to executable sample
      -p       Dump only specific PIDs
      -o       Specify process by physical memory offset
      -n       Use REGEX to specify process
      --dump-dir    Directory to save extracted files
 # vol.py procdump --dump-dir ./output -p 868

memdump      - Extract every memory section into one file
      -p       Dump memory sections from these PIDs
      -n       Use REGEX to specify process
      --dump-dir    Directory to save extracted files
 # vol.py memdump --dump-dir ./output -p 868

filescan     - Scan memory for FILE_OBJECT handles
  # vol.py filescan

dumpfiles    - Extract FILE_OBJECTs from memory
      -Q       Dump using physical offset of FILE_OBJECT
      -r       Extract using a REGEX (add -i for case insensitive)
      -n       Add original file name to output name
      --dump-dir Directory to save extracted files
  # vol.py dumpfiles -n -i -r \\.exe --dump-dir=./

svcscan      - Scan for Windows Service record structures
      -v       Show service DLL for svchost instances
  # vol.py svcscan -v

cmdscan      - Scan for COMMAND_HISTORY buffers
  # vol.py cmdscan

consoles     - Scan for CONSOLE_INFORMATION output
  # vol.py consoles
```

Figure 45 – SANS Poster - Memory Forensics Cheat Sheet v2.0

Create a TimeLine of events in memory

With the data extracted from volatile memory it is useful to create a Timeline, to make it possible to date and sort the indications in the system. This is a process involving the procedures described below:

Timeliner   create a timeline

C:\> volatility_2.6_win64_standalone.exe -f IE8WIN7.dmp --profile=Win7SP1x86_23418 timeliner -- output=body > timeliner.body

Read more: https://volatility-labs.blogspot.com/2013/05/movp-ii-23-creating-timelines-with.html

Mftparser   Obt (Master File Table).

C:\> volatility_2.6_win64_standalone.exe -f IE8WIN7.dmp --profile=Win7SP1x86_23418 mftparser -- output=body > mftparser.body

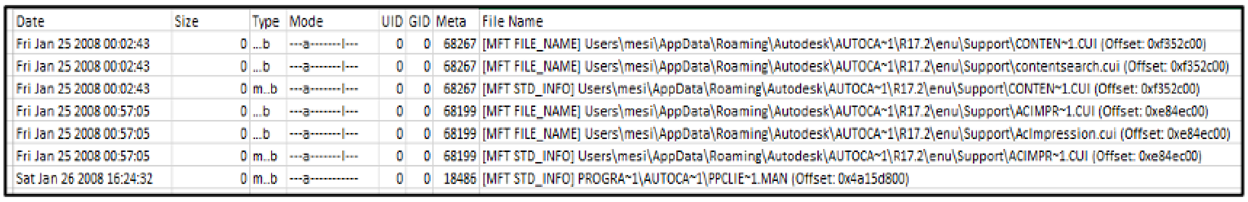
Combine the files concerning the **timeliner** and **mftparser** plugins**.**

# cat timeliner.body mftparser.body >> timeline.log

Mactime[1]   Generate the timeline from the combination of files

# mactime -d -b timeline.log > timeline.csv

Final result of the TimeLine procedures (Figure 46)



Date	Size	Type	Mode	UID	GID	Meta	File Name
Fri Jan 25 2008 00:02:43	0	...b	---a--------|---	0	0	68267	[MFT FILE_NAME] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\CONTEN~1.CUI (Offset: 0xf352c00)
Fri Jan 25 2008 00:02:43	0	...b	---a--------|---	0	0	68267	[MFT FILE_NAME] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\contentsearch.cui (Offset: 0xf352c00)
Fri Jan 25 2008 00:02:43	0	m..b	---a--------|---	0	0	68267	[MFT STD_INFO] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\CONTEN~1.CUI (Offset: 0xf352c00)
Fri Jan 25 2008 00:57:05	0	...b	---a--------|---	0	0	68199	[MFT FILE_NAME] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\ACIMPR~1.CUI (Offset: 0xe84ec00)
Fri Jan 25 2008 00:57:05	0	...b	---a--------|---	0	0	68199	[MFT FILE_NAME] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\AcImpression.cui (Offset: 0xe84ec00)
Fri Jan 25 2008 00:57:05	0	m..b	---a--------|---	0	0	68199	[MFT STD_INFO] Users\mesi\AppData\Roaming\Autodesk\AUTOCA~1\R17.2\enu\Support\ACIMPR~1.CUI (Offset: 0xe84ec00)
Sat Jan 26 2008 16:24:32	0	m..b	---a------------	0	0	18486	[MFT STD_INFO] PROGRA~1\AUTOCA~1\PPCLIE~1.MAN (Offset: 0x4a15d800)

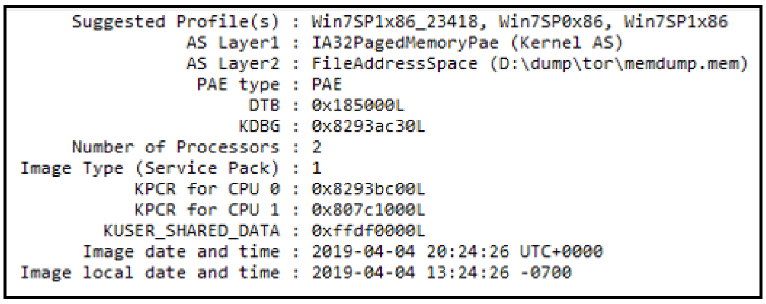
Figure 46 – Content of timeline.csv

With this table it is possible to easily identify the actions passed in the device memory, and these will complement the information obtained in the device analysis in dead-box forensics.

Example of identification of access to the TOR network

As an exemple of memory data analysis, we have the use of Tor Browser, since it does not store navigation information on the hard disk, although it is possible to identify and analyse it through memory.

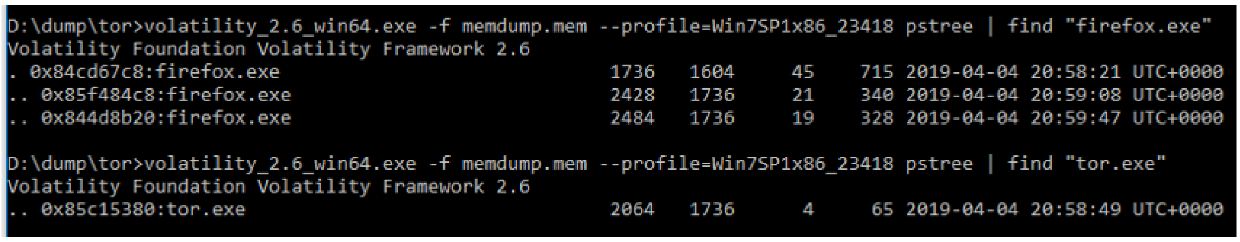
We start by confirming the **system profile** (Figure 47).



```
     Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
                AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                AS Layer2 : FileAddressSpace (D:\dump\tor\memdump.mem)
                 PAE type : PAE
                      DTB : 0x185000L
                     KDBG : 0x8293ac30L
     Number of Processors : 2
Image Type (Service Pack) : 1
           KPCR for CPU 0 : 0x8293bc00L
           KPCR for CPU 1 : 0x807c1000L
        KUSER_SHARED_DATA : 0xffdf0000L
      Image date and time : 2019-04-04 20:24:26 UTC+0000
Image local date and time : 2019-04-04 13:24:26 -0700
```

Figure 47 – Volatility imageinfo plugin

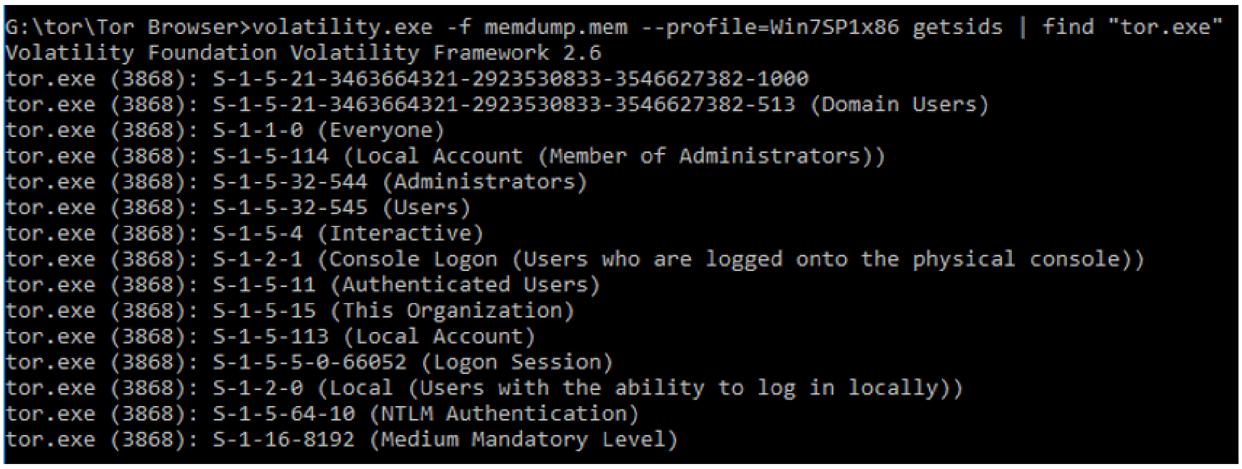
We used the **pstree** plugin to check the running processes, filtering the processes by the name "firefox.exe" (Figure 48), since Tor Browser uses this process, or directly by the name "tor.exe". To get more information about the processes in the device under analysis, we still have the possibility to use the **pslist**, **psscan** plugins.



```
D:\dump\tor>volatility_2.6_win64.exe -f memdump.mem --profile=Win7SP1x86_23418 pstree | find "firefox.exe"
Volatility Foundation Volatility Framework 2.6
. 0x84cd67c8:firefox.exe                            1736   1604     45    715 2019-04-04 20:58:21 UTC+0000
.. 0x85f484c8:firefox.exe                           2428   1736     21    340 2019-04-04 20:59:08 UTC+0000
.. 0x844d8b20:firefox.exe                           2484   1736     19    328 2019-04-04 20:59:47 UTC+0000

D:\dump\tor>volatility_2.6_win64.exe -f memdump.mem --profile=Win7SP1x86_23418 pstree | find "tor.exe"
Volatility Foundation Volatility Framework 2.6
.. 0x85c15380:tor.exe                               2064   1736      4     65 2019-04-04 20:58:49 UTC+0000
```

Figure 48 – Using volatility in process research

Getsids   Information about the start of the process, relating the process to the user (Figure 49).

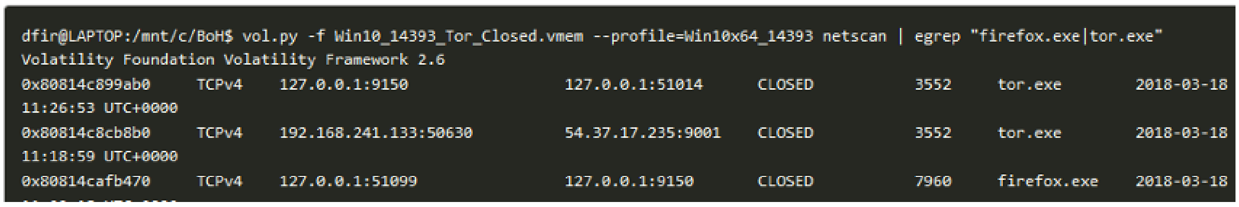


```
G:\tor\Tor Browser>volatility.exe -f memdump.mem --profile=Win7SP1x86 getsids | find "tor.exe"
Volatility Foundation Volatility Framework 2.6
tor.exe (3868): S-1-5-21-3463664321-2923530833-3546627382-1000
tor.exe (3868): S-1-5-21-3463664321-2923530833-3546627382-513 (Domain Users)
tor.exe (3868): S-1-1-0 (Everyone)
tor.exe (3868): S-1-5-114 (Local Account (Member of Administrators))
tor.exe (3868): S-1-5-32-544 (Administrators)
tor.exe (3868): S-1-5-32-545 (Users)
tor.exe (3868): S-1-5-4 (Interactive)
tor.exe (3868): S-1-2-1 (Console Logon (Users who are logged onto the physical console))
tor.exe (3868): S-1-5-11 (Authenticated Users)
tor.exe (3868): S-1-5-15 (This Organization)
tor.exe (3868): S-1-5-113 (Local Account)
tor.exe (3868): S-1-5-5-0-66052 (Logon Session)
tor.exe (3868): S-1-2-0 (Local (Users with the ability to log in locally))
tor.exe (3868): S-1-5-64-10 (NTLM Authentication)
tor.exe (3868): S-1-16-8192 (Medium Mandatory Level)
```

Figure 49 – Using volatility in process identification

netscan display the network connections

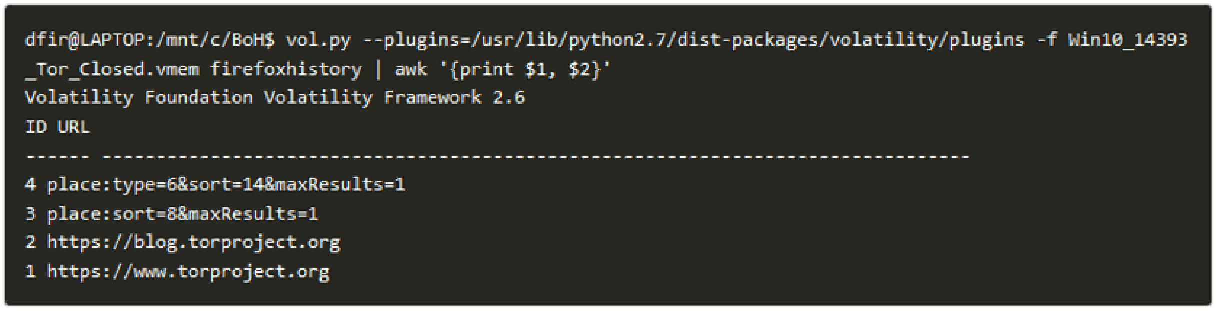
In this case, the “tor.exe” process indicates a completed connection to the destination IP “54.37.17.235” on port 9001 (Figure 50).



```
dfir@LAPTOP:/mnt/c/BoH$ vol.py -f Win10_14393_Tor_Closed.vmem --profile=Win10x64_14393 netscan | egrep "firefox.exe|tor.exe"
Volatility Foundation Volatility Framework 2.6
0x80814c899ab0     TCPv4    127.0.0.1:9150               127.0.0.1:51014      CLOSED         3552     tor.exe       2018-03-18
11:26:53 UTC+0000
0x80814c8cb8b0     TCPv4    192.168.241.133:50630        54.37.17.235:9001    CLOSED         3552     tor.exe       2018-03-18
11:18:59 UTC+0000
0x80814cafb470     TCPv4    127.0.0.1:51099              127.0.0.1:9150       CLOSED         7960     firefox.exe   2018-03-18
```

Figure 50 – Using Volatility in network identification

Firefoxhistory List the addresses (URLs) queried (Figure 51).



```
dfir@LAPTOP:/mnt/c/BoH$ vol.py --plugins=/usr/lib/python2.7/dist-packages/volatility/plugins -f Win10_14393
_Tor_Closed.vmem firefoxhistory | awk '{print $1, $2}'
Volatility Foundation Volatility Framework 2.6
ID URL
------ --------------------------------------------------------------------------------
4 place:type=6&sort=14&maxResults=1
3 place:sort=8&maxResults=1
2 https://blog.torproject.org
1 https://www.torproject.org
```

Figure 51 – Using volatility to retrieve URLs from memory

Source: https://blog.superponible.com/2014/08/31/volatility-plugin-firefox-history/

[1] https://wiki.sleuthkit.org/index.php?title=Mactime

5. Identification and analysis of information in Operating Systems

According to “https://gs.statcounter.com/os-market-share/desktop/worldwide”, the Microsoft Windows operating system has a market share of about 75 per cent, followed by Apple OS X with 14.5 per cent. A forensic analyst will find considerably more devices with Micrososft Operating systems than any other. This justifies the greater attention given to the analysis of this operating system.

SANS has carried out excellent research and education work in the area of digital forensics, one of the factors of interest being the regular publication of the so-called Poster, at: https://www.sans.org/posters/, with the result of that research. These Posters are also important sources of information regarding the location of artefacts of forensic interest, as MS Windows operating systems store numerous artefacts in the daily actions of its users.

5.1. MS Windows Registry

"*A central hierarchical database in Windows ... used to store information necessary to configure the system for one or more users, applications, and hardware devices. The Registry contains information that Windows continually references during operation, such as profiles for each user, the applications installed on the computer and the types of documents each can create, property sheet settings for folders and application icons, what hardware exists on the system, and which ports are being used.*"

Source: *Microsoft Computer Dictionary.--5th ed.*, Redmond, Washington, Microsoft Press, 2002, p. 445

It is therefore possible to state that the Windows registry, despite its structuring files, contains a logical structure in constant use by the operating system, storing a set of information necessary for its operation.

The logical structure of the Windows registry contains:

1. **Registry keys**, keys named "Software" and "System", belonging to the *hive* "HKEY_CURRENT_CONFIG".
2. **Registry sub-keys**, where registry information is stored (e.g.: "Fonts" sub-key).
3. **Registry values,** which contain the information by specifying their type in the respective column (e.g.: REG_DWORD - 32-bit binary value, REG_QWORD – 64-bit binary value).

The five main Hives in the logical structure of the MS Windows operating system can be seen in Figure 52.

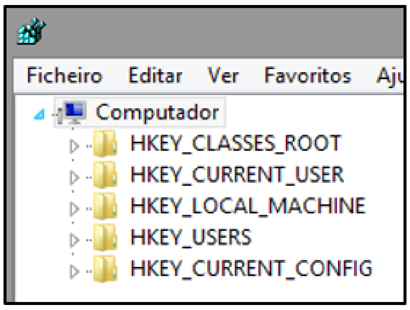


Figure 52 - Root Hive

Hive of the registry (Root Keys) are characterized by the prefix "HKEY_", short for "Handle to a KEY".

There are 5 main hives, stored in the various files that make up the registry, although only HKEY_USERS and HKEY_LOCAL_MACHINE are considered the true hives, the rest being shortcuts or aliases for branches within them.

Hive	**Abbreviation**	**Description**	***Link***
HKEY_CURRENT_USER	**HKCU**	Points to the user profile of the currently loggedon user	Subkey under HKEY_USERS corresponding to currently loggedon user
HKEY_USERS	**HKU**	Contains subkeys for all loaded user profiles	Not a link
HKEY_CLASSES_ROOT	**HKCR**	Contains file association and COM registration information	Not a direct link; rather, a merged view of HKLM\SOFTWARE\Classes and HKEY_CURRENT_USER\SOFTWARE\Classes
HKEY_LOCAL_MACHINE	**HKLM**	Global settings for the machine.	Not a link

HKEY_CURRENT_CONFIG	**HKCC**	Current hardware profile	HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current
HKEY_PERFORMANCE_DATA	**HKPD**	Performance counters	Not a link

Source: *Windows Internals.--6th ed., Part 1*, Redmond, Washington, Microsoft Press, 2012, p. 281

The log files are located in the following folders:

Operating system log files

C:\Windows\System32\Config\

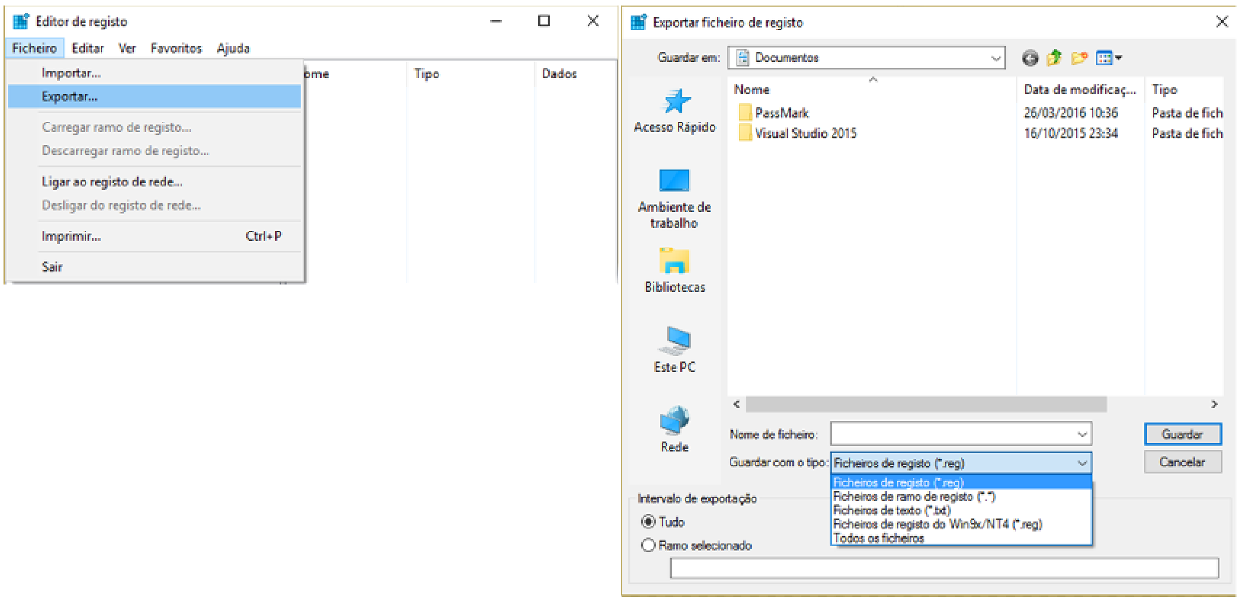
Registry files for each user

C:\Users\<username>\ntuser.dat

5.1.1. Registry editor (RegEdit)

The registry editor, in its graphical version, allows the export of one or more registry keys (Figure 53).

RegEdit, File > Export


Editor de registo
Ficheiro
Editar
Ver
Favoritos
Ajuda
Importar...
Exportar...
Carregar ramo de registo...
Descarregar ramo de registo...
Ligar ao registo de rede...
Desligar do registo de rede...
Imprimir... Ctrl+P
Sair
ome
Tipo
Dados
Exportar ficheiro de registo
Acesso Rápido
Ambiente de trabalho
Bibliotecas
Este PC
Rede
Nome
Data de modificaç...
Tipo
PassMark 26/03/2016 10:36 Pasta de fich
Visual Studio 2015 16/10/2015 23:34 Pasta de fich
Nome de ficheiro:
Guardar
Cancelar
Ficheiros de registo (*.reg)
Ficheiros de ramo de registo (*.*)
Ficheiros de texto (*.txt)
Ficheiros de registo do Win9x/NT4 (*.reg)
Todos os ficheiros
Ramo selecionado

Guardar em: Documentos

Guardar com o tipo: Ficheiros de registo (*.reg)

Intervalo de exportação: Tudo

Figure 53 – Registry Export through RegEdit

Via the command line:

```
regedit /e c:\output.reg "HKEY_LOCAL_MACHINE\System\..."
```

5.1.2. ERUNTgui

The application ERUNTgui (Figure 54), allows the backup, restoration and optimization of the registry, being of forensic interest the possibility of performing the backup of the registry, thus enabling its subsequent analysis.

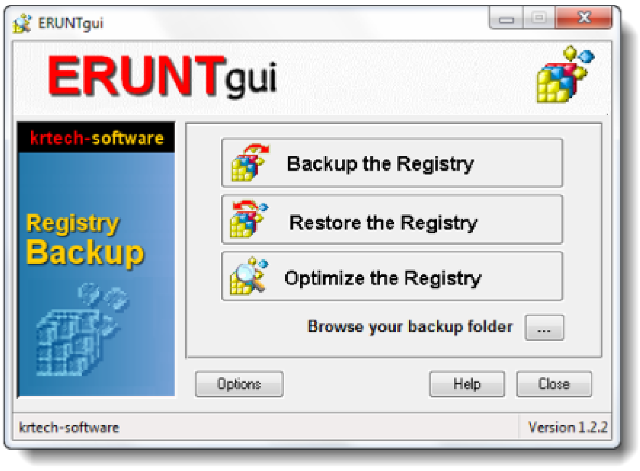
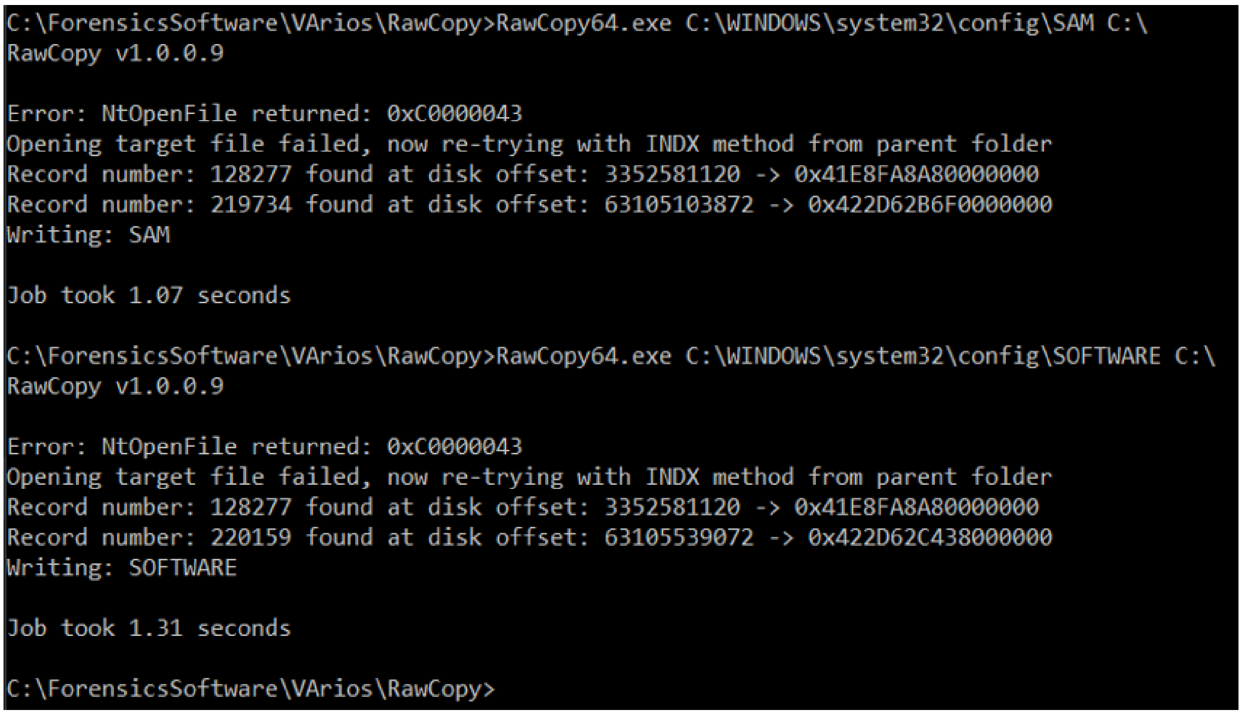


Figure 54 – Registry Export through ERUNTgui

5.1.3. RAWCopy

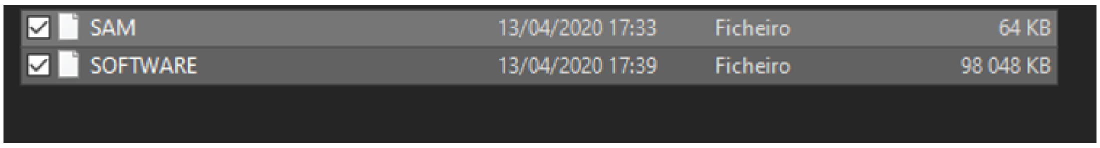
The RAWCopy application (Figure 55), allows copying the sectors of the disk where the files in use are, thus overcoming the limitation of copying files opened by the system.



```
C:\ForensicsSoftware\VArios\RawCopy>RawCopy64.exe C:\WINDOWS\system32\config\SAM C:\
RawCopy v1.0.0.9

Error: NtOpenFile returned: 0xC0000043
Opening target file failed, now re-trying with INDX method from parent folder
Record number: 128277 found at disk offset: 3352581120 -> 0x41E8FA8A80000000
Record number: 219734 found at disk offset: 63105103872 -> 0x422D62B6F0000000
Writing: SAM

Job took 1.07 seconds

C:\ForensicsSoftware\VArios\RawCopy>RawCopy64.exe C:\WINDOWS\system32\config\SOFTWARE C:\
RawCopy v1.0.0.9

Error: NtOpenFile returned: 0xC0000043
Opening target file failed, now re-trying with INDX method from parent folder
Record number: 128277 found at disk offset: 3352581120 -> 0x41E8FA8A80000000
Record number: 220159 found at disk offset: 63105539072 -> 0x422D62C438000000
Writing: SOFTWARE

Job took 1.31 seconds

C:\ForensicsSoftware\VArios\RawCopy>
```

Figure 55 – Registry Export through RAWCopy

Through RAWCopy it was possible to obtain a copy of the SAM and SOFTWARE file with the system running (Figure 56).



✓	SAM	13/04/2020 17:33	Ficheiro	64 KB
✓	SOFTWARE	13/04/2020 17:39	Ficheiro	98 048 KB

Figure 56 – Files exported by RAWCopy

Source: https://github.com/jschicht/RawCopy

5.2. Windows Registry Analysis

Windows registry analysis can be performed using forensic software, such as AccessData Registry Viewer, Eric Zimmerman's tools, RegRipper, or even any other software capable of extracting data from these registry files.

Due to the complexity of the Windows registry, identifying the location of each relevant piece of information can be a daunting task, however, we have enlisted the help of SANS FOR500 (https://digital-forensics.sans.org/docs/DFPS_FOR500_v4.11_0121.pdf), identifying many of the important locations where relevant information can be found.

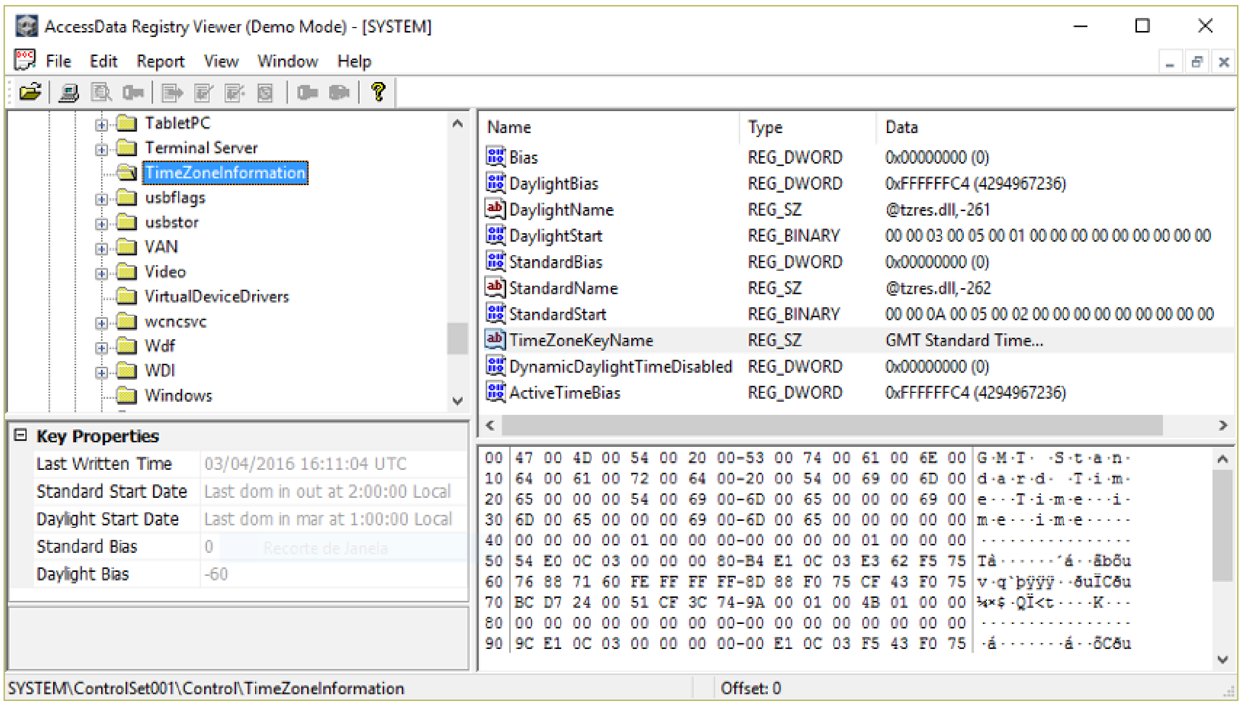
There may be a need to extract the log from a connected computer, the log being a huge source of forensically relevant information, it will be imperative to obtain all that information. (Read: https://resources.infosecinstitute.com/windows-registry-analysis-regripper-hands-case-study-2/). There are several ways to perform the registry dump, here we will focus on some different ways.

5.2.1. The Time Zone

The first information to be analysed should include the Time Zone (Figure 57), as this may lead us to make mistakes when faced with actions that present a date/time different from the true one, simply because the system is configured with a time zone different from the one that the forensic analyst is using.

This information can be identified in *hive* SYSTEM, at the following location :

SYSTEM\ControlSet001\Control\TimeZoneInformation



Name	Type	Data
Bias	REG_DWORD	0x00000000 (0)
DaylightBias	REG_DWORD	0xFFFFFFC4 (4294967236)
DaylightName	REG_SZ	@tzres.dll,-261
DaylightStart	REG_BINARY	00 00 03 00 05 00 01 00 00 00 00 00 00 00 00 00
StandardBias	REG_DWORD	0x00000000 (0)
StandardName	REG_SZ	@tzres.dll,-262
StandardStart	REG_BINARY	00 00 0A 00 05 00 02 00 00 00 00 00 00 00 00 00
TimeZoneKeyName	REG_SZ	GMT Standard Time...
DynamicDaylightTimeDisabled	REG_DWORD	0x00000000 (0)
ActiveTimeBias	REG_DWORD	0xFFFFFFC4 (4294967236)

Key Properties

Last Written Time	03/04/2016 16:11:04 UTC
Standard Start Date	Last dom in out at 2:00:00 Local
Daylight Start Date	Last dom in mar at 1:00:00 Local
Standard Bias	0
Daylight Bias	-60

```
00 47 00 4D 00 54 00 20 00-53 00 74 00 61 00 6E 00 G·M·T· ·S·t·a·n·
10 64 00 61 00 72 00 64 00-20 00 54 00 69 00 6D 00 d·a·r·d· ·T·i·m·
20 65 00 00 00 54 00 69 00-6D 00 65 00 00 00 69 00 e···T·i·m·e···i·
30 6D 00 65 00 00 00 69 00-6D 00 65 00 00 00 00 00 m·e···i·m·e·····
40 00 00 00 00 01 00 00 00-00 00 00 00 01 00 00 00 ················
50 54 E0 0C 03 00 00 00 80-B4 E1 0C 03 E3 62 F5 75 Tà······´á··ãbõu
60 76 88 71 60 FE FF FF FF-8D 88 F0 75 CF 43 F0 75 v·q`þÿÿÿ··ðuÏCðu
70 BC D7 24 00 51 CF 3C 74-9A 00 01 00 4B 01 00 00 ¼×$·QÏ<t····K···
80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
90 9C E1 0C 03 00 00 00 00-00 E1 0C 03 F5 43 F0 75 ·á·······á··õCðu
```

SYSTEM\ControlSet001\Control\TimeZoneInformation Offset: 0

Figure 57 – TimeZone in AccessData Registry Viewerr

5.2.2. USB devices

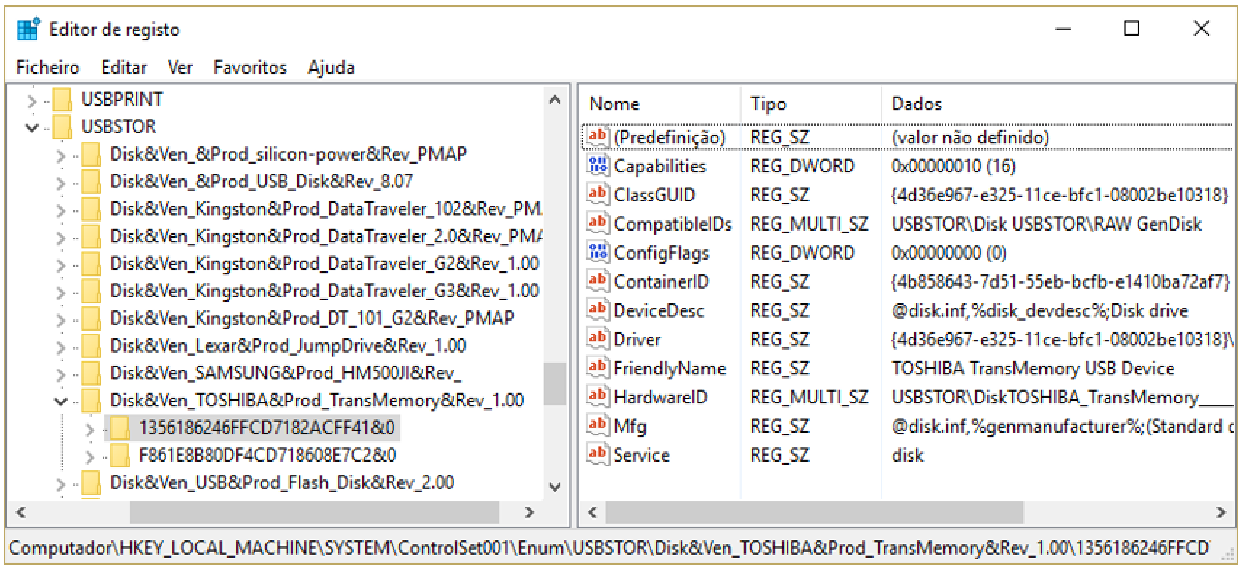
In the registry it is also possible to obtain information from USB devices that have connected to the system in the *hive* SYSTEM:

To obtain: Producer / Brand / serial no. / date / time of first and last connection to the system

HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR

HKLM\SYSTEM\CurrentControlSet\Enum\USB

By viewing (Figure 58).



```
USBPRINT
USBSTOR
  Disk&Ven_&Prod_silicon-power&Rev_PMAP
  Disk&Ven_&Prod_USB_Disk&Rev_8.07
  Disk&Ven_Kingston&Prod_DataTraveler_102&Rev_PM.
  Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_PMA
  Disk&Ven_Kingston&Prod_DataTraveler_G2&Rev_1.00
  Disk&Ven_Kingston&Prod_DataTraveler_G3&Rev_1.00
  Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP
  Disk&Ven_Lexar&Prod_JumpDrive&Rev_1.00
  Disk&Ven_SAMSUNG&Prod_HM500JI&Rev_
  Disk&Ven_TOSHIBA&Prod_TransMemory&Rev_1.00
    1356186246FFCD7182ACFF41&0
    F861E8B80DF4CD718608E7C2&0
  Disk&Ven_USB&Prod_Flash_Disk&Rev_2.00
```

Nome	Tipo	Dados
(Predefinição)	REG_SZ	(valor não definido)
Capabilities	REG_DWORD	0x00000010 (16)
ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW GenDisk
ConfigFlags	REG_DWORD	0x00000000 (0)
ContainerID	REG_SZ	{4b858643-7d51-55eb-bcfb-e1410ba72af7}
DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\
FriendlyName	REG_SZ	TOSHIBA TransMemory USB Device
HardwareID	REG_MULTI_SZ	USBSTOR\DiskTOSHIBA_TransMemory___
Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard d
Service	REG_SZ	disk

Computador\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\USBSTOR\Disk&Ven_TOSHIBA&Prod_TransMemory&Rev_1.00\1356186246FFCD

Figure 58 - Viewing USB devices in the Registry Editor

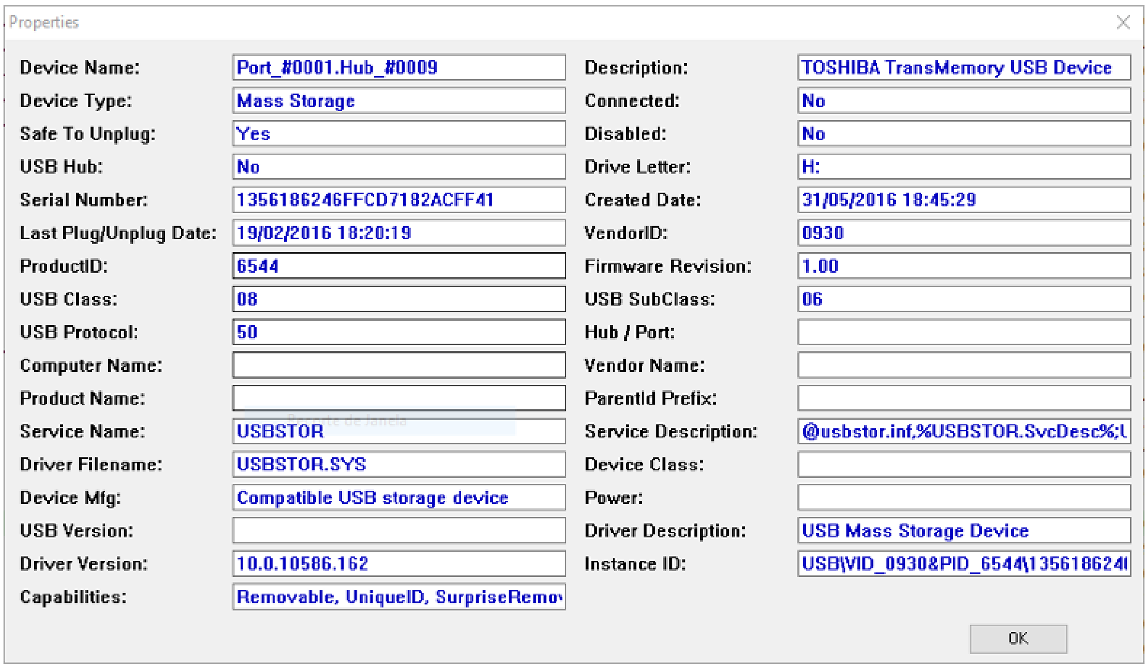
To get the letter assigned to the USB device

HKLM\SYSTEM\MountedDevices

To get the user who connected the device to the system

NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

The same information can be obtained by using specific tools to obtain information from USB devices, such as the 4Discovery or USBDeview tools(Figura 59).



Device Name:	Port_#0001.Hub_#0009	Description:	TOSHIBA TransMemory USB Device
Device Type:	Mass Storage	Connected:	No
Safe To Unplug:	Yes	Disabled:	No
USB Hub:	No	Drive Letter:	H:
Serial Number:	1356186246FFCD7182ACFF41	Created Date:	31/05/2016 18:45:29
Last Plug/Unplug Date:	19/02/2016 18:20:19	VendorID:	0930
ProductID:	6544	Firmware Revision:	1.00
USB Class:	08	USB SubClass:	06
USB Protocol:	50	Hub / Port:	
Computer Name:		Vendor Name:	
Product Name:		ParentId Prefix:	
Service Name:	USBSTOR	Service Description:	@usbstor.inf,%USBSTOR.SvcDesc%;l
Driver Filename:	USBSTOR.SYS	Device Class:	
Device Mfg:	Compatible USB storage device	Power:	
USB Version:		Driver Description:	USB Mass Storage Device
Driver Version:	10.0.10586.162	Instance ID:	USB\VID_0930&PID_6544\135618624
Capabilities:	Removable, UniqueID, SurpriseRemo		

Figure 59 – Viewing USB devices in USBDeview

Read:https://www.researchgate.net/publication/318514858_USB_Storage_Device_Forensics_for_Windows_10

5.2.3. Users

The information about the system users is stored in the Windows registry in the SAM hive, but there is also the NTUSER.DAT registry file for each user, which stores data specific to that user:

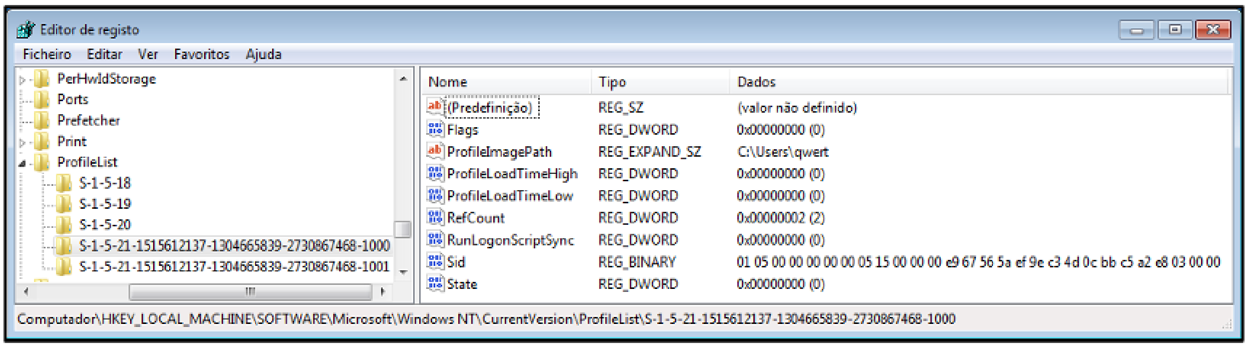
List of local user profiles

In the registry: HKLM\Software\Microsoft\Windows NT\CurrentVersion\ProfileList

System users

In the file: SAM\SAM\Domains\Account\Users\

Viewing this registry key through the Registry Editor, you can see how this information is displayed (Figure 60).

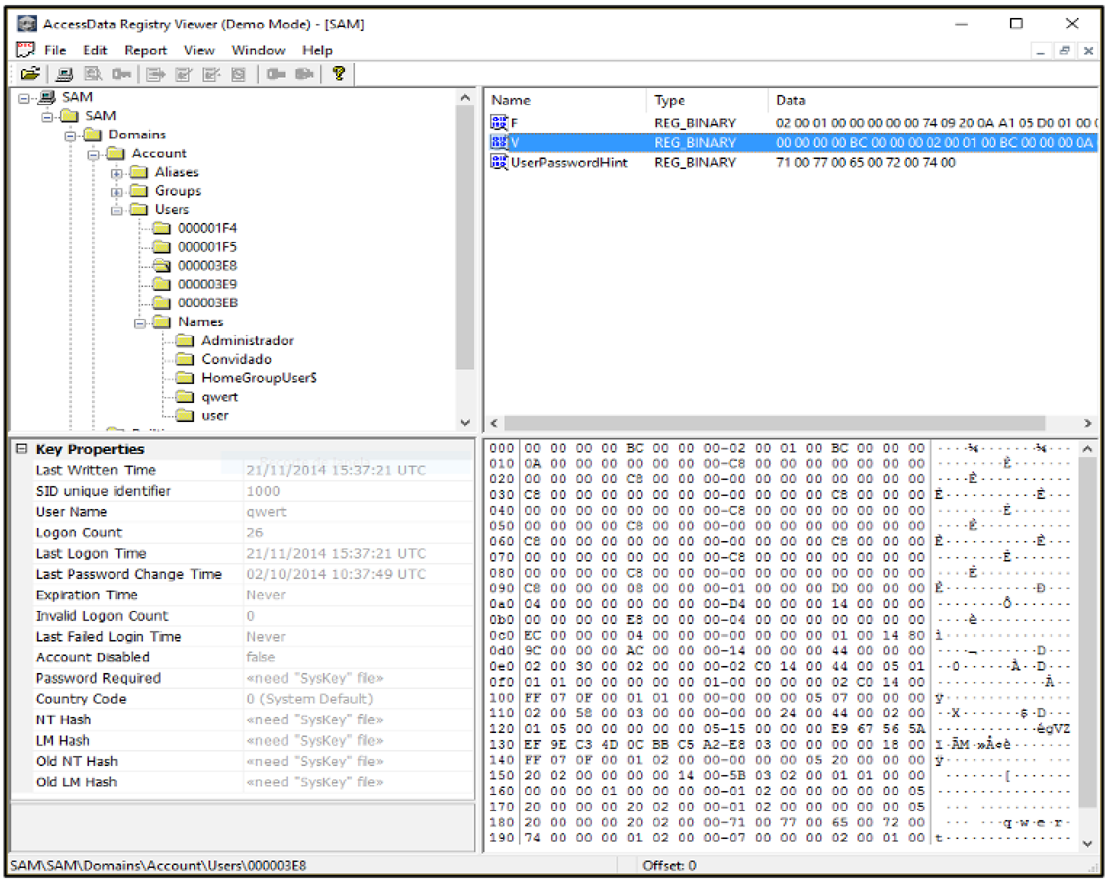

Editor de registo
Ficheiro Editar Ver Favoritos Ajuda
PerHwIdStorage
Ports
Prefetcher
Print
ProfileList
S-1-5-18
S-1-5-19
S-1-5-20
S-1-5-21-1515612137-1304665839-2730867468-1000
S-1-5-21-1515612137-1304665839-2730867468-1001

Nome	Tipo	Dados
(Predefinição)	REG_SZ	(valor não definido)
Flags	REG_DWORD	0x00000000 (0)
ProfileImagePath	REG_EXPAND_SZ	C:\Users\qwert
ProfileLoadTimeHigh	REG_DWORD	0x00000000 (0)
ProfileLoadTimeLow	REG_DWORD	0x00000000 (0)
RefCount	REG_DWORD	0x00000002 (2)
RunLogonScriptSync	REG_DWORD	0x00000000 (0)
Sid	REG_BINARY	01 05 00 00 00 00 00 05 15 00 00 00 e9 67 56 5a ef 9e c3 4d 0c bb c5 a2 e8 03 00 00
State	REG_DWORD	0x00000000 (0)

Computador\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-21-1515612137-1304665839-2730867468-1000

Figure 60 – Viewing users in the registry editor

The same information can be obtained with specific tools to obtain users' information, such as the AccessData Registry Viewer (Figure 61).



Name	Type	Data
F	REG_BINARY	02 00 01 00 00 00 00 00 74 09 20 0A A1 05 D0 01 00
V	REG_BINARY	00 00 00 00 BC 00 00 00 02 00 01 00 BC 00 00 00 0A
UserPasswordHint	REG_BINARY	71 00 77 00 65 00 72 00 74 00

Key Properties	
Last Written Time	21/11/2014 15:37:21 UTC
SID unique identifier	1000
User Name	qwert
Logon Count	26
Last Logon Time	21/11/2014 15:37:21 UTC
Last Password Change Time	02/10/2014 10:37:49 UTC
Expiration Time	Never
Invalid Logon Count	0
Last Failed Login Time	Never
Account Disabled	false
Password Required	«need "SysKey" file»
Country Code	0 (System Default)
NT Hash	«need "SysKey" file»
LM Hash	«need "SysKey" file»
Old NT Hash	«need "SysKey" file»
Old LM Hash	«need "SysKey" file»

```
000  00 00 00 00 BC 00 00 00-02 00 01 00 BC 00 00 00  ····¼·······¼···
010  0A 00 00 00 00 00 00 00-C8 00 00 00 00 00 00 00  ········È·······
020  00 00 00 00 C8 00 00 00-00 00 00 00 00 00 00 00  ····È···········
030  C8 00 00 00 00 00 00 00-00 00 00 00 C8 00 00 00  È···········È···
040  00 00 00 00 00 00 00 00-C8 00 00 00 00 00 00 00  ········È·······
050  00 00 00 00 C8 00 00 00-00 00 00 00 00 00 00 00  ····È···········
060  C8 00 00 00 00 00 00 00-00 00 00 00 C8 00 00 00  È···········È···
070  00 00 00 00 00 00 00 00-C8 00 00 00 00 00 00 00  ········È·······
080  00 00 00 00 C8 00 00 00-00 00 00 00 00 00 00 00  ····È···········
090  C8 00 00 00 08 00 00 00-01 00 00 00 D0 00 00 00  È···········Ð···
0a0  04 00 00 00 00 00 00 00-D4 00 00 00 14 00 00 00  ········Ô·······
0b0  00 00 00 00 E8 00 00 00-04 00 00 00 00 00 00 00  ····è···········
0c0  EC 00 00 00 04 00 00 00-00 00 00 00 01 00 14 80  ì···············
0d0  9C 00 00 00 AC 00 00 00-14 00 00 00 44 00 00 00  ····¬·······D···
0e0  02 00 30 00 02 00 00 00-02 C0 14 00 44 00 05 01  ··0·····À··D····
0f0  01 01 00 00 00 00 00 01-00 00 00 00 02 C0 14 00  ·············À··
100  FF 07 0F 00 01 01 00 00-00 00 00 05 07 00 00 00  ÿ···············
110  02 00 58 00 03 00 00 00-00 00 24 00 44 00 02 00  ··X·······$·D···
120  01 05 00 00 00 00 00 05-15 00 00 00 E9 67 56 5A  ············égVZ
130  EF 9E C3 4D 0C BB C5 A2-E8 03 00 00 00 00 18 00  ï·ÃM·»Å¢è·······
140  FF 07 0F 00 01 02 00 00-00 00 00 05 20 00 00 00  ÿ···············
150  20 02 00 00 00 00 14 00-5B 03 02 00 01 01 00 00   ·······[·······
160  00 00 00 01 00 00 00 00-01 02 00 00 00 00 00 05  ················
170  20 00 00 00 20 02 00 00-01 02 00 00 00 00 00 05   ···············
180  20 00 00 00 20 02 00 00-71 00 77 00 65 00 72 00   ·······q·w·e·r·
190  74 00 00 00 01 02 00 00-07 00 00 00 02 00 01 00  t···············
```

SAM\SAM\Domains\Account\Users\000003E8   Offset: 0

Figure 61 – Viewing users in AccessData Registry Viewer

5.2.4. Network

The log also contains various network information, such as the wireless networks the system has connected to:

HKLM\Software\Microsoft\Windows NT\CurrentVersion\NetworkList\

In this location it is possible to identify:

- Network name (SSID)
- Name of the domain / intranet
- Date/time of last connection (through the date/time that the respective key was written)

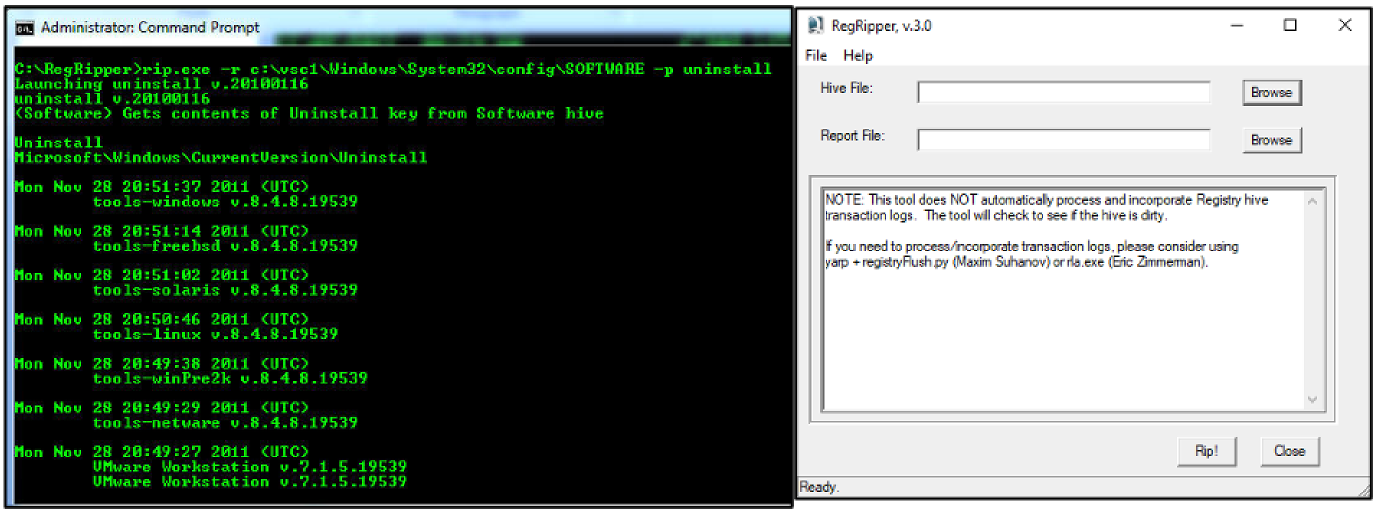
- Gateway MAC address

5.2.5. Windows Registry Analysis - RegRipper

Regarding Windows registry analysis, there are many forensic tools that we can use to facilitate the analysis of information contained in the Windows registry. We focus here on some free tools such as RegRipper, RegistryReport and Windows Registry Recovery.

RegRipper (http://github.com/keydet89) is an Open Source Forensic application, developed by Harlan Carvey and written in PERL, with the objective of extracting information from Windows Registry files in a readable way.

RegRipper (Figure 62) can be used through the command line and a graphic interface to extract specific information from each registry file. When using the command line it is possible to select the plugin to be applied to each registry hive, whereas in the case of the command line all the plugins available for the selected hive are applied. The result of the extracted information may be displayed on screen or saved in a text file, in the case of using the command line. Through its graphical interface it will be necessary to indicate the output location, called Report File, to create the text file with the result of all plugins applied to the respective registry hive..



```
C:\RegRipper>rip.exe -r c:\vsc1\Windows\System32\config\SOFTWARE -p uninstall
Launching uninstall v.20100116
uninstall v.20100116
(Software) Gets contents of Uninstall key from Software hive

Uninstall
Microsoft\Windows\CurrentVersion\Uninstall

Mon Nov 28 20:51:37 2011 (UTC)
        tools-windows v.8.4.8.19539

Mon Nov 28 20:51:14 2011 (UTC)
        tools-freebsd v.8.4.8.19539

Mon Nov 28 20:51:02 2011 (UTC)
        tools-solaris v.8.4.8.19539

Mon Nov 28 20:50:46 2011 (UTC)
        tools-linux v.8.4.8.19539

Mon Nov 28 20:49:38 2011 (UTC)
        tools-winPre2k v.8.4.8.19539

Mon Nov 28 20:49:29 2011 (UTC)
        tools-netware v.8.4.8.19539

Mon Nov 28 20:49:27 2011 (UTC)
        VMware Workstation v.7.1.5.19539
        VMware Workstation v.7.1.5.19539
```

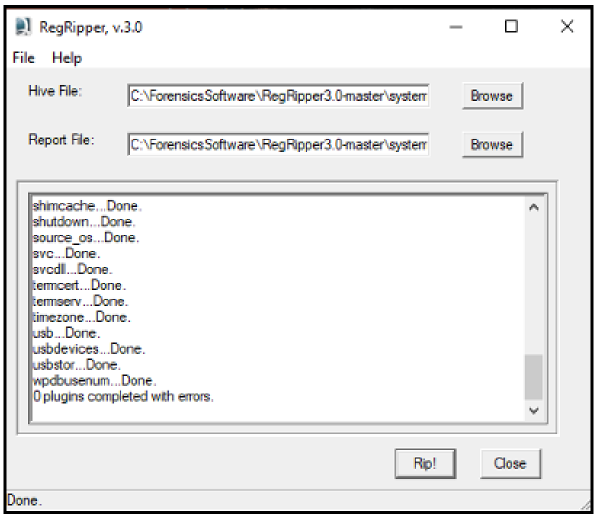
```
NOTE: This tool does NOT automatically process and incorporate Registry hive
transaction logs. The tool will check to see if the hive is dirty.

If you need to process/incorporate transaction logs, please consider using
yarp + registryFlush.py (Maxim Suhanov) or rla.exe (Eric Zimmerman).
```

Figure 62 – Usage of RegRipper

Through the command line it is possible to check the available plugins to apply through the argument "-l -c".

C:\RegRipper3.0-master>rip -l -c > c:\list.csv

In GUI mode (Figure 63), it is not possible for us to choose a single plugin, but the Hive that we want to analyse.

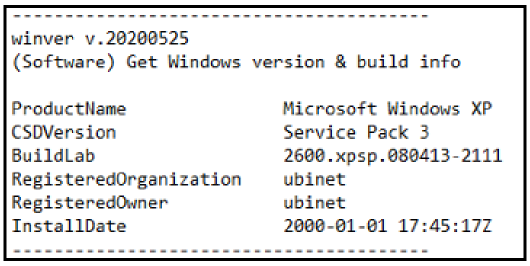


Hive File: C:\ForensicsSoftware\RegRipper3.0-master\system
Report File: C:\ForensicsSoftware\RegRipper3.0-master\system

```
shimcache...Done.
shutdown...Done.
source_os...Done.
svc...Done.
svcdll...Done.
termcert...Done.
termserv...Done.
timezone...Done.
usb...Done.
usbdevices...Done.
usbstor...Done.
wpdbusenum...Done.
0 plugins completed with errors.
```

Figure 63 - Using the RegRipper GUI

Output file (Figure 64)



```
-----------------------------------------
winver v.20200525
(Software) Get Windows version & build info

ProductName                Microsoft Windows XP
CSDVersion                 Service Pack 3
BuildLab                   2600.xpsp.080413-2111
RegisteredOrganization     ubinet
RegisteredOwner            ubinet
InstallDate                2000-01-01 17:45:17Z
-----------------------------------------
```

Figure 64 – RegRipper output file

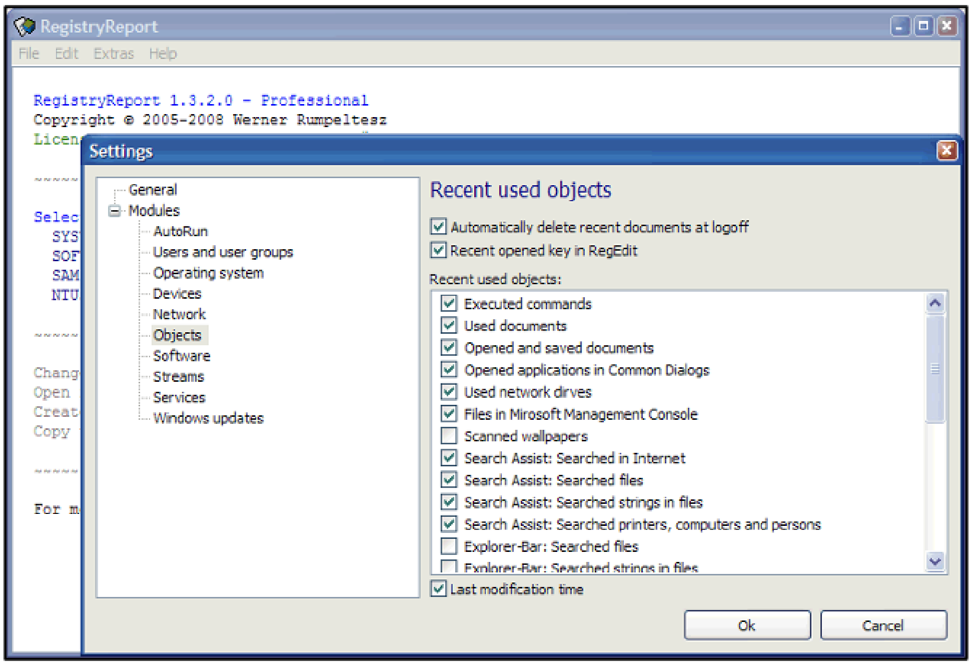
There are other forensic applications with the same objective of interpreting the contents of registry files, such as Registry Report and Windows Registry Recovery.

RegistryReport

Like RegRipper, Gaijin Registry Repport also presents the registry information in an easily readable and searchable way. It works simply, allowing you to select the information you want to retrieve from the registry via checkboxes, as shown in Figure 65.

Source: https://gaijin.at/en/files?dir=old-software registryreport

https://github.com/jschicht?tab=repositories



```
RegistryReport 1.3.2.0 - Professional
Copyright © 2005-2008 Werner Rumpeltesz
Licen
Selec
  SYS
  SOF
  SAM
  NTU
Chang
Open
Creat
Copy
For m
```

- General
- Modules
  - AutoRun
  - Users and user groups
  - Operating system
  - Devices
  - Network
  - Objects
  - Software
  - Streams
  - Services
  - Windows updates

Recent used objects

- [✓] Automatically delete recent documents at logoff
- [✓] Recent opened key in RegEdit

Recent used objects:

- [✓] Executed commands
- [✓] Used documents
- [✓] Opened and saved documents
- [✓] Opened applications in Common Dialogs
- [✓] Used network drves
- [✓] Files in Mirosoft Management Console
- [ ] Scanned wallpapers
- [✓] Search Assist: Searched in Internet
- [✓] Search Assist: Searched files
- [✓] Search Assist: Searched strings in files
- [✓] Search Assist: Searched printers, computers and persons
- [ ] Explorer-Bar: Searched files
- [ ] Explorer-Bar: Searched strings in files

- [✓] Last modification time

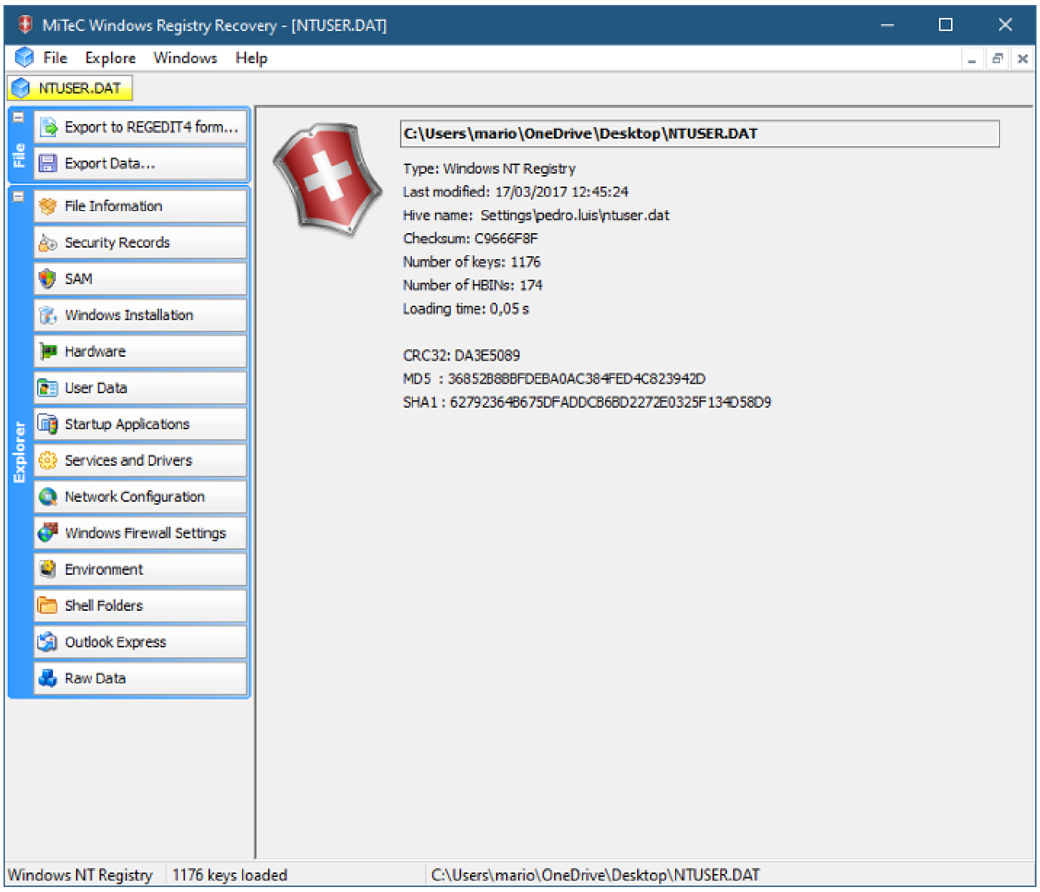
Figure 65 – Usage of RegistryReport

Source: https://www.gaijin.at/en/files?dir=old-software&sort=N&order=A registryreport

Windows Registry Recovery

WRR (Figure 66) is one of the applications we can use for Windows registry analysis

Windows Registry Recovery
The best tool for crashed machine registry configuration data recovery



```
C:\Users\mario\OneDrive\Desktop\NTUSER.DAT

Type: Windows NT Registry
Last modified: 17/03/2017 12:45:24
Hive name:  Settings\pedro.luis\ntuser.dat
Checksum: C9666F8F
Number of keys: 1176
Number of HBINs: 174
Loading time: 0,05 s

CRC32: DA3E5089
MD5  : 36852B8BBFDEBA0AC384FED4C823942D
SHA1 : 627923648675DFADDCB68D2272E0325F134D58D9
```

```
Windows NT Registry | 1176 keys loaded | C:\Users\mario\OneDrive\Desktop\NTUSER.DAT
```

Figure 66 – Using MiTeC Windows Registry Recovery

Source: http://www.mitec.cz/wrr.html

5.3. Analysis of Linux Based systems

The digital forensics in MS Windows operating systems, are widely disseminated, either through courses and scientific articles, or through new media such as videos. The digital forensics in Linux operating systems, are not so widespread, mainly because the analysis of these is also much smaller.

File Systems

The standard filesystem currently on Linux is Ext4 although it supports different types of filesystems

Linux File System	**Data**	
Ext	1992	Meaning "Extended file system", it was the first file system created for linux in 1992
Ext2	1993	It supported disks up to 2 TB and did not support journaling. Because it does not use journaling it can be used on USB sticks.
Ext3	1999	Same as Ext2, but with the advantage of journaling.
Ext4	2006	The current version of Ext. types has several advantageous features when compared to its predecessors, such as reduced system fragmentation, works with large files, and more. EXT4 supports 1EB (1 Exabyte) maximum filesystem size and 16TB maximum file size. It is possible to have an unlimited number of subdirectories

General considerations

1. There are no log files as in Windows OS
2. Information should be collected in dispersed locations
3. Different System file structures in different distributions

The file and folder structure of the Linux system can be summarised as shown in Figure 67.

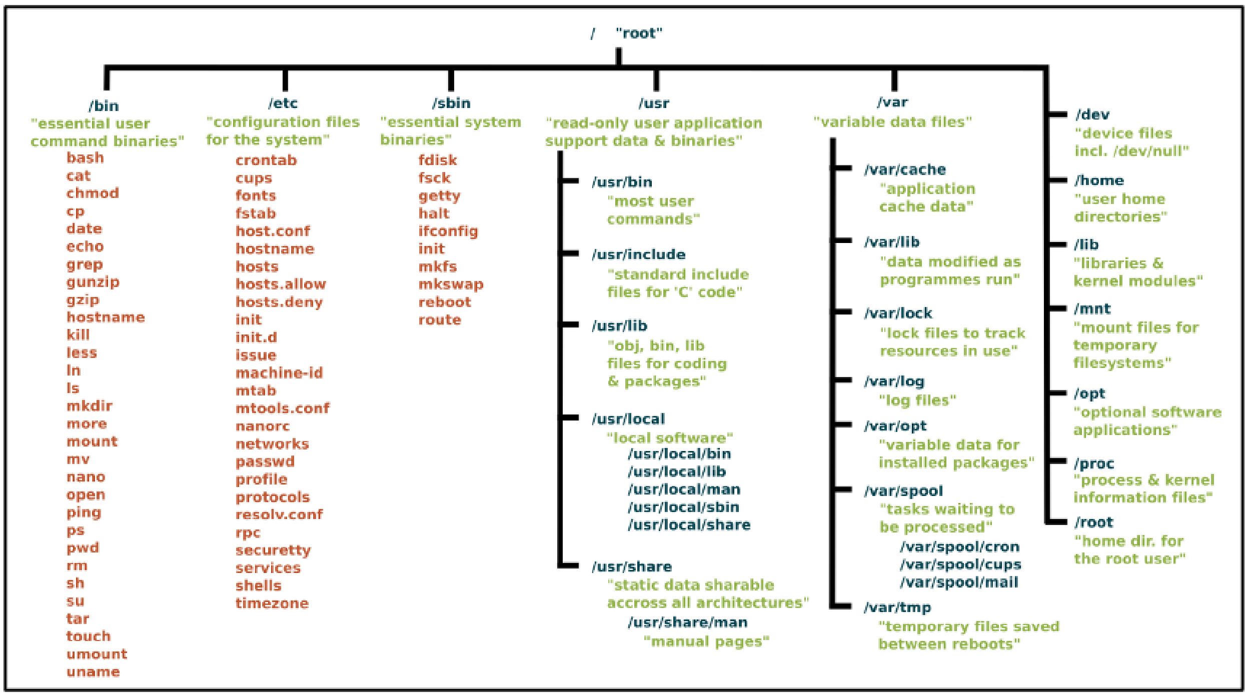


Figure 67 - Linux System File Structure

Source: The Linux Foundation – https://linuxfoundation.org/blog/classic-sysadmin-the-linux-filesystem-explained/

5.3.1. Points of Interest in Linux Systems

The analysis of user activity on Ubuntu Linux systems should follow a sequence of validations and information gathering, such as the one presented in Figure 68.

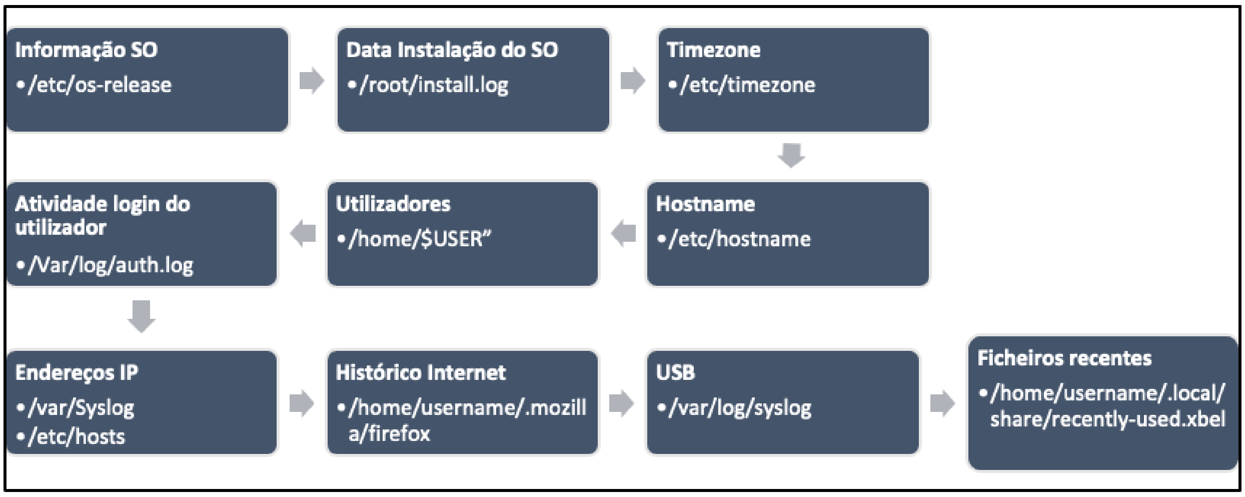


Figure 68 - Proposal for the collection of information on Linux

Autorun of programs running on the system :

Bear in mind that many programs are configured to start automatically at system startup. The information about the programs that should be run at startup is in the "**/etc/rc.local**" directory.

Documents accessed :

The examiner can know which documents have been accessed recently. The file containing this information is in **/home/user/.local/share/recently-used.xbel.** The **cat** command can be used to view the contents of the file. The .xbel file provides detailed information about the files that have been accessed by the user, such as access and modification time..

Installed Applications :

The application information is in the folder **/usr/bin** the libraries needed for the applications are in the folder **/usr/lib.** The list of applications can be obtained by the command **ls –l /usr/bin/.** It is possible to understand installation date, permissions, size, etc.

Network informations:

Ubuntu keeps a list of the networks connected to the system in: **/etc/NetworkManager/system-connections**

The file **/var/log/syslog** provides the date and time when a network connection was established.

Connected equipment:

The /dev directory provides information about the hardware connected to the system.

The file **/var/log/syslog** also has information about the devices that have been connected to the system.

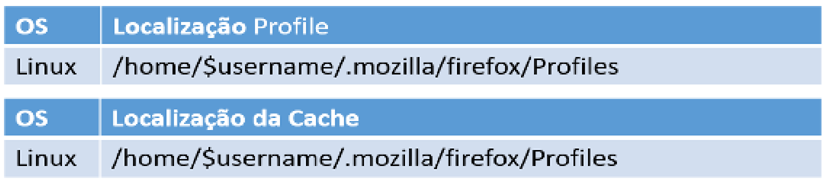
Last login and activity of the User::

The information about the last login can be obtained in **/var/log/lastlog**

Internet browsing activity:

We present the location of folders with navigation information, in two of the main browsers used in the Linux operating system (Figure 69 and Figure 70). After extracting these contents, it becomes possible to analyse them in the same way as in Windows .

Firefox Browser

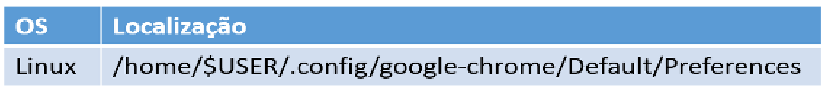


OS	Localização Profile
Linux	/home/$username/.mozilla/firefox/Profiles

OS	Localização da Cache
Linux	/home/$username/.mozilla/firefox/Profiles

Figure 69 - Firefox Browser Information Location

Google Chrome



OS	Localização
Linux	/home/$USER/.config/google-chrome/Default/Preferences

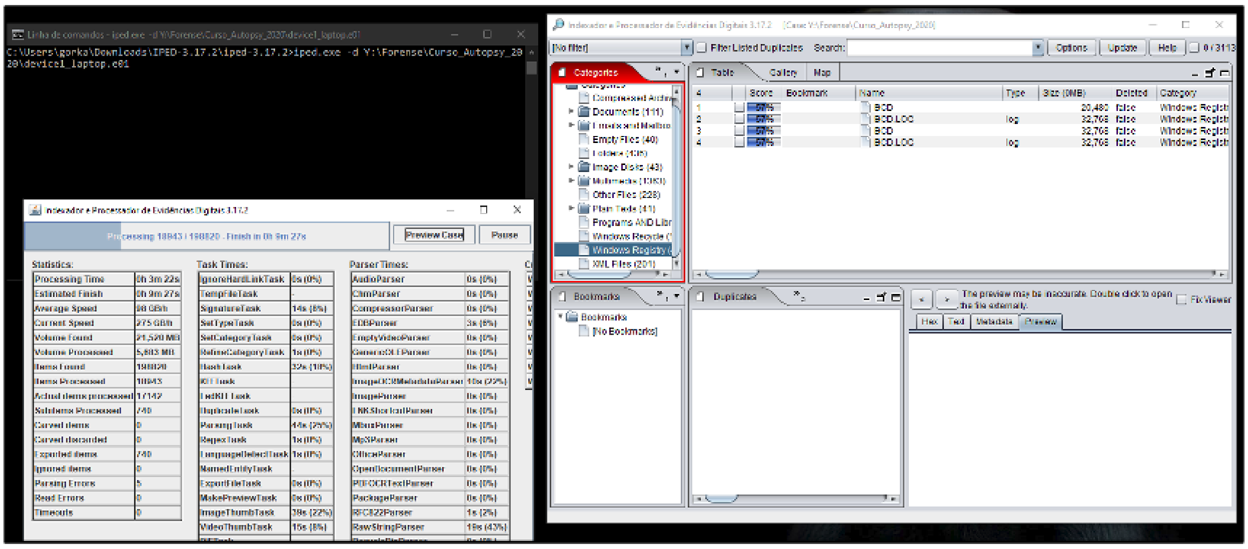
Figure 70 - Google Chrome browser information location

6. Forensic analysis with free suites of use

A forensic analysis requires knowledge of specific tools that can obtain and process the information we want. Many are commercial tools, such as OpenText EnCase, AccessData FTK, Magnet Axiom, among others. As free tools, the situation is quite different, since there are very few analysis suites available for use. This way, we will focus on 2 of these suites, IPED and Autopsy The Sleuth Kit.

6.1. IPED

IPED – Indexer and Digital Evidence Processor is an open source tool, developed in Java by the Brazilian Federal Police forensic investigation, being known for its good processing performance (Figure 71. It was developed to enable the analysis of large volumes of data by a large number of people, as it was intentionally developed for the investigation of the Lava Jato operation in Brazil. The high multithread performance of up to 400GB/h processing speed enables support for large cases with a large volume of data to be processed.



```
C:\Users\gorka\Downloads\IPED-3.17.2\iped-3.17.2>iped.exe -d Y:\Forense\Curso_Autopsy_20
20\device1_laptop.e01
```

Processing 18843 / 198820 - Finish in 0h 9m 27s

Statistics:

Processing Time	0h 3m 22s
Estimated Finish	0h 9m 27s
Average Speed	98 GB/h
Current Speed	275 GB/h
Volume Found	21,520 MB
Volume Processed	5,803 MB
Items Found	198820
Items Processed	18843
Actual items processed	17142
SubItems Processed	740
Carved items	0
Carved discarded	0
Exported items	740
Ignored items	0
Parsing Errors	5
Read Errors	0
Timeouts	0

Task Times:

IgnoreHardLinkTask	0s (0%)
TempFileTask	-
SignatureTask	14s (8%)
SetTypeTask	0s (0%)
SetCategoryTask	0s (0%)
RefineCategoryTask	1s (0%)
HashTask	32s (18%)
KFFTask	-
LedKFFTask	
DuplicateTask	0s (0%)
ParsingTask	44s (25%)
RegexTask	1s (0%)
LanguageDetectTask	1s (0%)
NamedEntityTask	-
ExportFileTask	0s (0%)
MakePreviewTask	0s (0%)
ImageThumbTask	39s (22%)
VideoThumbTask	15s (8%)

Parser Times:

AudioParser	0s (0%)
ChmParser	0s (0%)
CompressorParser	0s (0%)
EDBParser	3s (8%)
EmptyVideoParser	0s (0%)
GenericOLEParser	0s (0%)
HtmlParser	0s (0%)
ImageOCRMetadataParser	10s (22%)
ImageParser	0s (0%)
LNKShortcutParser	0s (0%)
MboxParser	0s (0%)
Mp3Parser	0s (0%)
OfficeParser	0s (0%)
OpenDocumentParser	0s (0%)
PDFOCRTextParser	0s (0%)
PackageParser	0s (0%)
RFC822Parser	1s (2%)
RawStringParser	19s (43%)

Linha de comandos - iped.exe -d Y:\Forense\Curso_Autopsy_2020\device1_laptop.e01
Indexador e Processador de Evidências Digitais 3.17.2
Preview Case
Pause
Categories
Table
Gallery
Map
Bookmarks
Duplicates
Options
Update
Help

Figure 71 – Processing with IPED

https://github.com/sepinf-inc/IPED

This is a software that requires some knowledge in its use, despite its simple and intuitive appearance, as can be seen in Figure 72.

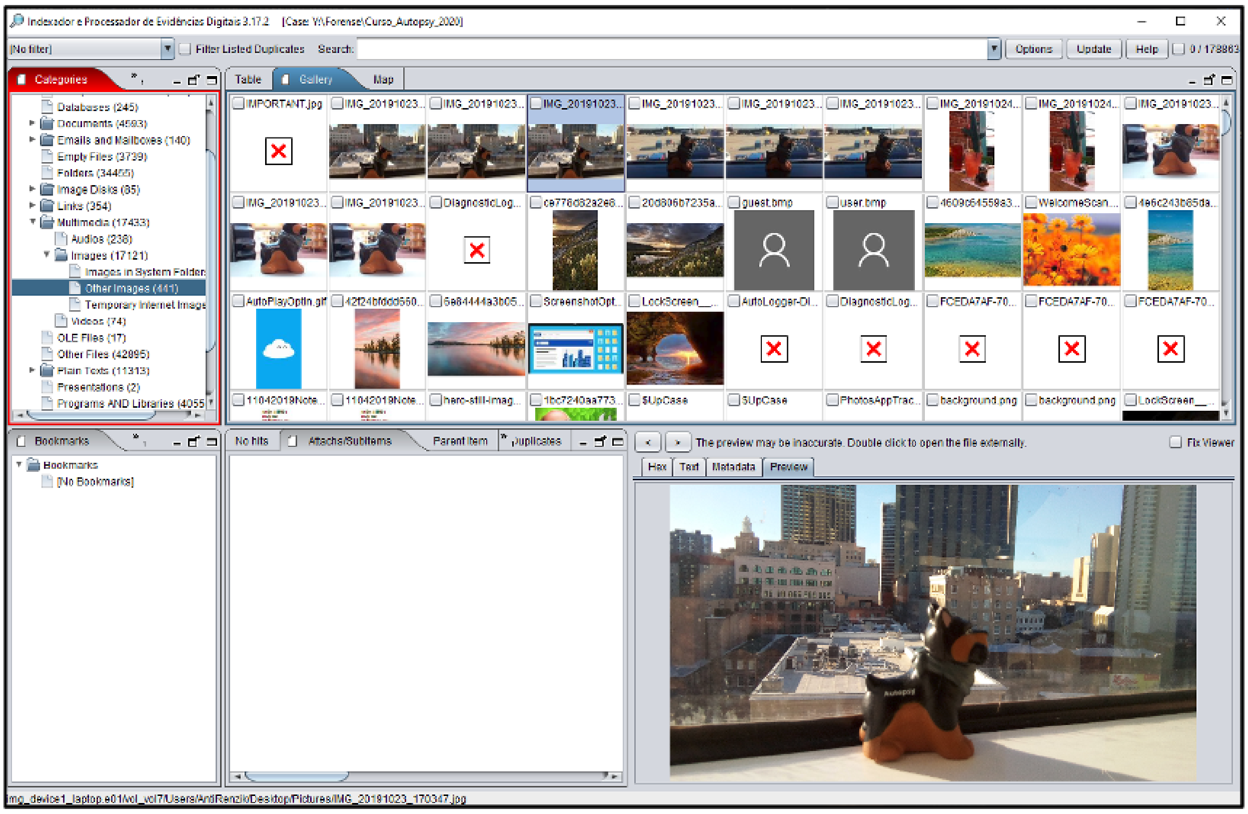


Figure 72 – Analysys with IPED

Source: https://github.com/sepinf-inc/IPED/wiki/Beginner's-Start-Guide

https://servicos.dpf.gov.br/ferramentas/IPED/

6.2. Autopsy The Sleuth kit

The Sleuth Kit® is a library and also a set of tools that allows the analysis of FAT, NTFS, Ext2/3/4 and UFS file systems, including those commonly used by the Linux Operating System, it also allows the analysis of files and folders, recover deleted files, create a *timeline* of file activities, perform expression searches and use databases of *hashs*.

https://github.com/sleuthkit/sleuthkit/blob/develop/NEWS.txt

Autopsy - Autopsy is the graphical user interface (GUI) of The Sleuth Kit. It is one of the open source platforms, developed to take advantage of The Sleuth Kit's capabilities to perform forensic analysis on devices such as hard drives, media cards, smartphones, among others. It also integrates other forensic tools, both open source and/or commercial, through plug-ins or complementary modules in Java or Python..

https://github.com/sleuthkit/autopsy/blob/develop/NEWS.txt

Autopsy's simple graphical interface is presented (Figure 73).

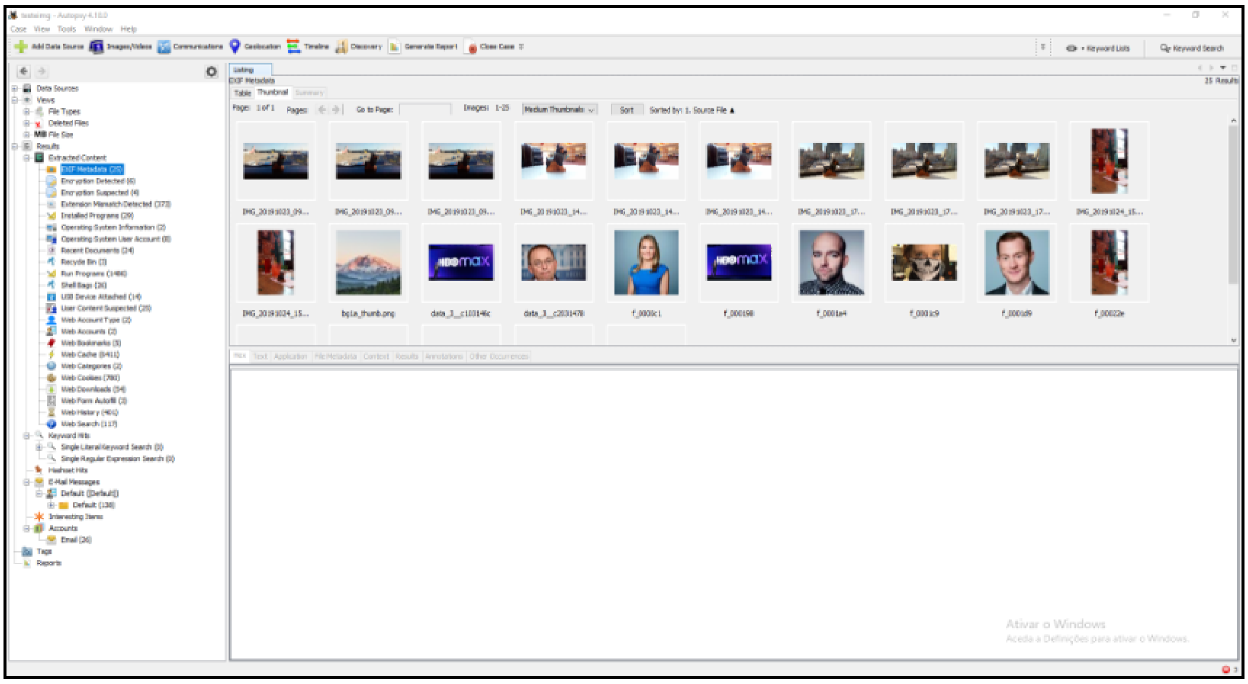


Figure 73 - Analysis with Autopsy

It contains a left side menu with categorised information, identifying files by extension type and MIME Type, but also all the categories they belong to (- Categorisation of files in Autopsy Figure 74).

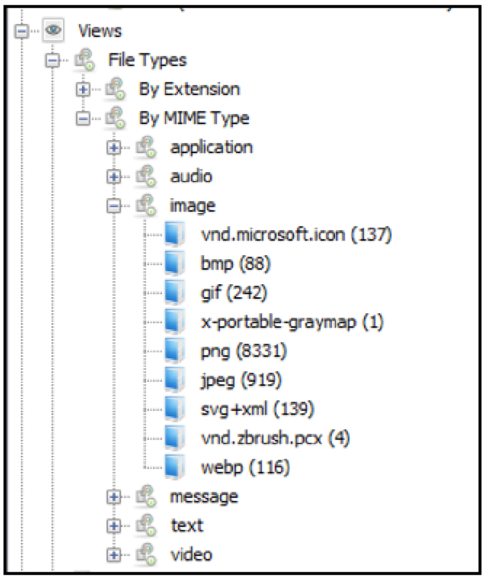


Figure 74 - Categorisation of files in Autopsy

Versions

https://github.com/sleuthkit/autopsy/releases/

Source Code

https://github.com/sleuthkit/autopsy

Tasks and requests

https://github.com/sleuthkit/autopsy/issues