UNIVERSIDAD POLITÉCNICA DE MADRID

Trabajo de Fin de Máster

ANALYSIS AND DESIGN OF DIGITAL FORENSICS AND INCIDENT RESPONSE PROCEDURE

Máster Universitario en Ciberseguridad

Javier Martínez Llamas

2019

TÍTULO: Analysis and Design of Digital Forensics and Incident Response Procedure

AUTOR: Javier Martínez Llamas

TUTOR: Lórien Doménech Ruiz

PONENTE: Víctor Villagrá González

Escuela Técnica Superior de Ingenieros de Telecomunicación

Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación

Escuela Técnica Superior de Ingenieros Informáticos

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

TRIBUNAL:

PRESIDENTE:

VOCAL:

SECRETARIO:

FECHA DE LECTURA:

CALIFICACIÓN:

Fdo: El Secretario del Tribunal

"*In our reasonings concerning matter of fact, there are all imaginable degrees of assurance, from the highest certainty to the lowest species of moral evidence. A wise man, therefore, proportions his belief to the evidence.*"

David Hume

Abstract

This project presents the analysis and design of an action procedure within a Digital Forensics and Incident Response (DFIR) framework. As to understand, and eventually be able to replicate, the methodology, tools and guidelines to be used in digital forensics.

Through the study of state of the art techniques, the aim is to perform DFIR tasks on Windows environments and gather information to be analysed. So an effective procedure can be documented and developed and thus facilitate future response in the event of cybersecurity incidents.

It is also intended to study, through practical scenarios (proposed by NIST and CCN-CERT), the procedures in the two main fields that comprise digital forensics; these being the analysis of volatile memory and non-volatile memory, in order to understand the similarities and differences in both fields.

Resumen

Este proyecto presenta el análisis y diseño de un procedimiento de actuación dentro de un marco DFIR. A fin de comprender, y eventualmente poder replicar, la metodología, herramientas y directrices que se utilizan en la informática forense.

A través del estudio del estado del arte de las técnicas, el objetivo es realizar tareas DFIR en entornos Windows y recopilar información para, posteriormente, ser analizada. De este modo, documentar y desarrollar un procedimiento eficaz que facilite la futura respuesta en caso de incidentes de ciberseguridad.

Así mismo se pretende estudiar, mediante casos prácticos (propuestos por el NIST y CCN-CERT), los procedimientos en los dos campos principales que componen la informática forense; siendo estos el análisis de la memoria volátil y la memoria no volátil, a fin de comprender las similitudes y diferencias en ambos campos.

Contents

List of Figures

List of Tables

Acronyms

ATT&CK Adversarial Tactics, Techniques & Common Knowledge.

CCN-CERT Centro Criptológico Nacional Computer Emergency Response Team.

DFIR Digital Forensics and Incident Response.

DHCP Dynamic Host Configuration Protocol.

FAT File Allocation Table.

HDD Hard Drive Disk.

IDS Intrusion Detection System.

IoC Indicator of Compromise.

IRT Incident Response Team.

MISP Malware Information Sharing Platform.

NIST National Institute of Standards and Technology.

NTFS New Technology File System.

RAM Random Access Memory.

SANS SysAdmin Audit, Networking and Security.

SMB Server Message Block.

SOC Cybersecurity Operations Center.

SSD Solid State Disk.

USB Universal Serial Bus.

USN Journal Update Secuence Number Journal.

UTC Universal Time Coordinated.

VHD Virtual Hard Disk.

VLAN Virtual Local Area Network.

Chapter 1

Introduction

1.1 Motivation

The field of cybersecurity is experiencing, in information societies, a growth both in relevance and notoriety and in criticality. It has become an intrinsic value to the development of these societies.

Parallel to relevance, the social consciousness increases regarding the vulnerabilities and exposure of the systems on which we depend and, as a reaction to this relevance, attacks or incidences on these computer systems proliferate and arise. Fostering the development of cybersecurity.

This is where the motivation for this project lies, the new criminal paradigm and the necessary countermeasures to be taken; specifically DFIR as the final instance of these defensive measures. Offering a diametrically opposed vision, where the threat has prevailed and the damage has been caused, to the preventive nature of cybersecurity.

1.2 Objectives

The objective behind this project is to comprehend the digital forensic procedure within DFIR scope. Doing so, it will be possible to analyse the results obtained in order to establish and document an effective procedure for incident response in computer systems.

The aim is to deepen in three main topics:

- Essential concepts in digital forensics that underpin all procedures and techniques, as well as to understand the internal functioning of Windows,

the operating system on which we will focus this project.

- Study and define a DFIR procedure that will allow us to perform a simulation or representation of two practical cases.
- Case studies on which to study different forensic procedures, as part of incident response, and tools applied to two scenarios opposed in nature: the analysis of volatile and non-volatile memory.

This documentation will include theoretical foundations on the analysed systems, methodology for incident response and practical outcomes based on simulated incidents, on which the forensic tools will be tested. In a case study, we will work on an information theft scenario developed by NIST [1] and on an intrusion scenario provided by the CCN-CERT.

1.3 Structure

This work is structured in a total of 6 chapters, the division of which aims to deepen on a specific field within DFIR in each chapter:

Introduction The first chapter and synthesis of the project itself. It defines and details the objectives of the project, its motivation and structure, in order to justify subsequent chapters and declare their content.

State of the Art Theoretical basis and previous studies on which the subsequent chapters are based. It includes a vision from the beginnings of forensic science to its extrapolation to cybersecurity and inclusion in the DFIR procedure. We also study and select the different tools that will later be used during the case studies.

DFIR Procedure This chapter integrates and defines the different procedures and standards related to Digital Forensics and Incident Response. Defining a methodology for incident response and the application of computer forensics in it.

Case Study: Non-volatile Memory Analysis Two case studies have been defined, on the basis of the previous chapter, in order to distinguish between the two most common scenarios within digital forensics. This chapter will include the analysis of non-volatile memory, always framed by the methodology defined and studied.

Case Study: Volatile Memory Analysis As a counterpart to the previous chapter, a case study focusing on volatile memory is defined in a complementary manner. Also framed in the methodology proposed in Chapter 3.

Conclusions Once the entire study has been carried out, a series of conclusions are synthesized and drawn that depict this project; different concerns or considerations are presented regarding DFIR and possible improvements that this field of cybersecurity could undergo.

Chapter 2

State of the Art

As a basis for the realization of the project, a theoretical study on the state of the art will be carried out, which will support and determine all the subsequent sections. Due to the great theoretical amplitude behind digital forensics, different phases in the study will be defined:

- Scope and Context
- Prior Concepts
- Available tools and their functionalities

This division is intended to start from a basis of file system, Windows behaviour, its internal structure and particularities to later scale to current forensic techniques and finally study and compare the various tools that make use of them. While each of these stages builds on its predecessors and is preceded by a general context.

This project will focus its efforts on the Microsoft Windows operating system. However, there are many parallels between the different environments and systems available. In which case the procedure could be, if not identical, very similar.

2.1 Historical Context

As societies developed, and their legal system with them, the need arose to clarify and justify the crimes committed. The first reference to a case solved by evidence goes back to medieval China [2]. Written around 1247, *The Washing Away of Wrongs* sets the first precedent of forensic science and forensic medicine while clarifying and discerning the possible cause of death in asphyxiation cases.

It was in the 19th century that forensic science experienced its greatest development. Supported by the rational values of the Enlightenment, criminology underwent a new orientation towards a more rational, evidence-based approach and procedure.

Since then, and following the progress of technology and cultural development, forensic science has become a critical aspect of law enforcement and crime resolution. Just as the complexity of crimes increases, forensic techniques and procedures are perfected.

This is where information technology comes in, the economic and cultural engine of information societies. Globalization and the ease of access to resources and information force a new paradigm in the security and crime scope.

Compared to traditional crimes, computer attacks benefit enormously in terms of cost/benefit ratio, where anonymity plays a major role. All this leads to a scenario where cybercrimes are constant and are perpetrated by numerous attacker profiles. Therefore, there is a need to replicate traditional forensic techniques in the field of computer science: digital forensics.

2.2 Cybersecurity: Defensive Scope

2.2.1 Security Operations Center (SOC)

Cybersecurity should be understood as a continuous protection of computer systems and networks and not as an isolated event. It is necessary, therefore, a whole structure and planning responsible for providing the necessary measures to ensure this protection.

In the defensive aspect of security that concerns us, it is common to have the presence of a Cybersecurity Operations Center (SOC) that acts as a central node and coordinator of all security policies and tools. This is where the Incident Response Team (IRT) engages, offering an incident response service for the most advanced or non-catalogued threats and therefore lacking procedure.

However, the SOC has different levels of security that include threat monitoring, detection, analysis and response. As well as the management of the different security tools of the organization. Therefore, cybersecurity management is distributed among numerous teams and it is at the last stage of this process when IRT intervenes.

2.2.2 Digital Forensics and Incident Response

Information security, in the defensive scope, involves, as mentioned, numerous stages, from prevention and monitorization to incident response, once the cyberattack has already occurred and succeed. Like traditional forensic science, computer forensics starts from the premise that the harm has already occurred and there is a possible victim.

Digital Forensics and Incident Response (DFIR), by definition, consists of two related but equally differentiated parts: Forensics and Incident Response. Once a security incident or threat has been detected and examined in a first approach, in the event that it could not be analysed in depth or so it was decided, an expert and specialized evaluation would be required.

The Incident Response stage consists, first and foremost, of isolating and containing the threat, minimising its potential damage. Eliminate the cause of the problem and take the necessary steps so that, in the future, the threat is prevented. This is where, if necessary, the forensic phase begins.

In a first stage of this analysis, use would be made of digital forensics, with the aim of extracting as much information as possible from the affected machines that could clarify what happened through its analysis.

2.3 Prior Concepts

This section aims to define and establish certain concepts necessary for further analysis and study. Therefore, its subject matter is not as strictly related to DFIR as the subsequent sections, but rather supports its entire procedure.

2.3.1 File Systems

There are numerous file systems, varying between different operating systems, drive types, or user preferences. However, there is some consensus within the same environment or removable drives.

Computer forensics has as one of the most relevant and critical aspects the recovery and extraction of the user's data. Since most information is stored in Solid State Disk (SSD) or Hard Drive Disk (HDD) it is necessary to understand both the structure and functioning of file systems. We will exclusively focus on the main systems used in Microsoft Windows: NTFS, FAT and its variants.

New Technology File System (NTFS)

Microsoft Windows operating system uses New Technology File System (NTFS) as its default file system, as opposed to the ext3 and ext4 systems of Linux, since Windows 2000.

A cluster is the unit or set of contiguous sectors that make up the smallest storage unit on a disk. If a file is larger than the cluster, it is divided among several. On the contrary, if the file is smaller than the size of a cluster, it is completely stored in it.

Although not going into detail, this concept is essential to understand how files are distributed and split within a unit and the need for certain characteristics of file systems such as allocation tables.

NTFS has the following cluster sizes depending on the size of the unit [3], this will be relevant in the different tools used.

Volume Size	Cluster Size
7MB	512 bytes
513MB - 1024MB	1KB
1025MB - 2GB	2KB
2GB - 2TB	4KB

Table 2.1: NTFS Cluster Size

And the following organization of the volume:

NTFS Boot Sector	Master File Table	File System Data	Master File Table Copy

Table 2.2: NTFS Organization

The Boot sector contains general information about the volume and structure. File System Data stores the actual files and the Master File Table Copy acts as a recovery partition. The Master File Table stores the previously mentioned information to allow the retrieve of files from a partition [4].

This pattern is tested on the NTFS disk that hosts a Windows operating system. To do so, Windows tool DiskPart is used:

```
Partition ###  Type              Size     Offset
-------------  ----------------  -------  -------
Partition 1    System            260 MB  1024 KB
Partition 2    Reserved           16 MB   261 MB
Partition 3    Primary           118 GB   277 MB
Partition 4    Recovery          980 MB   118 GB
```

Figure 2.1: DiskPart NTFS Partition Output [1]

It is possible to check the content of such partitions, however this will not be mentioned in this section but during the analysis of the tools.

NTFS has a very useful functionality for forensic analysis, a record of changes called Update Secuence Number Journal (USN Journal). It provides persistent log of changes made on the volume [5].

```
Usn              : 110327576
File name        : $dpx$.tmp
File name length : 18
Reason           : 0x80000100: File create | Close
Time stamp       : 3/5/2019 3:01:25
File attributes  : 0x00000010: Directory
File ID          : 0000000000000000009f00000001d503
Parent file ID   : 000000000000000000030000001e95e
Source info      : 0x00000000: *NONE*
Security ID      : 0
Major version    : 3
Minor version    : 0
Record length    : 96

Usn              : 110327656
File name        : $dpx$.tmp
File name length : 18
Reason           : 0x0000c000: Indexable change | Basic info change
Time stamp       : 3/5/2019 3:01:25
File attributes  : 0x00002012: Hidden | Directory | Not content indexed
File ID          : 0000000000000000009f00000001d503
Parent file ID   : 000000000000000000030000001e95e
Source info      : 0x00000000: *NONE*
Security ID      : 0
Major version    : 3
Minor version    : 0
Record length    : 96
```

Figure 2.2: Fsutil USN Journal Output

File Allocation Table (FAT)

File Allocation Table (FAT) was designed as a robust and simple file system for small units. Originally used in the early versions of Microsoft Windows it is still widely used on USB and peripheral devices. Specially on its latest versions and variants like FAT32 and exFAT.

It bases its operation, in the same way as NTFS, on the File Allocation Table that provides an index for the files contained in the volume, statically assigned during formatting. It follows a structure of linked lists, where each entry of the table points to the next cluster of the volume or file.

[1] The offset is an address which defines the distance from the base pointer of the memory.

As the size of volumes has increased over time, and the limitations of the FAT system were present, new versions have emerged: FAT12, FAT16, FAT32 and exFAT.

	FAT12	FAT16	FAT32	exFAT
Max. File Size	32MB	2GB	4GB	16EB
Max. Volume Size	32MB	4GB	2TB	128PB

Table 2.3: FAT Variants Comparison

FAT file systems are clearly inferior to the competition as NTFS. However, its extensive compatibility and history makes it predominant on small removable drives. Therefore, as far as forensic analysis is concerned, it is necessary to study such systems.

2.3.2 Write Block

A fundamental premise within digital forensics is the integrity of the evidence and information analysed. This should not be altered during all phases of analysis, especially in the judicial arena, where the chain of custody is critical.

Although in the DFIR environment this is not as relevant, it is equally necessary to preserve an original copy and write-protect the disks where evidence is stored.

Write Blockers can be both software and hardware. However, there may be incompatibilities among different operating systems in software blockers. Hardware blockers, on the contrary, tend to be software independent and, therefore, less susceptible to failure.

```
USB DISK 2.0 USB Device
Disk ID: 0005191D
Type   : USB
Status : Online
Path   : 0
Target : 0
LUN ID : 0
Location Path : UNAVAILABLE
Current Read-only State : Yes
Read-only  : Yes
Boot Disk  : No
Pagefile Disk  : No
Hibernation File Disk  : No
Crashdump Disk  : No
Clustered Disk  : No

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 4     F   UNIT         FAT32  Removable     14 GB  Healthy
```

Figure 2.3: DiskPart Read-Only Exemplification

2.3.3 Chain of Custody

The chain of custody is a procedure for performing documentation to the evidence in chronological events so that it is accepted in court. The authenticity and integrity of the evidence must be preserved during all stages of the investigation, so its state must be identical to that originally discovered [6].

This is particularly relevant in the information technology field, where the large number of crimes committed and the enormous amount of information associated with them, as well as their fragility, constitute a major aspect of any investigation.

2.3.4 Media

A critical aspect of forensic analysis is the extraction of evidence. While there are many sources of data within a system, much of the information resides in data storage media. It is not surprising, therefore, that a fundamental part of the dedication during the analysis is devoted to these devices.

Random Access Memory (RAM)

Random Access Memory stores data and code that is currently in use on a computer. Faster than data storage units such as hard disks, it acts as a proxy between the processor and the volume where persistent information is stored.

Importance in Forensics RAM is volatile. Meaning all information stored in it will be deleted in the event that memory is disconnected. This raises two scenarios: The loss of relevant and fragile information and the use of new malware techniques.

Depending on the environment or circumstances to which a system was subjected, a traditional approach may not be profitable or even possible. A similar case is malware analysis, where, in order to go unnoticed, the new malicious code is loaded entirely into RAM and non-persistent. Therefore avoiding permanent traces on disk and hindering their detection.

This new paradigm results in a branch within digital forensics known as Live Forensics (3.2.2).

Data Storage Devices

Non-volatile storage units such as Hard Drive Disk (HDD) and Solid State Disk (SSD) make it possible to persistently store large amounts of data. Not surprisingly, as the capacity of these systems increases, so does the relevant information they can hold.

Therefore, much of the dedication during a forensic investigation is devoted to the analysis of these storage units.

SSD was a great evolution with respect to HDD, resembling the operation of a RAM memory (without disks or mobile parts). However, by changing its architecture, it also means a change in forensic treatment.

The most immediate difference is the form of erasure. HDD is capable of overwriting old data, if a file is deleted it is only removed from the file tables (2.3.1) and therefore its information remains accessible until it is overwritten. On the contrary, SSD needs to remove each block prior to the new writing.

2.3.5 Encryption

Encryption, by definition, seeks to obfuscate the information and make it illegible to unauthorized third parties (who do not have the decryption key). Forensic analysis of an encrypted disk would be virtually impossible using a traditional approach [7].

To mention only, stenography also makes it possible, albeit not with the same guarantees, to hinder the forensic procedure and the extraction of relevant information by inserting such data in non-corresponding files. However, this information is not necessarily encrypted and therefore represents a different type of impediment.

Therefore, new methodologies such as Live Forensics are necessary for a correct examination of the disks, in case of not having access by other means such as brute force on the encryption algorithm.

2.3.6 Metadata

Metadata, from the Greek $\mu\varepsilon\tau\alpha$ (after or beyond) and from the Latin *datum* (a given), are, by definition, data about data. Or, in other words, information about a file or data.

In computer forensics this is of great relevance, providing information about not just files themselves, but indications of obfuscated information or correlating data.

There are different types of metadata. Those inherent to the file system, such as MAC times (timestamps). Information about images and their characteristics. And the information of files themselves, as their creator.

2.4 Forensic Tools

This section aims to illustrate the tools selected for this project and their utilities. They have been chosen on the basis of specific characteristics and needs and, therefore, there are numerous additional tools, some with the same purpose as those shown here, intended for different forensic fields and techniques. However, the intention is to show a standard vision in computer forensics.

FTK Imager

Concise and powerful forensic tool developed by AccessData for image treatment. Allows the mounting and browsing of images in either the proprietary EnCase or SMART format or in raw (dd) format.

```
0294306410 02 D8 22 09 56 D7 C5 1D-9E 2C B2 F0 24 20 08 27 ·Ø"·V×Å··,²ð$ ·'
0294306420 57 51 ED 14 4A B9 B8 30-60 95 D5 6A ED 86 D3 36 WQí·J¹¸0`·Õjí·Ó6
0294306430 05 1E F0 34 B3 70 5B 91-34 2A 63 DE 67 C3 86 73 ··ð4³p[·4*cÞgÃ·s
0294306440 07 B6 95 80 1B 60 27 8F-9F 1D F7 C2 58 89 F0 07 ·¶···`'···÷ÂX·ð·
0294306450 F7 DB 74 A7 4B 9F C7 25-F7 8F E8 CE 09 7E 7A 2D ÷Ût§K·Ç%÷·èÎ·~z-
0294306460 F7 E2 62 AD CC 4A FB FD-24 DB D1 86 4F D5 75 1F ÷âb-ÌJûý$ÛÑ·OÕu·
0294306470 F2 EA 6D 1A 5F 25 93 04-64 51 1A F7 61 E6 E0 BE òêm·_%··dQ·÷aæà¾
0294306480 18 61 46 16 23 A5 95 F2-EB 2D 71 4E 2E 8C 0B 01 ·aF·#¥·òë-qN.···
0294306490 1E 08 9C 2C 21 46 01 AC-17 E5 D0 12 DE 34 75 AC ···,!F·¬·åÐ·Þ4u¬
02943064a0 2E A4 CF 04 85 13 5C 14-1D 5E C2 F5 3B D9 43 02 .¤Ï···\··^Âõ;ÙC·
02943064b0 36 6E F6 13 18 8B 78 82-AC 27 61 3E 33 9F 80 00 6nö···x·¬'a>3···
02943064c0 E8 C7 69 7F 24 5B 2E 19-0B BD 3E 83 95 60 E9 41 èÇi·$[.··½>··`éA
02943064d0 32 06 60 8E 9C 1B D0 0F-60 9E 8B 4A 84 55 22 EC 2·`···Ð·`··J·U"ì
02943064e0 C1 8A B0 C9 D5 A8 74 2B-4A C0 91 6E 42 77 69 FE Á·°ÉÕ¨t+JÀ·nBwiþ
```

Figure 2.4: Partial FTK Image Hex Content

It also allows the cloning of connected physical disks and integrates a RAM capture function. As mentioned in previous sections, when cloning a disk it is necessary to do it on a write-blocked device.

Belkasoft RAM Capturer

Forensic tool for the acquisition of volatile memory, allowing bypass of anti-debugging and anti-dumping techniques. It does not require installation in order to minimize the footprints in the system to analyse, existing versions of both 32bits and 64bits.

Despite the fact that for this project the duly captured memory dumps will already be available, it is necessary to emphasize the acquisition tool to be used if this were not the case, both volatile and non-volatile memory, as it is a critical aspect of any forensic investigation.

Volatility

Open-source memory forensics framework for RAM images analysis implemented in Python, making it independent of the operating system treated. It works through profiles that determine the operating system to which the image belongs and supports the integration of numerous plug-ins.

```
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x64_17134, Win10x64_14393, Win10x64_10586, Win10x64_16299
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/mnt/d/Forense/Imágenes/Win10EmpireCon.mem)
                      PAE type : No PAE
                           DTB : 0x1ad002L
                          KDBG : 0xf802a33aa520L
          Number of Processors : 2
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xfffff802a2439000L
                KPCR for CPU 1 : 0xffffad8090343000L
             KUSER_SHARED_DATA : 0xffffff78000000000L
           Image date and time : 2019-04-03 08:44:32 UTC+0000
     Image local date and time : 2019-04-03 01:44:32 -0700
```

Figure 2.5: Volatility Image Profile Output Example

During this project version 2.6.1 of Volatility will be used along Python 2.7 with *pycripto* 2.6.1 and *distorm3* 3.4.1.

Autopsy

Digital forensics platform and graphical interface for The Sleuth Kit [8] allowing the analysis of disk images. Based on a modular approach it provides:

- Time-line Analysis.
- Hash Filtering.
- Keyword Search.
- Web Artifacts.
- Data Carving.
- Multimedia Metadata
- Indicator of Compromise

During this project version 4.10 of Autopsy will be used.

OST Viewer

Tool to access and view Microsoft Outlook OST files. These files serve as an offline copy of the server information so that, in the event of not having an internet connection, the emails can be accessed. OST Viewer version 5.0.0 will be used for this project.

USB Historian

It allows you to retrieve, through Windows registries, a history of the devices connected to the system. Version 1.3 of USB Historian will be used for this project.

ShellBags Explorer

GUI for browsing shellbags data and registry explorer under user's hives. During this project version 1.3.2.0 of ShellBags Explorer will be used.

SQLiteStudio

Open-source SQLite database manager for managing *.db files in Microsoft Windows. SQLiteStudio version 3.1.1 will be used for this project.

Thumbcache Viewer

Thumbcache Viewer allows the extraction of thumbnail images from the thumbcache_*.db and iconcache_*.db database files found on Microsoft Windows. Thumbcache Viewer version 1.0.3.6 will be used for this project.

Nirsoft ESEDatabaseView

ESEDatabaseView utility to read and display the data stored inside Extensible Storage Engine (ESE) database (.edb file). Nirsoft ESEDatabaseView version 1.62 will be used for this project.

ShadowCopyView

Tool for Windows that lists the snapshots of your hard drive created by the 'Volume Shadow Copy' service of Windows. Allows browsing older version of

files and folders and their export. ShadowCopyView version 1.05 will be used for this project.

Bulk Extractor

Computer forensics tool that scans a disk image, a file, or a directory of files and extracts useful information without parsing the file system or file system structures. Bulk Extractor version 1.5.2 will be used for this project.

Wireshark

Network protocol analyser. Allows the inspection of protocols, live traffic capture and analysis. Wireshark version 3.0.1 will be used for this project.

Chapter 3

DFIR Procedure

This section is intended to illustrate the main areas within Digital Forensics and Incident Response and to provide an understanding of the procedures to be followed during the conduct of a forensic investigation.

3.1 Incident Response Life Cycle

As prevention and reaction to the different security incidents emerges the incident response scope. This is one of the many security measures and cannot and should not be understood as an isolated proposal.

Incident response is the final stage of defence, where, once a cybersecurity incident has occurred, it is necessary to intervene to contain, study, and eradicate it. This whole process serves, in the future, to prevent incidents of the same nature.

It is divided into different phases, their number varying according to the guides and sources consulted. However, for this project, we will refer to SysAdmin Audit, Networking and Security (SANS) Institute and National Institute of Standards and Technology (NIST) to define these phases.

SANS Institute [9] defines 6 phases during the incident response life cycle, while NIST [10] defines a total of 4 phases. This difference is, broadly speaking, a difference in structure. The concepts and conclusions are similar in both cases.

In the case of SANS, it is structured as follows:

1. Preparation
2. Identification

3. Containment
4. Eradication
5. Recovery
6. Lessons Learned

For clarity and synthesis we will opt for the one offered by NIST.

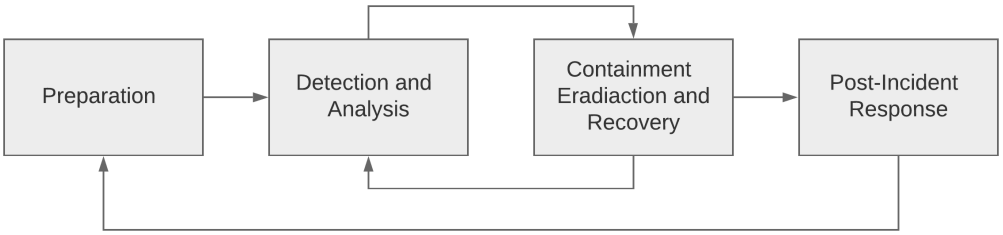

Figure 3.1: Incident Response Life Cycle [10]

Framing computer forensics within this cycle would result in:

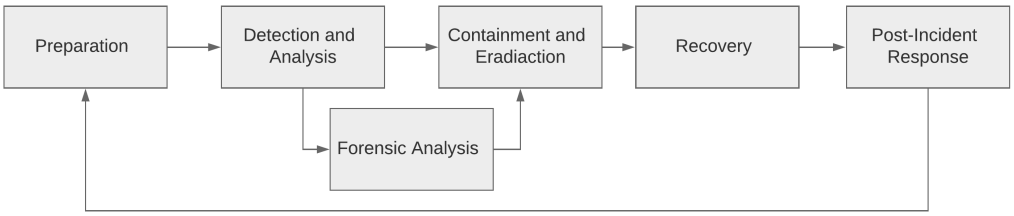

Figure 3.2: Incident Response Life Cycle and Forensics

3.1.1 Preparation

Every procedure and methodology needs, for its proper functioning, a solid base and fundamentals. This is especially true in the case of incident response. Not only it is necessary to have an effective and forceful response capacity, but, as possible, it is essential to prevent these incidents in order to ensure infrastructure security.

While, as previously mentioned, the Incident Response Team (IRT) is not responsible for prevention (there are different levels of security), their proper performance and feedback is critical to the process.

To this end, there are several points to be fulfilled which, synthesised, would be: ensuring the technical capacity of the teams involved, developing documen-

tation and action policies that are duly verified and robust, and providing the teams with the necessary technological and human resources.

All these points contribute to maintaining minimum and manageable levels of incidents, with preparedness and prevention being the greatest defence system. If these procedures were not properly developed, the level of incidents and, therefore, the workload of IRT (and previous teams) would be overwhelming and its capacity for action and effectiveness would be nullified.

3.1.2 Detection and Analysis

Incidents, in order to be treated, must be detected. One of the most common methods is the use of monitoring systems. In the case we are concerned with, DFIR, if the incident has been escalated, it is because either the seriousness and complexity requires it or because there are no procedures adapted to it and, therefore, the threat is unknown.

Once an incident has been declared or malware has been detected, it is necessary to identify if it is indeed a threat to be considered. For this purpose, different techniques according to the nature of the threat are available. Due to the subject matter of this project, the following points will focus on forensic analysis. However, there are other alternatives such as malware analysis, both dynamic and reverse engineering. It is worth mentioning that these analysis techniques are not exclusive, being possible to perform a forensic analysis in the first place and, from the extracted information, obtain the malicious binary and dissect it.

The purpose of this analysis, beyond mitigating its impact, is the extraction of information and intelligence to prevent the threat in the future. A clear example of this is the extraction of Indicator of Compromise (IoC) that will be fed back into IDS or collective intelligence tools such as Malware Information Sharing Platform (MISP).

3.1.3 Containment, Eradication and Recovery

Once the scope of the threat, its origin and techniques have been determined, it is necessary to forestall its diffusion in order to minimize its potential damage or the one already caused.

To do so there are different procedures to follow. From the most immediate measures such as the isolation of the system and its network to the formatting of its disks. However, before that, it may be necessary in DFIR case, to make a clone of the image of the machine in order to have an instance of the infection if required.

Once the compromised machine has been restored and sanitized, it is necessary, so that it does not happen again, to take the necessary measures if required.

3.1.4 Post-Incident Response

The final phase of any incident response process is, in itself, one of the most important. It is about learning and improvement. Highlighting what has happened and the intelligence that has been extracted from it is fundamental to the entire cybersecurity infrastructure, feeding back with each incident and threat. Helping to prevent future threats.

There are different ways to achieve this response. It is essential for a good cybersecurity management that the knowledge and expertise gained from the threat is passed on to the lower levels of the SOC, facilitating the future work of all teams and improving their experience. It is also possible to share this knowledge with the community through the above-mentioned MISP or the MITRE ATT&CK framework.

3.2 Digital Forensics Areas

Although the term digital forensics refers to a global idea, and although its original conception adhered to a single definition, it has now been partially fragmented by the great evolution that computer science has undergone. It is divided into the following areas: Computer Forensics, Live Forensics, Mobile Forensics, Network Forensics and Database Forensics.

All these fields are of importance, however, due to the structure and approach of this project we will focus mostly on Computer and Live Forensics on Microsoft Windows environments.

3.2.1 Computer Forensics

Originally synonymous with digital forensics, this specialization has served to rename itself as computer forensics. The aim is to examine and analyse, by definition, stand-alone computers, their media and data.

As it is the primitive branch of digital forensics, it is reasonable to think that its development and study was significantly more advanced than the rest of areas. However, the lack of standardization and procedures in these branches has forced the community to focus its efforts on these new fields [11].

This leaves computer forensics in an already established and consolidated position but which, as computing progresses, becomes outdated and lethargic.

3.2.2 Live Forensics

In contrast to the traditional methodology of the previous area, Live Forensics considers of great importance the data that could be lost when turning off a system. While the former is inspired by traditional forensic techniques from classic medicine, the latter needs the system to be running (alive) due to this fragility in information.

This raises a number of issues. In which it is impossible to perform a live analysis without substantially altering the original state of the system. For this reason, the aim is to minimise this impact on the integrity of the information during the collection of evidence.

This casuistry is due to the operations carried out while analysing, inevitably writing in the system. Modifying from records to dates and memory during the acquisition. Write blockers tend to be used to perform bit-to-bit copy of the original medium [12].

Analysis of volatile information can provide large amount of information not present in traditional procedures, such as access to documents in the cloud or passwords. This prompts a question: When to perform a live analysis or a traditional one?

Live Forensics or traditional approach

There are different scenarios where, inevitably, it is necessary to perform a live analysis. This is the case of critical systems where the interruption of their normal operation would be of an enormous cost. Another feasible scenario is where the urgency of the analysis prevails, where not all the time necessary to acquire evidence is available.

Encryption of information, in post-mortem analysis, represents a total impediment. In this case it is necessary to perform a Live Forensics, prior a verification of whether, in fact, there is encrypted information in the system. So that the encryption keys used are not lost.

Another case where Live Forensics is needed is malware analysis. Where the malicious code only resides in memory, being non-persistent.

Based on the foregoing, it is reasonable to conclude that there is no unanimous decision about the need for Live Forensics (except in the inevitable) but a specific assessment of each case.

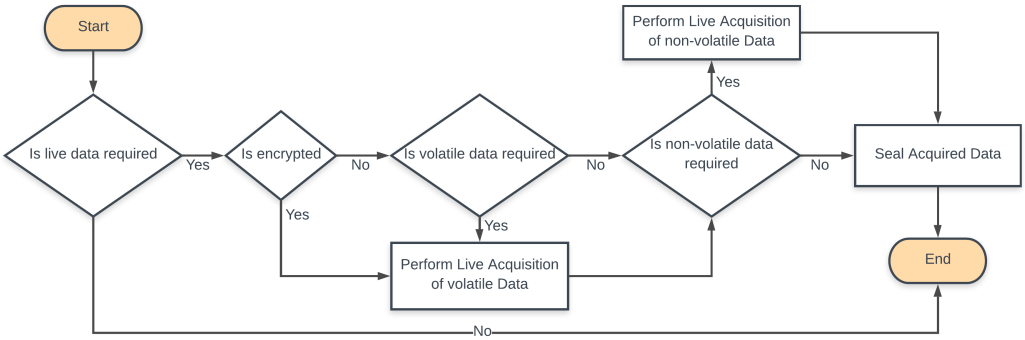

Figure 3.3: Simplified Guidelines for Data Acquisition [13]

3.2.3 Mobile Forensics

Following the line of computer forensics, the rise of mobile devices fosters a parallel branch: mobile forensics. Understanding mobile device as not only mobile phones but any portable system or machine with communication.

These devices, having more mobility than their traditional counterparts, have a more personal nature and link the user in their displacements. This results in a greater amount of sensitive and personal information. Such as locations, messages, photos, social networks, email and the like.

Lacking such a well-established trajectory as computer forensics, there is not as much standardization in forensic procedures. There are certain guidelines like the one exposed by NIST [14], but since this is not the objective of the project, it will not be deepened.

3.2.4 Network Forensics

Network Forensics is the capture, recording, and analysis of network events in order to be able to determine and discover the sources of attacks or incidents.

This branch, unlike the rest of variants, has a different approach, not so much a post-mortem analysis but a proactive methodology. It bases its behaviour on detecting and preventing cyberattacks and crime.

For this purpose, different tools and techniques present in other fields of cybersecurity (defensive stance) are used.

- Anomaly detection: patterns of the normal behaviour of the network are created so any disturbance and mismatch is evaluated and monitored.
- Signature scan: signatures and hashes of malicious activity are periodically checked within network traffic and, in case of matches, alerted.
- Intrusion Detection System (IDS): Monitoring tool deployed in strategic nodes of the network, analysing all traffic in it and comparing with a preloaded set of rules.
- Access Control List (ACL): Prevent traffic with certain headers matching preloaded rules.
- Honeypots: Fake networks to deceive attackers. Emulating a real environment so that intruders are monitored and studied, revealing their tactics and techniques without jeopardizing real infrastructure and assets.

3.2.5 Database Forensics

Database forensics consists of the forensic study dedicated to databases and their information. Since database servers are, by design, traditional computers there is similarity in treatment and approach but focused on databases.

It consists of examining records and logs, in order to check transactions and users. It is also related to mobile forensics and databases such as SQLite.

3.3 Forensic Methodology

There are numerous guides and standardized procedures for computer forensics as it is the original field. In this study, two of the main references in evidence-gathering procedures will be taken into account: RFC-3227 [15] and ISO/IEC-27037 [13].

There are certain common steps to follow during an analysis, regardless of technology or crime. Although it is necessary to emphasize that there are certain differences between those investigations that originate from a judicial sphere and those that originate from private initiative and, especially, in the DFIR field that concerns us.

Therefore, we will obviate the judicial aspects of a forensic investigation, of great relevance in the judicial scope, and we will detail the other stages common to all investigations.

It is necessary to emphasize that, regardless of where the motivation of the investigation comes from, it is an indispensable requirement that there be an

agreement between the competent and responsible parties so that, legally, the analysis can be carried out.

The phases to be addressed during an investigation are as follows:

- Physical Crime Scene Securing
- Identification (3.3.1)
- Collection and Acquisition (3.3.2)
- Analysis (3.3.3)
- Reporting (3.3.4)

3.3.1 Assessment and Scenarios

The first phase of the forensic investigation itself. It consists of searching, identifying potential sources of information and cataloguing them. To this end, all devices that may contain evidence must be identified.

In addition, according to RFC-3227 [15], it is necessary to take into consideration the order of volatility (from more volatile to less volatile) when identifying and collecting, thus determining the urgency at the time of treatment and avoiding the loss of information. These levels of urgency are, in order:

1. Registers, cache
2. Routing table, ARP cache, process table, kernel statistics, memory
3. Temporary file systems
4. Disk
5. Remote logging and monitoring data that is relevant to the system in question
6. Physical configuration, network topology
7. Archival media

3.3.2 Evidence and Artifact Extraction

The extraction process is critical and fundamental during a forensic investigation, since the entire investigation is based on this and its results. It is necessary that the procedure followed is documented in detail and in a transparent manner, so that the process is as standardized as possible and guided (minimizing errors during extraction).

There are two terms referring to evidence extraction: acquisition and collection.

Collection is the physical obtaining of the devices and media holding the evidence, removing them from their original and documented location.

Acquisition is the extraction of the information contained in the physical devices and media. Obtaining a digital image or forensic image of the evidence.

As mentioned in the section (3.2.2) it is necessary to determine which scenario is contemplated and how to act. In some cases it is not possible to collect evidence due to volatility. However, in case the system is on, both acquisition and collection will be conducted.

In DFIR domain it is common, due to time management in incident response, to first perform a triage of the affected machine, acquiring the most relevant immediate information. In order to assess the severity of the incident or threat and proceed as appropriate. In the event that the severity of the incident or threat is considerable and the scope has not been determined with this initial triage, a more complete forensic analysis would be carried out.

For the extraction, in the case of Live Forensics, it is necessary to have access to the system (unlocked). If so, the main concern must be to minimize the contamination of the environment.

In that case, the main concern must be to minimize pollution of the environment. A removable medium will be used with the necessary tools to carry out the acquisition, usually a memory dump.

If, on the other hand, the system is blocked, it would be necessary to access it through other techniques (exploits). However, this type of methodology is based on duly substantiated and accredited legal bases.

RFC-3227 [15] establishes a series of recommendations when performing live extraction.

- Since volatile data can be lost do not shut down the device until the evidence has been extracted. When doing so the system should be carefully disconnected, checking if no scripts or programs can destroy the evidence.
- Don not trust pre-installed programs on the system. Programs can be altered and only should run controlled evidence gathering programs from appropriately protected media.
- Don not run MAC times altering programs.
- Acquisition should be done in previously mentioned order (3.3.1).

3.3.3 Analysis

In-depth analysis of previously extracted evidence. Determine the importance of the information and reconstruct facts and formulate conclusions based on the evidence. The reference standard for forensic analysis is ISO/IEC-27042 [16].

NIST also establishes a series of recommendations or guidelines when conducting a forensic investigation [17]:

- A methodical approach must be followed. The entire procedure must be accurately documented in order to trace and clarify all conclusions (or their absence) drawn.
- Preserve the original files and work on copies so as not to alter any original evidence.
- The reliability and fidelity of the sources should be evaluated. Primacy of original sources over previously treated information.
- To consider file types according to the headers and content, never by the indicated extension, being easily manipulable.
- Prioritize research and objectives. Certain scenarios, such as the attribution of authorship, entail a great cost while they do not suppose improvement in the resolution of the threat or incidence itself.
- Consider the logistical and technical complexity of the analyses. The enormous amount of information and its fragility prevents, in certain cases, its treatment from being feasible or practical.
- The different sources of information must be integrated in order to contrast and correlate the different events or evidences.

3.3.4 Report

Finally, once the conclusions have been drawn, it is necessary to explain them as well as all the procedures and methodologies followed. This report must, without exception, be written in a plain and accessible format and language, avoiding technical terminology. In such a manner that people outside this field can easily understand it. If necessary to include technical terms, they should be referenced and annotated separately.

Many forensic tools come with a built-in reporting feature that typically follows predefined templates and may allow customization of the report structure. However, depending on the data and type of information handled, it may not be compatible with a fixed report.

The report has a predefined and consistent structure, we will adhere to the one exposed by the NIST [14]. Including:

- Identity of the reporting agency
- Case investigator
- Identity of the submitter
- Date of evidence receipt
- Date of report
- Descriptive list of items submitted for examination, including serial number, make, and model
- Identity and signature of the examiner
- The equipment and set up used in the examination
- Brief description of steps taken during examination, such as string searches, graphics image searches, and recovering erased files
- Supporting materials such as printouts of particular items of evidence, digital copies of evidence, and chain of custody documentation
- Details of findings
- Report conclusions

Should the case be defended in court, the clarity and objectivity of this document will be of critical importance.

Chapter 4

Case Study: Non-volatile Memory Analysis

Once the Digital Forensics and Incident Response (DFIR) fundamentals are understood, it is intended to perform a forensic analysis, using open-source tools, of two scenarios that comprise the main areas of digital forensics: Computer Forensics and Live Forensics.

The first of them an analysis of Data Leakage Case [1] proposed by NIST. Covering computer forensics.

For the second one, volatile memory analysis, use will be made of a scenario proposed by the Centro Criptológico Nacional Computer Emergency Response Team (CCN-CERT) on its Atenea platform, where a memory dump will be provided. Covering live forensics.

4.1 Preparation

Due to the nature of the scenario and project proposed, it is not possible to fully simulate or detail the cybersecurity infrastructure and plans. Therefore, all procedures will be considered to have been properly declared and documented, monitoring tools are fully operational, and sufficient trained personnel are available to deal with any threat and incident.

4.2 Detection and Analysis

As stated in the forensic methodology (Section 3.3), forensic analysis should be performed in four phases. However, due to the approach of the practical case, where it has already been identified, prioritized and acquired, and is posed from a didactic point of view, it has been decided to merge the analysis and reporting phase, structuring it into sections that deal with or group together different concepts and forensic techniques.

4.2.1 Scenario Context

When carrying out forensic analysis, whether within the DFIR framework or a traditional investigation, it is essential to have a general or introductory context about the problematic or incident we are facing. Specifically, for this research, the following scenario is presented.

National Institute of Standards and Technology (NIST) provides the so-called Data Leakage Case [1]. Which, as can be inferred from its title, comprises an information leakage case in a company.

We have been informed that an employee named "Mr. Informant" is suspected of leaking information after he attempted to sneak out storage devices, detected at the company's security checkpoint.

The company has the following security policy:

- Confidential electronic files should be stored and kept in the authorized external storage devices and the secured network drives.
- Confidential paper documents and electronic files can be accessed only within the allowed time range from 10:00 AM to 16:00 PM with the appropriate permissions.
- Non-authorized electronic devices such as laptops, portable storages, and smart devices cannot be carried onto the company.
- All employees are required to pass through the 'Security Checkpoint' system.
- All storage devices such as HDD, SSD, USB memory stick, and CD/DVD are forbidden under the 'Security Checkpoint' rules.

A total of one personal computer and three removable disks have been intervened on which to perform a forensic analysis. Detailed below:

Personal Computer	Type	Virtual System
	CPU	1 Processor (2 Core)
	RAM	2,048 MB
	HDD Size	20 GB
	File System	NTFS
	IP Address	10.11.11.129
	Operating System	Microsoft Windows 7 Ultimate (SP1)
Removable Media 1	Type	USB removable storage device
	Serial No.	4C530012450531101593
	Size	4 GB
	File System	exFAT
Removable Media 2	Type	USB removable storage device
	Serial No.	4C530012550531106501
	Size	4 GB
	File System	FAT32
Removable Media 3	Type	CD-R
	Size	700 MB
	File System	UDF

Table 4.1: Target Systems for NIST Case Study [1]

Removable Media 1 is an authorized USB memory stick for managing confidential electronic files of the company. In order to maintain a structure during the analysis, the questions posed by the NIST [1] case itself will be followed and answered in sequential order.

NIST presents the challenge through sequential and structured questions. Due to the approach of this project, which aims to provide an overview of the forensic techniques used (and not an exhaustive execution of the proposal), it has been opted to cluster and select the questions and content that best fit this objective. Therefore, the script will not be followed thoroughly but will be adapted to this project.

4.2.2 Evidence Integrity

The first step to undertake, before proceeding with the analysis, would be to verify the hashes of memory dumps. If those obtained at the time of acquisition (and therefore the originals) and those corresponding to the images we have are identical. Meaning that the evidence has not been altered.

In this case we do not dispose of the hash of the integral images but rather we are provided with the hash of the compressed parts of the images. For this reason, we will only check that these hashes correspond to those of the downloaded files.

File Name	SHA-1
cfreds_2015_data_leakage_pc.7z.001	F07632FAA66A47088DEB07BDB45CC568E4BF650B
cfreds_2015_data_leakage_pc.7z.002	5DEE46ABF6FA833268E5AE199A13854CCF42689B
cfreds_2015_data_leakage_pc.7z.003	1687686F819092E05047F195F102D8FA0C38ED66
cfreds_2015_data_leakage_rm#1.E01	FFD0F3CBA3DFE3291F786B845A06A8AA56C1CD8C
cfreds_2015_data_leakage_rm#2.7z	DDFE97AA3D8D0B33CC6092123090A8154945F38E
cfreds_2015_data_leakage_rm#3_type2.7z	AE26235F6FB5EDDFFB670DD060EF109EDA91EB8F

Table 4.2: Hash List for NIST Case Study [1]

We use the sha1sum program present in Linux to obtain the SHA-1 hash of each of the files. By placing ourselves in the directory where we store these files and executing the command *sha1sum ./**. Resulting in the following output:

File Name	SHA-1
./cfreds_2015_data_leakage_pc.7z.001	F07632FAA66A47088DEB07BDB45CC568E4BF650B
./cfreds_2015_data_leakage_pc.7z.002	5DEE46ABF6FA833268E5AE199A13854CCF42689B
./cfreds_2015_data_leakage_pc.7z.003	1687686F819092E05047F195F102D8FA0C38ED66
./cfreds_2015_data_leakage_rm#1.E01	FFD0F3CBA3DFE3291F786B845A06A8AA56C1CD8C
./cfreds_2015_data_leakage_rm#2.7z	DDFE97AA3D8D0B33CC6092123090A8154945F38E
./cfreds_2015_data_leakage_rm#3_type2.7z	AE26235F6FB5EDDFFB670DD060EF109EDA91EB8F

Table 4.3: SHA-1 Check-list Verification

Comparing both hashes we can verify that the integrity of the images has been maintained and therefore they have not been altered.

4.2.3 First Approach

For the analysis of the information available, the open-source forensic tool Autopsy will be mainly used as a reference. It allows the execution of different plug-ins and options, however, at first, we will only load the disk image with the default options and ingest modules offered.

Beginning with the personal computer, the image corresponds to a 20GB disk in NTFS format consisting of a total of four partitions. The first and last unallocated, the second for boot and the third for storing information and the operating system. The latter being the one relevant for research.

Name	ID	Starting Sector	Length in Sectors	Description	Flags
vol1 (Unallocated: 0-2047)	1	0	2048	Unallocated	Unallocated
vol2 (NTFS / exFAT (0x07): 2048-206847)	2	2048	204800	NTFS / exFAT (0x07)	Allocated
vol3 (NTFS / exFAT (0x07): 206848-41940991)	3	206848	41734144	NTFS / exFAT (0x07)	Allocated
vol4 (Unallocated: 41940992-41943039)	4	41940992	2048	Unallocated	Unallocated

Figure 4.1: PC Study Case Disk Partitions

The various hives belonging to HKEY_LOCAL_MACHINE are stored under *C:\Windows\System32\config*. Including:

- HKEY_LOCAL_MACHINE\SYSTEM: *\system32\config\system*
- HKEY_LOCAL_MACHINE\SAM: *\system32\config\sam*
- HKEY_LOCAL_MACHINE\SECURITY: *\system32\config\security*
- HKEY_LOCAL_MACHINE\SOFTWARE: *\system32\config\software*

These records, contained in the above-mentioned files (hives), shall contain registries and information relating to the configuration and settings of the computer. For the analysis, hives are firstly exported with Autopsy and then parsed with RegRipper to extract registry keys.

We note that the computer belongs, indeed, to Mr Informant. With Windows 7 Ultimate and Service Pack 1 installed. It was first installed on Sun Mar 22 14:34:26 2015 (UTC). It can be extracted from the hive file *SOFTWARE*.

Microsoft\Windows NT\CurrentVersion
LastWrite Time Sun Mar 22 15:21:53 2015 (UTC)
RegisteredOrganization :
CurrentVersion : 6.1
CurrentBuild : 7601
CurrentBuildNumber : 7601
CSDBuildNumber : 1130
SoftwareType : System
InstallationType : Client
EditionID : Ultimate
RegisteredOwner : informant
SystemRoot : C:\Windows
PathName : C:\Windows
CSDVersion : Service Pack 1
ProductName : Windows 7 Ultimate
CurrentType : Multiprocessor Free
ProductId : 00426-292-0000007-85262
BuildLab : 7601.win7sp1_gdr.130828-1532
InstallDate : Sun Mar 22 14:34:26 2015 (UTC)

Table 4.4: CurrentVersion Registry

4.2.4 Time Zone

When conducting a forensic investigation it is vitally important to understand the time zone of the system being analysed in order to correctly determine the time stamps of possible evidence.

In the SYSTEM hive it is possible to find the registry TimeZoneInformation which provides us with the time zone of the computer.

TimeZoneInformation key
ControlSet001\Control\TimeZoneInformation
LastWrite Time Wed Mar 25 10:34:25 2015 (UTC)
DaylightName ->@tzres.dll,-111
StandardName ->@tzres.dll,-112
Bias ->300 (5 hours)
ActiveTimeBias ->240 (4 hours)
TimeZoneKeyName->Eastern Standard Time

Table 4.5: TimeZoneInformation Registry

It has different fields. The bias is the difference, in minutes, with respect to Universal Time Coordinated (UTC), in the same way that the ActiveTime-Bias is the difference, in minutes, with respect to UTC regardless of daylight saving.

Both DayLightName and StandardName are referred to tzres.dll instead of the actual name. Checking this library we can see that code 111 corresponds to Eastern Daylight Time and code 112 to Eastern Standard Time.

Therefore, and based on the registries, we can determine that the time zone of the computer being analysed is Eastern Standard Time. Nonetheless, it is possible that the time stamps are not always shown with this format but that UTC is used, although always specified.

4.2.5 Users

In the SAM register (Security Account Manager) we find all the user accounts in the system. Among the registered data are the name, number of logins, last access. The system accounts, without relevance, have been filtered. We can see that the last login was done by 'informant' on Wed Mar 25 14:45:59 2015 Z

Username : informant [1000]
SID : S-1-5-21-2425377081-3129163575-2985601102-1000
Full Name :
User Comment :
Account Type : Default Admin User
Account Created : Sun Mar 22 14:33:54 2015 Z
Name :
Password Hint : IAMAN
Last Login Date : Wed Mar 25 14:45:59 2015 Z
Pwd Reset Date : Sun Mar 22 14:33:54 2015 Z
Pwd Fail Date : Wed Mar 25 14:45:43 2015 Z
Login Count : 10
–>Password does not expire
–>Password not required
–>Normal user account
Username : admin11 [1001]
SID : S-1-5-21-2425377081-3129163575-2985601102-1001
Full Name : admin11
User Comment :
Account Type : Default Admin User
Account Created : Sun Mar 22 15:51:54 2015 Z
Name :
Last Login Date : Sun Mar 22 15:57:02 2015 Z
Pwd Reset Date : Sun Mar 22 15:52:10 2015 Z
Pwd Fail Date : Sun Mar 22 15:53:02 2015 Z
Login Count : 2
–>Password does not expire
–>Normal user account

Table 4.6: PC Users List Fragment

4.2.6 Shutdown

In order to know the last shutdown of the computer we resort to the events of the Windows system. Located in:

C:\Windows\System32\Winevt\Logs\System.evtx

We export this file and load it with the Windows Event Viewer. We filter the events by the following ID: 1074, 6006, 6008. Corresponding to system shut-down behaviours.

Información	25/03/2015 16:31:00	EventLog	6006	Ninguno
Información	25/03/2015 16:30:58	USER32	1074	Ninguno
Información	25/03/2015 16:30:55	USER32	1074	Ninguno
Información	24/03/2015 22:07:27	EventLog	6006	Ninguno
Información	24/03/2015 22:07:25	USER32	1074	Ninguno
Información	24/03/2015 22:07:23	USER32	1074	Ninguno

Figure 4.2: Windows Shutdown Events

As can be seen in the last shutdown of the equipment the user started the power off of the equipment (1074) at 25/03/2015 16:30:58 and then stopped the event service (6006) at 25/03/2015 16:31:00.

4.2.7 Network Interface

Following a procedure similar to all the above, the network interfaces are extracted. In the hive SYSTEM we can find the register Tcp Interfaces.

Interface E2B9AEEC-B1F7-4778-A049-50D7F2DAB2DE
Name: Local Area Connection
Control\Network key LastWrite time Sun Mar 22 14:35:09 2015 (UTC)
Services\Tcpip key LastWrite time Wed Mar 25 15:24:51 2015 (UTC)
DhcpDomain = localdomain
DhcpIPAddress = 10.11.11.129
DhcpSubnetMask = 255.255.255.0
DhcpNameServer = 10.11.11.2
DhcpServer = 10.11.11.254

Table 4.7: TCP Interfaces Registry

We can see that the system uses a Dynamic Host Configuration Protocol (DHCP) server to obtain the IP address (10.11.11.129).

4.2.8 Programs

For the verification of the applications installed in the equipment we resort to the hive SOFTWARE, in the registry Microsoft\Windows\CurrentVersion\Uninstall we can find the programs installed, their date and version.

Wed Mar 25 14:57:31 2015 (UTC)
Eraser 6.2.0.2962 v.6.2.2962
Wed Mar 25 14:54:33 2015 (UTC)
Microsoft .NET Framework 4 Extended v.4.0.30319
...
Mon Mar 23 20:00:58 2015 (UTC)
Bonjour v.3.0.0.10

Table 4.8: Uninstall Registry Segment

There are several programs that could be investigated, such as *Eraser*, *Bojour*, *Apple Software Update* or *Google Chrome Update Helper*.

The execution of programs in Windows is registered in different ways. One of them is to access NTUSER hive. This file records the user's information and is located in *C:\Users\<user>\NTUSER.DAT*.

Registry \CurrentVersion\Explorer\UserAssist tracks every GUI-based program launched from the desktop, date and count. Allowing us to know the programs executed by the user.

Path values are ROT-13 (Caesar cipher) encoded:

- 6D809377 - Program Files x64
- 7C5A40EF - Program Files x86
- 1AC14E77 - System

Wed Mar 25 15:28:47 2015 Z
{1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\xpsrchvw.exe (1)
Wed Mar 25 15:24:48 2015 Z
{6D809377-6AF0-444B-8957-A3773F02200E}\Microsoft Office\Office15\WINWORD.EXE (4)
Wed Mar 25 15:21:30 2015 Z
{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\Google\Drive\googledrivesync.exe (1)
Wed Mar 25 15:15:50 2015 Z
{6D809377-6AF0-444B-8957-A3773F02200E}\CCleaner\CCleaner64.exe (1)
Wed Mar 25 15:12:28 2015 Z
{6D809377-6AF0-444B-8957-A3773F02200E}\Eraser\Eraser.exe (1)
Wed Mar 25 14:57:56 2015 Z
C:\\informant\Desktop\\ccsetup504.exe (1)
Wed Mar 25 14:50:14 2015 Z
C:\Users\informant\Desktop\Download\Eraser 6.2.0.2962.exe (1)

Table 4.9: UserAssist Registry Segment

4.2.9 Hours of Use

In the same way that we can know the start and shutdown of the computer through Windows events we can know both the login and logoff of users. These events are in the same path, *C:\Windows\System32\Winevt\Logs*, but they are stored in the Security.evtx file instead.

We filter by event ID 4624, 4647, 4672. Corresponding to Logon, Logoff and Special Logon respectively.

Información	25/03/2015 16:30:57	Microsoft Windows security auditing.	4647	Logoff
Información	25/03/2015 16:18:54	Microsoft Windows security auditing.	4624	Logon
Información	25/03/2015 16:18:54	Microsoft Windows security auditing.	4672	Special Logon
Información	25/03/2015 15:57:18	Microsoft Windows security auditing.	4672	Special Logon
Información	25/03/2015 15:57:18	Microsoft Windows security auditing.	4624	Logon
Información	25/03/2015 15:57:18	Microsoft Windows security auditing.	4672	Special Logon

Figure 4.3: Windows Login/Logoff Events

The time stamps shown correspond to the time zone of the system being analysed. Therefore we can verify that the user informant uses the computer from 11:15 to 16:30.

4.2.10 Web Artifacts

As for the browsers used by the user these can be known by studying the programs installed and executed, as previously mentioned. In addition, using the Autopsy tool, we can check the browsers used and their data in the web artifacts extracted. Specifically, 'Mr Informant' user uses Google Chrome and Internet Explorer.

Google Chrome and Internet explorer history can be found in:

- *C:\Users\informant\AppData\Local\Google\Chrome\User Data\Default\History*
- *C:\Users\informant\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\index.dat*

We can observe that there are different suspicious records, suggesting intentionality to commit a data leak and techniques to use. It also suggests that cloud storage may have been used for this purpose:

Web Title	Date Accessed
Tools:Data Recovery - ForensicsWiki	2015-03-23 19:19:21
DEFCON-20-Perklin-AntiForensics.pdf	2015-03-23 19:18:00
cloud storage - Google Search	2015-03-23 19:06:27
FBI — Intellectual Property Theft	2015-03-23 19:05:55
how to leak a secret - Google Search	2015-03-23 19:05:48
intellectual property theft - Google Search	2015-03-23 19:05:22
data leakage methods - Google Search	2015-03-23 19:02:09

Table 4.10: Suspicious Web History Sample

4.2.11 Mail

As we can see from the programs executed by the user (4.2.8), the email client used is Outlook :

{6D809377-6AF0-444B-8957-A3773F02200E}\Microsoft Office\Office15\.EXE (5)

Outlook stores copies of the exchange server offline in order to consult the mail in case there is no internet connection. This OST file is located under the path *C:\Users\informant\AppData\Local\Microsoft\Outlook*. The corresponding file with 'Mr Informant' account is *iaman.informant@nist.gov.ost*. Therefore, the user's email account is iaman.informant@nist.gov.

With the OST Viewer application we consult this file. Only three folders contain relevant information: Inbox, Sent Items and Deleted Items.

	spy	Hello, Iaman	iaman;	23/03/2015 18:29:27	23/03/2015 18:29:29	13
	spy	Good job, buddy.	iaman;	23/03/2015 20:14:58	23/03/2015 20:15:00	13
	spy	RE: Good job, buddy.	iaman;	23/03/2015 20:20:41	23/03/2015 20:20:41	16
	spy	Important request	iaman;	23/03/2015 20:26:22	23/03/2015 20:26:23	13
	spy	Last request	iaman;	24/03/2015 14:25:57	24/03/2015 14:25:59	13

Figure 4.4: Inbox Mails

	spy	RE: It's me	iaman;	23/03/2015 21:41:19	23/03/2015 21:41:22	16
	iaman	RE: Last request	spy;	24/03/2015 14:35:10	24/03/2015 14:35:00	8
	iaman	RE: Watch out!	spy;	24/03/2015 20:34:02	24/03/2015 20:34:00	5
	iaman	Done	spy;	24/03/2015 22:05:09	24/03/2015 22:05:00	5

Figure 4.5: Deleted Mails

Figure 4.6: Sent Mails

Analysing the emails and their content we see that it is an exchange of emails between 'spy' and 'Mr Informant'. The former requests information and gives recommendations to the latter that reports on his progress.

No files attached are detected, however, in one of the deleted emails with subject 'RE: It's me' from spy to iaman two links to Google Drive are included in the message body. What appear to be two files: 'happy_holiday.jpg' and 'do_u_wanna_build_a_snow_man.mp3'.

If we download these files and check their type of file using the Linux command *file* it indicates that they are:

./do_u_wanna_build_a_snow_man.mp3:	Microsoft PowerPoint 2007+
./happy_holiday.jpg:	Microsoft Excel 2007+

However, once converted to ppt and xlsx, we find that they are actually two Office documents related to a secret project.

4.2.12 USB History

To check the devices connected to the system we use the USB Historian tool. First we extract the SYSTEM and SOFTWARE hives of Windows and the NTUSER.DAT hive of the informant user.

Once loaded and analysed by the tool, it indicates that two USB have been connected:

Name	Serial No	Usb Stor DateTime
SanDisk Cruzer Fit USB Device	4C530012450531101593	24/03/2015 13:38:00
SanDisk Cruzer Fit USB Device	4C530012550531106501	24/03/2015 13:58:33

Table 4.11: USB History

4.2.13 Network Drives

Registry *Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2* holds information about network drives and their IP addresses. This record is accessible from NTUSER.DAT hive.

We can verify that the informant user mounts a network drive on the private IP address 10.11.11.128:

Remote Drives:
Mon Mar 23 20:26:04 2015 (UTC)
##10.11.11.128#secured_drive

Table 4.12: Mount Points Registry

When navigating through a directory, the paths traversed are recorded in the registry, keeping Windows a journal of folders. This information can be found in the files NTUSER.DAT and UsrClass.dat under the path:

Users\Informant\AppData\Local\Microsoft\Windows

We use ShellBags Explorer to visualize in a more comfortable way these registries, previously extracted with Autopsy. We can verify that, inside the network drive, the user has navigated through the following folders. With the resulting tree:

- 10.11.11.128\secured_drive
  - Past Projects
  - Common Data
  - Secret Project Data
    - pricing decision
    - final
    - progress
    - proposal
    - technical review
    - design

4.2.14 Cloud Storage

Since we have previously found traces indicating user's usage of Google Drive we proceed to ensure whether there is anything of relevance. The folder synchronized with the Google Drive server is located in *C:\Users\informant\Google Drive*.

Browsing with Autopsy we can verify that there was only one file and it has been deleted. However, by name, it appears to be the same file received by mail (4.2.11).

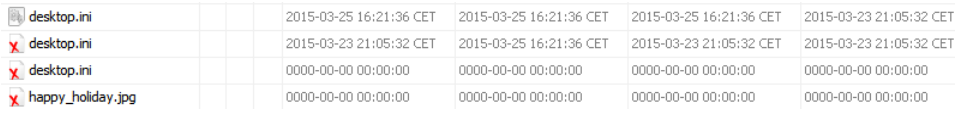

Figure 4.7: Google Drive Directory

Under *C:\Users\informant\AppData\Local\Google\Drive\user_default* we find the following deleted files: sync_config.db and snapshot.db.

These files contain important account information. However, when exporting them with Autopsy or FTK Imager they are not detected as valid for SQLiteStudio. Nonetheless, parsing their strings we can extract the user: iaman.informant.personal@gmail.com. And a record of the following files:

- C:\Users\informant\Google Drive\happy_holiday.jpg
- C:\Users\informant\Google Drive\do_u_wanna_build_a_snow_man.mp3
- C:\Users\informant\Google Drive\happy_holiday.jpgG

4.2.15 Evidence Correlation

When looking at the user's desktop we find a Word document (docx) named *Resignation_Letter_(Iaman_Informant)*. This is a letter of resignation from Iaman Informant to his Manager indicating that, from 25 March 2015, he will no longer work for the company.

This file has the following time stamps:

Modified	2015-03-24 19:59:30
Accessed	2015-03-24 19:59:30
Created	2015-03-24 19:48:40
Changed	2015-03-24 19:59:30

Therefore, and based on the time stamps of all the files analysed, we can assume that 'Mr Informant', once the data leakage was finished, intended to leave his job in the company.

In addition, in the same Desktop directory, we find an xps file. This Microsoft's proprietary format defines the appearance of a document for printing. Therefore the resignation letter was printed, according to time stamps, on March 25 at 16:28, one day after the creation of the document.

4.2.16 Thumbcache

Microsoft Windows stores thumbnail images for Windows Explorer when a user switches a folder to thumbnail mode or views pictures via a slide show.

There are several files, depending on the resolution and size, under:

C:\Users\informant\AppData\Local\Microsoft\Windows\Explorer

We extract thumbcache_256.db file from the directory using Autopsy and open it with Thumbcache Viewer in order to visualize the images.

Filename	Cache Entry Offset	Cache Entry S...	Data Offset	Data Size	Data Checksum	Header Checksum
::{645FF040-5081-101...	24 B	0 KB	152 B	0 KB	0000000000000000	4d8b419f9128fe1d
::{645FF040-5081-101...	152 B	0 KB	280 B	0 KB	0000000000000000	4d8b419f9128fe1d
501f0f481ce56b76.png	280 B	2 KB	360 B	2 KB	fd6b0381c05983be	c4547cee07c7ebd1
e040e320d1ccaaf7.png	2565 B	4 KB	2645 B	4 KB	9f7e4f05a24d3cdf	9e60f944168113bb
edced76bb795254b.png	7376 B	2 KB	7456 B	2 KB	28f90df3389ebd93	7c62e5ec1657d925
a22e9d71d3aad949.png	9644 B	2 KB	9724 B	2 KB	e5f08fd7db3de60f	01363539944097ed

Figure 4.8: Thumbcache File Sample

After examining the thumbcache we see that the user has opened and visualized Microsoft Office presentations about secret projects.

4.2.17 Windows Search

The Windows Search database is a Windows Search service file, which provides content indexing, property caching, and search results.

It is stored in the file Windows.edb under:

C:\ProgramData\Microsoft\Search\Data\Applications\Windows

This database can be of great use when it comes to forensic analysis. It can help with:

- Partial recovery of the content of indexed documents or email messages
- Indicate the former existence of files
- Time-line analysis

We export the Windows.edb file with Autopsy in order to open it with the Nirsoft ESEDatabaseView tool.

It has a total of 8 tables.

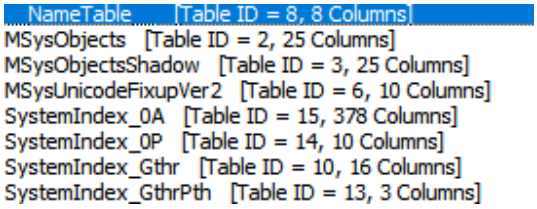

Figure 4.9: Windows Search Database Tables

First, we analyse indexed email communications. To do so, we add a filter in ESEDatabaseView tool to allow us to show only items that include the string 'E-mail'. Only the table SystemIndex_0A contains matching rows.

We export these filtered results to a CSV file and clean the columns without information to improve their legibility. A total of 18 rows.

If we apply a filter again to show the columns that provide relevant information we have the following result:

```
spy;Inbox`;Outlook E-Mail Message@□;Hello, Iaman0;Hello, Iaman0;E-mail
spy;Inbox`;Outlook E-Mail Message@□;Last request0;Last request0;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Last request□;Last request0;E-mail
spy;Inbox`;Outlook E-Mail Message@□;RE: Last request□;Last request0;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Hello, Iaman□;Hello, Iaman0;E-mail
spy;Inbox`;Outlook E-Mail Message@□;Good job, buddy.□;Good job, buddy.□;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Last request□;Last request0;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Good job, buddy.0;Good job, buddy.□;E-mail
spy;Inbox`;Outlook E-Mail Message@□;RE: Good job, buddy.0;Good job, buddy.□;E-mail
spy;Inbox`;Outlook E-Mail Message@□;Important request□;Important request□;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Important request`;Important request□;E-mail
iaman;Sync Issues□;Outlook E-Mail Message@□;Synchronization Log: `;Synchronization Log: `;E-mail
spy;Inbox`;Outlook E-Mail Message@□;Watch out!;Watch out!;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;RE: Watch out!@□;Watch out!;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;It's me;It's me;E-mail
spy;Inbox`;Outlook E-Mail Message@□;RE: It's me□;It's me;E-mail
iaman;Sent Items;Outlook E-Mail Message@□;Done0;Done0;E-mail
```

Figure 4.10: Filtered Windows Search Email Analysis

As can be seen, this is an index of all emails from Iaman's Exchange server, including synchronization errors.

Likewise, we check the string '\Users\informant\Desktop\' as it is one of the directories most used by Iaman. Exporting and filtering, again, the CSV file.

```
System_ItemFolderPathDisplay;System_ItemName;System_ItemType
C:\Users\informant\Desktop;desktop.ini□;.ini0
C:\Users\informant\Desktop\Download□;IE11-Windows6.1-x64-en-us.exe`;.exe0
C:\Users\informant\Desktop;Resignation_Letter_(Iaman_Informant).docx□;.docx`
C:\Users\informant\Desktop;Google Drive.lnk□;
C:\Users\informant\Desktop\temp;IE11-Windows6.1-x64-en-us.exe`;.exe0
C:\Users\informant\Desktop\temp;Chrysanthemum.jpg□;.jpg0
C:\Users\informant\Desktop\temp;Desert.jpg;.jpg0
C:\Users\informant\Desktop\temp;Hydrangeas.jpg@□;.jpg0
C:\Users\informant\Desktop\temp;Jellyfish.jpg`;.jpg0
C:\Users\informant\Desktop\temp;Koala.jpg□;.jpg0
C:\Users\informant\Desktop\temp;Lighthouse.jpg@□;.jpg0
C:\Users\informant\Desktop\temp;Penguins.jpg0;.jpg0
C:\Users\informant\Desktop\temp;Tulips.jpg;.jpg0
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\design□;[secret_project]_detailed_design.pptx`;.pptx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\final;[secret_project]_final_meeting.pptx□;.pptx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\pricing decision;(secret_project)_market_analysis.xlsx`;.xlsx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\pricing decision;(secret_project)_market_shares.xls;.xls0
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\pricing decision;(secret_project)_price_analysis_#1.xlsx;.xlsx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\proposal;[secret_project]_detailed_proposal.docx;.docx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\proposal;[secret_project]_proposal.docx@□;.docx`
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\design□;space_and_earth.mp4□;.mp40
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\design□;winter_storm.amr□;.amr0
C:\Users\informant\Desktop\S data\Secret Project Data\Secret Project Data\design□;winter_whether_advisory.zip□;.zip0
```

Figure 4.11: Filtered Windows Search Desktop Analysis

Therefore there is a difference between the files indexed on the desktop and those that can be retrieved from the Desktop path itself in the disk image (4.2.15). They have been deleted and cannot be accessed.

4.2.18 Volume Shadow Copies

The Volume Shadow Copy Service is a technology provided by Microsoft Windows to automatically perform volume backups, even while applications are running [18].

Creates snapshots of the data, Shadow Copies, which allow you to retrieve the information or restore a previous point.

In the registry *CurrentControlSet\Control\BackupRestore* there are exclusion lists of: snapshots, backup and registry keys. So the data mentioned here will not be present in the Shadow Copies.

In order to access the Shadow Copies of the image it is necessary to perform a series of previous steps.

It is necessary to convert the memory image in raw format (dd) to a Virtual Hard Disk (VHD). Using vhdtool we convert this image: *VhdTool.exe /convert cfreds_2015_data_leakage_pc.dd*

Status: Converting "cfreds_2015_data_leakage_pc.dd" to a fixed format VHD.
Status: Attempting to open file "cfreds_2015_data_leakage_pc.dd"
Status: File opened, current size is 21474836480
Status: Performed seek to end of file.
Status: VHD footer generated.
Status: VHD footer appended.
Status: Complete

Table 4.13: VHD Conversion Output

As can be seen, the conversion is nothing more than the addition of headers VHD to the image. Therefore, once finished, we only rename the file extension: cfreds_2015_data_leakage_pc.vhd.

Using Windows Disk Management we mount the VHD image previously created as read-only.

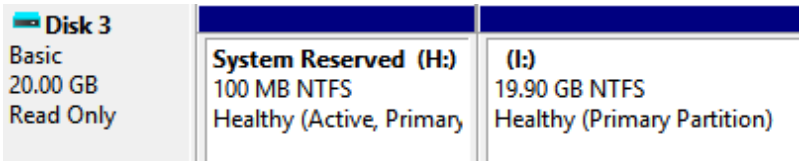

Figure 4.12: Disk Management VHD Mounted Image

Now we can visualize with the tool ShadowCopyView the possible Shadow Copies that were in the disk. Specifically, there is only one snapshot of the system, created on 03/25/2015 at 15:57:27.

We extract the files sync_config.db and snapshot.db (4.2.14). We can see that, in this case, they can be recovered without damage. When being examined by SQLiteStudio we observe that the tables of snapshot.db are are practically empty.

Using SQLiteDeletedRecordsParser tool [19] we get the deleted data from the database:

```
Type        Offset  Length  Data
Unallocated 1034    986 v EK M 0Bz0ye6gXtiZaVl8yVU5mWHlGbWcdo_u_wanna_build_a_snow_man.mp3TUxmho2c4553f99533d85adb104b3a5c38521afilej/ M 0Bz0ye6gXtiZaakx6d
Unallocated 3080    1016    #E0Bz0ye6gXtiZaVl8yVU5mWHlGbWcroot%0Bz0ye6gXtiZaakx6d3R3c0JmM1Uroot
Unallocated 7186    206 rN##Utablecloud_entrycloud_entryCREATE TABLE cloud_entry (doc_id TEXT, filename TEXT, modified INTEGER, created INTEGER, acl_role I
Unallocated 8202    964 *P^KMdo_u_wanna_build_a_snow_man.mp3A2#T(2c4553f99533d85adb104b3a5c38521ahoKY/Mhappy_holiday.jpgA2`0c77d6a2704155dbfdf29817769b7478
Unallocated 9232    324 ) ` tablemappingmappingCREATE TABLE mapping (inode_number INTEGER, doc_id TEXT, UNIQUE (inode_number), FOREIGN KEY (inode_number) R
Unallocated 10248   1016    %)%)
Unallocated 13322   997 `E0Bz0ye6gXtiZaVl8yVU5mWHlGbWc)E0Bz0ye6gXtiZaakx6d3R3c0JmM1U
Unallocated 19464   1016    }8w \\?\C:\Users\informant\Google Drive\happy_holiday.jpgG \\?\C:\Users\informant\Google Drive\do_u_wanna_build_a_snow_man.mp3
```

Figure 4.13: Snapshot.db Recovered Data

4.2.19 Recycle Bin

The first thing to bear in mind when analysing the recycle bin is its organisation. It is divided into folders with alphanumeric names. These folder names correspond to the SID associated with each user (Table 4.6).

Therefore, in the following directory tree:

- $Recycle.Bin
  - S-1-5-21-2425377081-3129163575-2985601102-1000
  - S-1-5-21-2425377081-3129163575-2985601102-1001
  - S-1-5-21-2425377081-3129163575-2985601102-1003

S-1-5-21-2425377081-3129163575-2985601102-1000 corresponds to the SID of the informant user.

There are two file types: those that start with *$I* and those that start with *$R*. Corresponding to the metadata of the deleted file and the actual content of the file respectively.

When examining this folder with Autopsy we see that the files have been deleted and it is not possible to recover them from the image. However, using the Shadow Copy from the previous section, we find that the recycle bin contains the intact files.

Analysing the recovered directories and files we found a folder, $RXWGVWC, that contains three files that, when analysing (as explained in section 4.2.11), its format does not match the one shown:

./my_friends.svg:	Composite Document File V2 Document
./my_smartphone.png:	Microsoft Word 2007+
./new_year_calendar.one:	Microsoft Word 2007+

After renaming the extensions to doxc and using Linux tool unoconv to convert my_friends.svg we get three 'secret project' related documents with the following titles: *Progress #1.doc*, *Progress #2.doc* y *Progress #3.doc*

We apply the same procedure to the folder $RT12FO0, obtaining two presentations related to the design of the secret project.

./winter_storm.amr:	Composite Document File V2 Document
./winter_whether_advisory.zip:	Microsoft PowerPoint 2007+

In the folder $R40295N we find two documents with proposals of the secret project:

./a_gift_from_you.gif:	Microsoft Word 2007+
./landscape.png:	Microsoft Word 2007+

In $R55Z163 folder we find market share and price analysis documents and Excel sheets of the secret project:

./my_favorite_cars.db:	Composite Document File V2 Document
./my_favorite_movies.7z:	Microsoft Excel 2007+
./new_years_day.jpg:	Microsoft Excel 2007+
./super_bowl.avi:	Composite Document File V2 Document

Finally, in folder $R9M7UMY , we find technical review documents of the secret project:

./diary_#1d.txt:	Microsoft Word 2007+
./diary_#1p.txt:	Microsoft PowerPoint 2007+
./diary_#2d.txt:	Microsoft Word 2007+
./diary_#2p.txt:	Composite Document File V2 Document
./diary_#3d.txt:	Composite Document File V2 Document
./diary_#3p.txt:	Composite Document File V2 Document

4.3 Containment, Eradication and Recovery

As has been proven during the analysis no computer system has been compromised by malware or external agents and, therefore, there is no need for a containment or eradication phase. It would only be necessary to requisition the devices that had been used by 'Mr Informant' and restore them to erase any information.

4.4 Post-Incident Response

Parallel to the previous section, the post-incident phase would consist, first and foremost, of a review of security policies or the improvement of monitoring or surveillance systems, if any.

Chapter 5

Case Study: Volatile Memory Analysis

This section is intended to provide an overview of the forensic process carried out on volatile memory and tools used.

5.1 Preparation

Due to the nature of the scenario and project proposed, it is not possible to fully simulate or detail the cybersecurity infrastructure and plans. Therefore, all procedures will be considered to have been properly declared and documented, monitoring tools are fully operational, and sufficient trained personnel are available to deal with any threat and incident.

5.2 Detection and Analysis

As stated in the forensic methodology (Section 3.3), forensic analysis should be performed in four phases. However, due to the approach of the practical case, where it has already been identified, prioritized and acquired, and is posed from a didactic point of view, it has been decided to merge the analysis and reporting phase, structuring it into sections that deal with or group together different concepts and forensic techniques.

In this scenario, certain concepts or techniques previously mentioned in Chapter 4 will not be fully detailed, but referenced, in an attempt to synthesize and avoid redundancies.

5.2.1 Detection: Scenario Context

As mentioned in the previous scenario, any investigation starts from an assumption or context that leads and models the actions to be taken in the following.

This scenario is based on a CCN-CERT forensic challenge [20]:

One of the internal networks of an organization has been the target of an intrusion. An IDS has identified unusual traffic that could reflect lateral movements to other equipment on the same network. It is suspected that the systems of the VLAN may have been compromised.

A memory dump, memory.1221191d.img, of one of the computers in the network has been acquired in order to obtain information about the cause of infection and relevant Indicator of Compromise (IoC).

5.2.2 Evidence Integrity

We are provided with the following MD5 hash: 9452fd27235597dc3bdb09c1b9f2a76a. When calculating the MD5 hash (4.2.2) of the ZIP file with Linux tool md5sum we obtain the next result:

9452fd27235597dc3bdb09c1b9f2a76a memory.1221191d.img.zip

Therefore, since both match, the integrity of the image has been maintained and its information not altered.

5.2.3 Network Packets

Since we know that the intrusion has occurred in one of the organization's internal networks and strange traffic has been detected, we start the investigation recovering possible traces of the system's network activity.

Bulk_extractor tool allows extracting network packets from a memory dump into a pcap file. To do this we execute the following command on our image and examine the file obtained with Wireshark tool:

bulk_extractor -x all -e net -o network/ memory.1221191d.img

First thing we notice is the unusual amount of Server Message Block (SMB) packets.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Frame	100.0	313	100.0	176855	—	0	0
Ethernet	100.0	313	2.5	4382	—	0	0
Internet Protocol Version 4	100.0	313	3.5	6260	—	0	0
User Datagram Protocol	25.9	81	0.4	648	—	0	0
NetBIOS Name Service	16.0	50	1.6	2800	—	50	2800
NetBIOS Datagram Service	2.2	7	0.7	1314	—	0	0
SMB (Server Message Block Protocol)	2.2	7	0.4	740	—	0	0
SMB MailSlot Protocol	2.2	7	0.1	175	—	0	0
Microsoft Windows Browser Protocol	2.2	7	0.1	138	—	7	138
Link-local Multicast Name Resolution	7.7	24	0.3	550	—	24	550
Transmission Control Protocol	74.1	232	91.0	160901	—	154	92283
NetBIOS Session Service	24.9	78	68.5	121078	—	31	40724
SMB (Server Message Block Protocol)	15.0	47	45.3	80166	—	47	80242

Figure 5.1: Protocol Hierarchy Statistics

When analysing the packets, we find that the IP corresponding to our machine is 10.0.15.100. Detecting a large amount of SMB and TCP incoming traffic from IP 10.0.15.20.

```
226 0.000000    10.0.15.20    10.0.15.100    SMB     191 Negotiate Protocol Request
227 0.000000    10.0.15.20    10.0.15.100    SMB     194 Session Setup AndX Request, User: anonymous
228 0.000000    10.0.15.20    10.0.15.100    SMB     146 Tree Connect AndX Request, Path: \\10.0.15.100\IPC$
229 0.000000    10.0.15.20    10.0.15.100    SMB     136 Trans2 Request, SESSION_SETUP
230 0.000000    10.0.15.20    10.0.15.100    SMB    1138 NT Trans Request, <unknown>
231 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=1536 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
232 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=2996 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
233 0.000000    10.0.15.20    10.0.15.100    SMB    1287 Trans2 Secondary Request, FID: 0x0000
234 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=5689 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
235 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=7149 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
236 0.000000    10.0.15.20    10.0.15.100    SMB    1287 Trans2 Secondary Request, FID: 0x0000
237 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=9842 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
238 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=11302 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
239 0.000000    10.0.15.20    10.0.15.100    SMB    1287 Trans2 Secondary Request, FID: 0x0000
240 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=13995 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
241 0.000000    10.0.15.20    10.0.15.100    TCP    1514 1346 → 445 [ACK] Seq=15455 Ack=453 Win=63788 Len=1460 [TCP segment of a reassembled PDU]
242 0.000000    10.0.15.20    10.0.15.100    SMB    1287 Trans2 Secondary Request, FID: 0x0000
```

Figure 5.2: TCP Stream

5.2.4 EternalBlue

Microsoft Windows Server Message Block (SMB) protocol allows, among others, remote access to files and printers. In 2017, a vulnerability [21] was disclosed for this protocol: EternalBlue.

First, a negotiation is established between attacker and victim in order to establish the session (with hard-coded victim's IP), and then, through specifically crafted packages, exploit this vulnerability. [22].

```
226 0.000000    10.0.15.20    10.0.15.100    SMB     191 Negotiate Protocol Request
227 0.000000    10.0.15.20    10.0.15.100    SMB     194 Session Setup AndX Request, User: anonymous
228 0.000000    10.0.15.20    10.0.15.100    SMB     146 Tree Connect AndX Request, Path: \\10.0.15.100\IPC$
```

Figure 5.3: SMB Session Setup

After the session is established, a series of no-operations (blank) is sent to alter machine's state with the following content (packet 230 in Figure 5.2):

```
00 00 4b 00 00 00 d0 03  00 00 68 00 00 00 01 00   ··K····· ··h·····
00 00 00 ec 03 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Figure 5.4: SMB Transmission Content Fragment

This serves as a starting point for subsequently, in a series of SMB transmissions, introducing the encrypted payload (packets 233, 236, 239, 242 in Figure 5.2):

```
34 4e 34 6d 53 6b 6e 2f  73 52 6c 79 2b 6a 64 62   4N4mSkn/ sRly+jdb
4f 69 56 55 42 6f 78 37  37 4b 34 50 2b 51 45 2b   OiVUBox7 7K4P+QE+
6b 4d 35 69 54 73 70 58  58 41 79 65 43 6c 67 36   kM5iTspX XAyeClg6
2f 41 5a 4e 36 65 56 69  54 61 75 55 45 33 56 45   /AZN6eVi TauUE3VE
42 30 33 6c 37 39 72 36  4c 74 30 52 4b 44 35 2b   B03l79r6 Lt0RKD5+
35 37 33 47 63 35 67 33  71 6e 42 6e 74 35 5a 33   573Gc5g3 qnBnt5Z3
6b 54 79 34 77 53 4b 74  38 77 45 78 38 63 54 2b   kTy4wSKt 8wEx8cT+
56 32 78 71 42 36 6e 7a  2b 62 7a 6f 58 6f 54 46   V2xqB6nz +bzoXoTF
79 4e 41 53 65 46 37 39  30 42 34 4a 2f 4c 6b 4a   yNASeF79 0B4J/LkJ
37 47 78 69 33 4f 76 50  72 46 4b 36 4b 66 68 51   7Gxi3OvP rFK6KfhQ
45 52 35 62 61 6a 72 6b  4f 32 75 32 31 66 66 6d   ER5bajrk O2u21ffm
```

Figure 5.5: SMB Transmission Encrypted Payload Fragment

5.2.5 Profiling

During this scenario, the Volatility tool will be used as a baseline. Implemented in Python it allows the extraction of volatile memory artifacts.

The first step when analysing a memory dump is to identify the corresponding operating system. Each OS assigns different memory addresses so it is necessary to have prior knowledge of the profile and version to map and accurately locate the desired information.

Therefore, using Volatility command:

vol.py -f memory.1221191d.img imageinfo

We obtain the possible profiles, corresponding to the operating system, that match our image.

```
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/linux/volatility/memory.1221191d.img)
                      PAE type : No PAE
                           DTB : 0x185000L
                          KDBG : 0x82923ea8L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82924d00L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2017-08-07 20:23:00 UTC+0000
     Image local date and time : 2017-08-07 22:23:00 +0200
```

Figure 5.6: Volatility Image Profiling

Therefore, and based on these results, we can determine that the memory dump belongs to a Windows 7 system and dates from 07-08-2017 at 20:23:00 UTC.

It also suggests a series of profiles, ordered from highest to lowest probability, to be used: Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86.

5.2.6 Process Listing

We check if the previously suggested profiles are valid, for it is enough to run any Volatility plugin stating the profile to be tested. If the information is extracted correctly the profile is suitable.

We list the processes running in memory with the following command:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 pstree

Name	Pid	PPid	Thds	Hnds	Time
0x85a41030:explorer.exe	1368	1344	24	891	2017-08-07 20:12:41 UTC+0000
. 0x858b2030:firefox.exe	2272	1368	47	606	2017-08-07 20:13:04 UTC+0000
.. 0x8589ebc8:firefox.exe	2512	2272	19	294	2017-08-07 20:13:06 UTC+0000
. 0x856734f0:vlc.exe	2076	1368	7	327	2017-08-07 20:12:58 UTC+0000
. 0x8565f818:swriter.exe	1584	1368	1	16	2017-08-07 20:12:56 UTC+0000
.. 0x84fe2d28:soffice.exe	1824	1584	1	61	2017-08-07 20:12:56 UTC+0000
... 0x8565daa8:soffice.bin	1240	1824	16	355	2017-08-07 20:12:57 UTC+0000
. 0x841bd030:cmd.exe	2232	1368	1	22	2017-08-07 20:21:34 UTC+0000
. 0x856f9620:FoxitReader.ex	2380	1368	25	483	2017-08-07 20:13:05 UTC+0000
.. 0x856f5a40:FoxitReaderUpd	2560	2380	0	------	2017-08-07 20:13:07 UTC+0000
. 0x83fca320:calc.exe	352	1368	3	75	2017-08-07 20:19:30 UTC+0000
. 0x8571fd28:cmd.exe	3528	1368	1	24	2017-08-07 20:13:28 UTC+0000
.. 0x859d3180:Memoryze.exe	3588	3528	2	96	2017-08-07 20:22:59 UTC+0000
. 0x85abe030:VBoxTray.exe	1636	1368	12	152	2017-08-07 20:12:42 UTC+0000
. 0x859005f0:msiexec.exe	1512	1368	4	148	2017-08-07 20:18:38 UTC+0000
0x8559b030:csrss.exe	388	372	9	425	2017-08-07 20:12:37 UTC+0000
. 0x858b7840:conhost.exe	336	388	2	54	2017-08-07 20:22:59 UTC+0000
.. 0x84f8e968:wininit.exe	380	336	5	84	2017-08-07 20:12:37 UTC+0000
... 0x8575f030:services.exe	472	380	9	206	2017-08-07 20:12:38 UTC+0000
... 0x84fc3c30:lsass.exe	480	380	0	------	2017-08-07 20:12:39 UTC+0000
.... 0x841b41f0:rundll32.exe	300	480	1	51	2017-08-07 20:22:46 UTC+0000
... 0x84fc3208:lsm.exe	488	380	10	150	2017-08-07 20:12:39 UTC+0000
. 0x8407ad28:conhost.exe	3920	388	2	55	2017-08-07 20:21:34 UTC+0000
0x8556ed28:winlogon.exe	428	372	4	117	2017-08-07 20:12:38 UTC+0000
. 0x841a7030:wlrmdr.exe	3008	428	0	------	2017-08-07 20:22:47 UTC+0000
0x83f2fba0:System	4	0	86	527	2017-08-07 20:12:33 UTC+0000
. 0x84e44d28:smss.exe	268	4	2	29	2017-08-07 20:12:33 UTC+0000

Figure 5.7: Process List (Shortened)

As the extraction of the processes has been done successfully and, therefore, the profile used, Win7SP1x86_23418, is accurate, it will be used in the remainder of the analysis.

We observe a process *rundll32* with *lsass* as parent. Rundll32 allows the loading of dll libraries into memory for use by other programs, this process is child of the lsass process or the Local Security Authority Subsystem Service of Windows that controls access to the system. At the moment we cannot deepen more so we will return to this analysis later on.

Listing processes using *pstree* (or *pslist*) allows you to list running processes. However, it only lists the visible ones.

There may be a situation where there are hidden or unlinked processes. Using *psscan* it is possible to obtain these processes as well as those already finished and, therefore, it is advisable to look for discrepancies between both lists of processes.

We run *psxview* to directly compare processes:

Offset(P)	Name	PID	pslist	psscan	thrdproc	pspcid	csrss	session	deskthrd	ExitTime
0x1e342500	SearchIndexer.	1596	True	True	True	True	True	True	True	
0x1e1a1240	svchost.exe	1056	True	True	True	True	True	True	False	
0x1efe2d28	soffice.exe	1824	True	True	True	True	True	True	True	
0x1e0b2030	firefox.exe	2272	True	True	True	True	True	True	True	
0x1e103790	svchost.exe	712	True	True	True	True	True	True	False	
0x1e65f818	swriter.exe	1584	True	True	True	True	True	True	True	
0x1e1d3180	Memoryze.exe	3588	True	True	True	True	True	True	True	
0x1e2be030	VBoxTray.exe	1636	True	True	True	True	True	True	True	
0x1f9b41f0	rundll32.exe	300	True	True	True	True	True	True	True	
0x1f89e710	msiexec.exe	1688	True	True	True	True	True	True	True	
0x1e7387f0	WmiPrvSE.exe	2988	True	True	True	True	True	True	True	
0x1e1a2370	audiodg.exe	1020	True	True	True	True	True	True	True	
0x1e115178	VBoxService.ex	660	True	True	True	True	True	True	False	
0x1e232428	dwm.exe	1356	True	True	True	True	True	True	True	
0x1efc3c30	lsass.exe	480	True	True	False	True	False	True	False	2017-08-07 20:22:47 UTC+0000
0x1e6f5a40	FoxitReaderUpd	2560	True	True	False	True	False	True	False	2017-08-07 20:13:08 UTC+0000
0x0072fba0	System	4	True	True	True	True	False	False	False	
0x1f88f030	msiexec.exe	588	True	True	False	True	False	True	False	2017-08-07 20:16:43 UTC+0000
0x1e59b030	csrss.exe	388	True	True	True	True	False	True	True	
0x1ef431d8	csrss.exe	344	True	True	True	True	False	True	False	
0x1f9a7030	wlrmdr.exe	3008	True	True	False	True	False	True	False	2017-08-07 20:22:53 UTC+0000
0x1ee44d28	smss.exe	268	True	True	True	True	False	False	False	
0x0bb6bba0	System	4	False	True	False	False	False	False	False	
0x06513ba0	System	4	False	True	False	False	False	False	False	
0x1fcdf428	dwm.exe	1356	False	True	False	False	False	False	False	
0x1f8e2d28	svchost.exe	2024	False	True	False	False	False	False	False	2017-08-07 20:22:40 UTC+0000
0x1f90da40	dllhost.exe	3660	False	True	False	False	False	False	False	2017-08-07 20:17:13 UTC+0000

Figure 5.8: Process List Comparison (Shortened)

The last 5 processes are listed in *psscan* and not in *pslist*. This is not uncommon in the case of terminated processes, however, in active processes it may indicate the presence of malicious code. As in the previous case, we will resume their analysis further on.

5.2.7 Command History

We can see the different commands entered by the user in Windows Command Prompt:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 cmdscan

```
CommandProcess: conhost.exe Pid: 3536
CommandHistory: 0x1a1eb8 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 27 LastAdded: 26 LastDisplayed: 26
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x1964d0: cd ..
Cmd #1 @ 0x1820e8: cd "Program Files"
Cmd #2 @ 0x1964e8: cd ..
Cmd #3 @ 0x198168: cd ProgramData
Cmd #4 @ 0x19e908: cd MANDIANT
Cmd #6 @ 0x19f308: dir
Cmd #7 @ 0x19e928: cd Memoryze
Cmd #16 @ 0x19f368: dir
Cmd #18 @ 0x196548: cd ..
Cmd #19 @ 0x1a7590: cd "Archivos de programa"
Cmd #20 @ 0x19f378: dir
Cmd #21 @ 0x196560: cd ..
Cmd #22 @ 0x1a7078: cd "Program Files"
Cmd #23 @ 0x19e8e8: cd MANDIANT
Cmd #24 @ 0x19e988: cd Memoryze
Cmd #25 @ 0x198190: ping 10.0.15.20
Cmd #26 @ 0x1981e0: MemoryDD.bat

CommandProcess: conhost.exe Pid: 3920
CommandHistory: 0x280e88 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x27e858: netstat -an

CommandProcess: conhost.exe Pid: 336
CommandHistory: 0x61e68 Application: Memoryze.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
```

Figure 5.9: Command Prompt User's Input (Shortened)

However, once analysed in depth (volatility plugin *consoles*), we see that these are commands related to the acquisition of the RAM and not to the infection.

5.2.8 Network Scan

After analysing the network traffic of the system (5.2.3), and noticing suspicious network traffic, it is necessary to check the TCP and UDP connections of our equipment. To do this we run the plugin *netscan* of volatility that allows us to list these connections:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 netscan

It is also possible to analyse the console output (plugin *consoles*) of the command *netstat -an* entered by the user in the previous section or contrast both outputs in case of discrepancies.

Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner
0x1e10ff50	UDPv4	10.0.15.100:1900	*:*		3956	svchost.exe
0x1e126830	UDPv4	10.0.15.100:52904	*:*		3956	svchost.exe
0x1e127800	UDPv6	::1:52903	*:*		3956	svchost.exe
0x1e135508	UDPv4	127.0.0.1:1900	*:*		3956	svchost.exe
0x1e1c46d0	UDPv4	127.0.0.1:52905	*:*		3956	svchost.exe
0x1e8d5588	UDPv4	0.0.0.0:0	*:*		660	VBoxService.ex
0x1e0c37a8	TCPv4	0.0.0.0:49153	0.0.0.0:0	LISTENING	760	svchost.exe
0x1e112bb0	TCPv6	:::135	:::0	LISTENING	712	svchost.exe
0x1e11e970	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	380	wininit.exe
0x1e11f378	TCPv4	0.0.0.0:49152	0.0.0.0:0	LISTENING	380	wininit.exe
0x1e30a7b8	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	232	svchost.exe
0x1e30a7b8	TCPv6	:::49156	:::0	LISTENING	232	svchost.exe
0x1e380618	TCPv4	0.0.0.0:445	0.0.0.0:0	LISTENING	4	System
0x1e380618	TCPv6	:::445	:::0	LISTENING	4	System
0x1e3870b0	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	472	services.exe
0x1e3870b0	TCPv6	:::49155	:::0	LISTENING	472	services.exe
0x1e387758	TCPv4	0.0.0.0:49155	0.0.0.0:0	LISTENING	472	services.exe
0x1e5ea980	TCPv4	0.0.0.0:8080	0.0.0.0:0	LISTENING	300	rundll32.exe
0x1e60d080	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	232	svchost.exe
0x1e3296a8	TCPv4	127.0.0.1:49162	127.0.0.1:49161	ESTABLISHED	-1	
0x1e32d508	TCPv4	127.0.0.1:49161	127.0.0.1:49162	ESTABLISHED	-1	
0x1e721ca0	TCPv4	127.0.0.1:49159	127.0.0.1:49160	ESTABLISHED	-1	
0x1e724008	TCPv4	127.0.0.1:49160	127.0.0.1:49159	ESTABLISHED	-1	
0x1edd33f8	UDPv4	0.0.0.0:500	*:*		960	svchost.exe
0x1edd3e98	UDPv4	0.0.0.0:4500	*:*		960	svchost.exe
0x1edd3e98	UDPv6	:::4500	*:*		960	svchost.exe
0x1ee5f580	UDPv4	10.0.15.100:138	*:*		4	System
0x1ee91330	UDPv4	10.0.15.100:137	*:*		4	System
0x1ee7b930	TCPv4	10.0.15.100:139	0.0.0.0:0	LISTENING	4	System
0x1f8ff908	UDPv4	0.0.0.0:0	*:*		2272	firefox.exe
0x1f8ff908	UDPv6	:::0	*:*		2272	firefox.exe

Figure 5.10: Netscan (Shortened)

Note that the process *rundll32* with PID 300, previously mentioned, is listed again listening on port 8080.

5.2.9 Threat Analysis

Malfind command finds hidden or injected code and DLLs in user mode memory. We run the command:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 malfind

In order to check if any injection is detected in any of the processes previously catalogued as suspicious.

```
Process: rundll32.exe Pid: 300 Address: 0x70000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 1, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x00070000  fc e8 82 00 00 00 60 89 e5 31 c0 64 8b 50 30 8b   ......‘..1.d.P0.
0x00070010  52 0c 8b 52 14 8b 72 28 0f b7 4a 26 31 ff ac 3c   R..R..r(..J&1..<
0x00070020  61 7c 02 2c 20 c1 cf 0d 01 c7 e2 f2 52 57 8b 52   a|.,........RW.R
0x00070030  10 8b 4a 3c 8b 4c 11 78 e3 48 01 d1 51 8b 59 20   ..J<.L.x.H..Q.Y.

0x00070000 fc               CLD
0x00070001 e882000000       CALL 0x70088
0x00070006 60               PUSHA
0x00070007 89e5             MOV EBP, ESP
0x00070009 31c0             XOR EAX, EAX
0x0007000b 648b5030         MOV EDX, [FS:EAX+0x30]
0x0007000f 8b520c           MOV EDX, [EDX+0xc]
0x00070012 8b5214           MOV EDX, [EDX+0x14]
0x00070015 8b7228           MOV ESI, [EDX+0x28]
0x00070018 0fb74a26         MOVZX ECX, WORD [EDX+0x26]
0x0007001c 31ff             XOR EDI, EDI
0x0007001e ac               LODSB
0x0007001f 3c61             CMP AL, 0x61
0x00070021 7c02             JL 0x70025
0x00070023 2c20             SUB AL, 0x20
0x00070025 c1cf0d           ROR EDI, 0xd
0x00070028 01c7             ADD EDI, EAX
0x0007002a e2f2             LOOP 0x7001e
0x0007002c 52               PUSH EDX
0x0007002d 57               PUSH EDI
0x0007002e 8b5210           MOV EDX, [EDX+0x10]
0x00070031 8b4a3c           MOV ECX, [EDX+0x3c]
0x00070034 8b4c1178         MOV ECX, [ECX+EDX+0x78]
0x00070038 e348             JECXZ 0x70082
0x0007003a 01d1             ADD ECX, EDX
0x0007003c 51               PUSH ECX
0x0007003d 8b5920           MOV EBX, [ECX+0x20]
```

Figure 5.11: Rundll32 Injected Code

There is only one coincidence: rundll32 process with PID 300. Therefore, based on the previous hints, we proceed to examine this process.

As a first approach we perform a memory dump of the injected section detected by malfind:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 malfind -p 300 -D ./malfind/

It is also possible, knowing the memory offset where the injection begins (0x00070000), to dump the memory section:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 vaddump -p 300 -b 0x00070000 -D ./memdump/

Once the dump is finished it is automatically detected by Windows Defender and classified as:

Trojan:Win32/Meterpreter.gen!R

The process acts as a Meterpreter payload, a remote command interpreter. Meterpreter is widely used in post-exploitation phase to gain control of the equipment. It is stored entirely in volatile memory, leaving no traces on disk. Therefore it is to suppose that the listening in the port 8080 of this process has as purpose the connection with the attacking system.

Once the payload is known the next step is to ascertain the input vector and the vulnerability.

We resume the analysis of hidden processes and dump their memory sections. We also dump rundll32 process entirely for further information:

vol.py -f memory.1221191d.img –profile=Win7SP1x86_23418 memdump -p 300 -D ./memdump

We perform the extraction on the hidden processes *System* and *dwm*, with PID 4 and 1356 respectively (Figure 5.8).

Under certain circumstances, within the DFIR scope, it is not advisable (rather discouraged) to use public malware analysis services due to their own nature and public exposure of the submitted samples.

Nonetheless, given the underlying characteristics of this project and the limitations of both length and resources, where there is not a properly prepared laboratory with the necessary malware analysis tools, Virus Total platform will be used to perform analysis and contrast results of the various files classified as potentially malicious during the course of this analysis.

No malicious behaviour is found in the hidden process *dmw*. However, when analysing the *System* process and *rundll32*, both are catalogued as *Sf:WNCryLdr-A [Trj]* (by Avast and AVG): Wannacry.

The ransomware Wannacry uses a vulnerability in the SMB protocol of Microsoft Windows to gain control over the system, exploited by EternalBlue and DoublePulsar backdoor. The observed traffic matching EternalBlue behaviour (Sections 5.2.3 and 5.2.4) confirms this evidence.

Therefore, we can conclude that the network has been violated using CVE-2017-0143 vulnerability [23] and that, subsequently, the computers are controlled remotely by command interpreter.

5.2.10 Signature Scan

The automatic antivirus scan has allowed us to determine the type of malware, however, and not to depend exclusively on the updates of these antivirus, we will perform an analysis looking for specific signatures or IoC.

Based on the knowledge obtained in previous sections, where:

- The network behavior corresponds to an EternalBlue exploit.
- Windows Defender classifies as Meterpreter the injected section of code into the *rundll32* process.

And knowing that Meterpreter is the most used command interpreter by Metasploit framework (used in penetration testing) it is logical to suppose that a specific module of EternalBlue of Metasploit [24] will have been used to intrude our systems.

If we examine the most popular EternalBlue module we find a function, *make_kernel_shellcode*, that generates the hex code to be injected.

```
def make_kernel_shellcode(proc_name)
        # Length: 1019 bytes

        # "\xcc"+
        "\x31\xC9\x41\xE2\x01\xC3\xB9\x82\x00\x00\xC0\x0F\x32\x48\xBB\xF8" +
        "\x0F\xD0\xFF\xFF\xFF\xFF\xFF\x89\x53\x04\x89\x03\x48\x8D\x05\x0A" +
        "\x00\x00\x00\x48\x89\xC2\x48\xC1\xEA\x20\x0F\x30\xC3\x0F\x01\xF8" +
        "\x65\x48\x89\x24\x25\x10\x00\x00\x00\x65\x48\x8B\x24\x25\xA8\x01" +
        "\x00\x00\x50\x53\x51\x52\x56\x57\x55\x41\x50\x41\x51\x41\x52\x41" +
        "\x53\x41\x54\x41\x55\x41\x56\x41\x57\x6A\x2B\x65\xFF\x34\x25\x10" +
        "\x00\x00\x00\x41\x53\x6A\x33\x51\x4C\x89\xD1\x48\x83\xEC\x08\x55" +
```

Listing 5.1: Github Exploit Code Segment [24]

This code will be injected, in case this module has been used, in the memory of our system and, therefore, we will be able to carry out a scan in search of matches.

To do so, we define the following Yara rule resembling the previous code:

```
rule Metasploit_ms17_010_EternalBlue {
        meta:
                author = "Javier Martinez Llamas"
        strings:
                $hex1 = {0F D0 FF FF FF FF FF 89 53 04 89 03 48 8D 05 0A}
                $hex2 = {00 00 00 48 89 C2 48 C1 EA 20 0F 30 C3 0F 01 F8}
                $hex3 = {65 48 89 24 25 10 00 00 00 65 48 8B 24 25 A8 01}
                $hex4 = {00 00 50 53 51 52 56 57 55 41 50 41 51 41 52 41}
                $hex5 = {53 41 54 41 55 41 56 41 57 6A 2B 65 FF 34 25 10}
                $hex6 = {00 00 00 41 53 6A 33 51 4C 89 D1 48 83 EC 08 55}
        condition:
                all of them
}
```

Listing 5.2: EternalBlue Exploit Yara Rule

Yara is a tool that allows the classification of malware samples based on string or binary patterns. In this case we will limit ourselves to develop a rule that searches the hexadecimal code of the public repository.

Execute Yara using the following command, on processes dumped *rundll32* (PID 300) and *System* (PID 4):

yara Metasploit_ms17_010_EternalBlue.yara memdump/300.dmp

yara Metasploit_ms17_010_EternalBlue.yara memdump/4.dmp

Matching both cases with the defined rule. If we list the matched strings we visualize the following:

```
Metasploit_ms17_010_EternalBlue memdump/4.dmp
0xcd7495:$hex1: 0F D0 FF FF FF FF FF 89 53 04 89 03 48 8D 05 0A
0xd0a495:$hex1: 0F D0 FF FF FF FF FF 89 53 04 89 03 48 8D 05 0A
...
0x1b3e305:$hex2: 00 00 00 48 89 C2 48 C1 EA 20 0F 30 C3 0F 01 F8
0x616b2c5:$hex2: 00 00 00 48 89 C2 48 C1 EA 20 0F 30 C3 0F 01 F8
0xcd74b5:$hex3: 65 48 89 24 25 10 00 00 00 65 48 8B 24 25 A8 01
0xd0a4b5:$hex3: 65 48 89 24 25 10 00 00 00 65 48 8B 24 25 A8 01
...
0x1b3e325:$hex4: 00 00 50 53 51 52 56 57 55 41 50 41 51 41 52 41
0x616b2e5:$hex4: 00 00 50 53 51 52 56 57 55 41 50 41 51 41 52 41
0xcd74d5:$hex5: 53 41 54 41 55 41 56 41 57 6A 2B 65 FF 34 25 10
0xd0a4d5:$hex5: 53 41 54 41 55 41 56 41 57 6A 2B 65 FF 34 25 10
...
0x1b3e345:$hex6: 00 00 00 41 53 6A 33 51 4C 89 D1 48 83 EC 08 55
0x616b305:$hex6: 00 00 00 41 53 6A 33 51 4C 89 D1 48 83 EC 08 55
```

Figure 5.12: Yara Matching Strings on System Process (Shortened)

This method of analysis, with a more manual approach, makes it possible to refine and fine-tune the search, as opposed to automatic analyses with antivirus. However, since it is applied to specific cases and is based on previous evidence or suspicions, it is possible that, if the rules are not correctly defined, it could lead to misinterpretation and a large number of false positives.

5.3 Containment, Eradication and Recovery

Once the analysis has been carried out and the scope of the threat and its entry vector has been determined and verified, it is necessary to take the necessary measures to prevent its diffusion and, once it has been contained, to eliminate it completely from our systems. For this, the logical sequence of action would be:

- Isolation of the affected and invaded VLAN to limit diffusion within the organization's network and ensure that it has not expanded to other network areas, in which case to proceed in the same manner with these.
- Thoroughly check the affected systems and isolate them from the Internet and then restore and update them or apply a patch to correct Microsoft's SMB vulnerability.
- Deepen the analysis of the threat in order to obtain new IoC to help determine the origin of the infection and be able to remedy or block it in case they are known external or internal agents.

These measures, despite being a general guide to action, will be affected by any security plan of the company that requires it to operate differently or requires additional actions.

5.4 Post-Incident Response

Unlike the previous case study, the incident was caused by an exploit and malware and, therefore, feedback from the experience gained is essential for the proper future functioning of the entire defence system.

All the information relating to this exploit must be duly documented so that, in the event of a recurrence, the lower levels of the SOC or any specialist responsible for or involved in the security of the infrastructure can react optimally.

To this end, training must be carried out to pass on this knowledge and expertise to all personnel along documentation. In this case we will use MITRE for this purpose.

5.4.1 ATT&CK MITRE

There are numerous threat intelligence tools or databases to improve the response of organizations and teams, both on the offensive and defensive sides. The framework ATT&CK MITRE is a knowledge base which objective is the description and cataloguing of adverse behaviours and techniques.

This knowledge is divided into three matrices: preparation, enterprise and mobile. Corresponding to techniques and tactics performed by attackers before an exploit, techniques applied to Windows, Linux and MacOS and techniques applied to mobile devices respectively.

Based on the forensic analysis carried out during the incident, where the threat has been catalogued as WannaCry, we will use the database of the various exploits to develop a behavioural profile [25] that can serve us in the future to improve the response and prevention plan.

The matrices are divided into tactics (columns) and techniques (cells), so to achieve a tactic an attacker would use different techniques. The resulting matrix of ransomware WannaCry would be:

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
33 items	59 items	28 items	67 items	19 items
Regsvr32	File System Permissions Weakness	New Service	Execution Guardrails	Securityd Memory
Rundll32	Hidden Files and Directories	Path Interception	Exploitation for Defense Evasion	Two-Factor Authentication Interception
Scheduled Task	Hooking	Plist Modification	Extra Window Memory Injection	
Scripting	Hypervisor	Port Monitors	File Deletion	
Service Execution	Image File Execution Options Injection	Process Injection	File Permissions Modification	
Signed Binary Proxy Execution	Kernel Modules and Extensions	Scheduled Task	File System Logical Offsets	
Signed Script Proxy Execution	Launch Agent	Service Registry Permissions Weakness	Gatekeeper Bypass	
Source	Launch Daemon	Setuid and Setgid	Group Policy Modification	
Space after Filename	Launchctl	SID-History Injection	Hidden Files and Directories	
Third-party Software	LC_LOAD_DYLIB Addition	Startup Items	Hidden Users	
Trap	Local Job Scheduling	Sudo	Hidden Window	
Trusted Developer Utilities	Login Item	Sudo Caching	HISTCONTROL	
User Execution	Logon Scripts	Valid Accounts	Image File Execution Options Injection	
Windows Management Instrumentation	LSASS Driver	Web Shell	Indicator Blocking	
Windows Remote Management	Modify Existing Service		Indicator Removal from Tools	
XSL Script Processing	Netsh Helper DLL		Indicator Removal on Host	
	New Service		Indirect Command Execution	
			Install Root Certificate	
			InstallUtil	

Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
22 items	17 items	13 items	22 items	9 items	14 items
Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
File and Directory Discovery	Logon Scripts	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Network Service Scanning	Pass the Hash	Data from Network Shared Drive	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Network Share Discovery	Pass the Ticket	Data from Removable Media	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Network Sniffing	Remote Desktop Protocol	Data Staged	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Peripheral Device Discovery	Remote Services	Input Capture	Fallback Channels		Resource Hijacking
Permission Groups Discovery	Replication Through Removable Media	Man in the Browser	Multi-hop Proxy		Runtime Data Manipulation
Process Discovery	Shared Webroot	Screen Capture	Multi-Stage Channels		Service Stop
Query Registry	SSH Hijacking	Video Capture	Multiband Communication		Stored Data Manipulation
Remote System Discovery	Taint Shared Content		Multilayer Encryption		Transmitted Data Manipulation
Security Software Discovery	Third-party Software		Port Knocking		
System Information Discovery	Windows Admin Shares		Remote Access Tools		
System Network Configuration Discovery			Remote File Copy		
System Network Connections Discovery					

Figure 5.13: WannaCry ATT&CK MITRE Matrix [25]

With the following tactics:

- Execution - Execution of adversary-controlled code on a local or remote system.
- Persistence - Allows the attacker to maintain control of the system in situations where the connection would be interrupted, such as restarts or shutdowns.
- Privilege Escalation - Obtains a higher level of permissions on a system or network.
- Defence Evasion - Evades detection or avoid other defence systems.
- Discovery - Gains knowledge about the system and internal network.
- Lateral Movement - Accesses and controls other systems in a network or cloud.
- Command And Control - Attacker communicates remotely with the target system.
- Impact - Reduces the availability or integrity of a system, service, or network.

As can be seen, its behaviour matches the hints and evidences discovered during the forensic analysis. All the tactics and techniques mentioned in the matrix are detailed in depth in MITRE's knowledge base, being especially useful for post-incident response.

Chapter 6

Conclusion

6.1 Challenges

6.1.1 Cross-cutting Concerns

As mentioned during this document, digital forensics has stalled in recent years. This has been motivated, inter alia, by the lack of evolution in terms of techniques and tools that allow the field to remain in the state of the art in view of the increasing heterogeneity and complexity of the systems.

Diversity is one of the biggest challenges facing digital forensics today. As new technologies are developed and computer paradigms are altered, new standardized forensic methods of analysis and specialization are required. It is, therefore, advisable to constantly update the forensic tools in order to counteract and minimize this difference in pace between analysed and analysing systems development life cycle.

The analysis of timelines is of vital importance during an investigation. It allows the evidences to be sequenced and relationships and conclusions to be drawn. To do so, it is necessary to obtain the different timelines from all available sources.

As is to be expected, in an increasingly global and heterogeneous scenario, these timelines and time uses can vary enormously, and it may not be clear which corresponds in each case. This is due to different reasons, among which are: diverse interpretations of time uses in evidence, interpretation in timestamp formats or synchronization of clocks.

All these cases, being of a general nature, can affect any of the areas that comprise digital forensics.

6.1.2 Computer Forensics

The biggest problem facing computer forensics is the exponential growth in the size of different storage media. This increasingly large size is a problem in evidence acquisition, specially in raw copying (bit-to-bit).

Having repercussions on the requirements of forensic analysis. The successive copies and extraction of evidences suppose, as the investigation is prolonged in time, an enormous consumption of size in disks. The same applies to the computational capacity needed to process all information.

6.1.3 Live Forensics

The main problem of live forensics comes from its own nature: inconsistency. Information, being volatile, is unstable and in a continuous state of change.

This presents, not so much a problem in the procedure itself, but in legal aspects as the integrity of the data cannot be demonstrated. The state of the system, once the memory dump has been carried out, is different from the state prior to acquisition.

6.2 Ethical Responsibility

In spite of not dealing during the course of this project with the legal aspects concerning digital forensics (due to its complexity and depth) it is necessary to mention, for its relevance to be stated, the ethics that underlies the forensic procedure.

There are different approaches to this point of view: from a pragmatic and legal point of view or from an ethical point of view.

Any forensic procedure must adhere to a contract or legal guarantees that allow and justify, by law, conducting the pertinent investigation. Strictly abiding by what is contemplated in the contract, if any. However, there are several aspects that are not reflected or their compliance is difficult to control. This is where the ethics of the responsible for the analysis comes in.

The first area is privacy and confidentiality, where the differentiation between sensitive data and evidence could be questioned. Therefore, information that could be of a sensitive nature and that does not constitute part of the evidence must be treated with total delicacy. Avoiding at all times any leakage of information.

Another relevant aspect is to provide the affected or interested party with full guarantees during the analysis, so that they can know or defend innocence if necessary. This is especially present in business and internal investigations (such as the one discussed in the case study).

Finally, the absence of ethics itself will be mentioned: manipulation or falsification of analysis. In order to obtain certain benefits, there is the possibility that investigations may be maliciously altered. This point not only appeals directly to the ethics of the person involved but also incurs a crime.

Parallel to incident response, where a response, evaluation or recommendation can be adulterated, modified or aggravated, having an enormous potential impact on the client.

6.3 Future Research

6.3.1 Linux Forensics

As seen during the course of this project, computer forensics is, due to its technical nature, largely influenced by the operating system and the version on which it operates. It is for this reason that, despite the fact that in environments such as Windows this branch of cybersecurity may be more established, regardless of its possible drawbacks and stagnation, in Linux environments and distributions the paradigm is different.

Computer forensics in Linux starts from a basic problem. Where the large number of distributions and kernels available simultaneously means that, despite the similarities between them all, the slightest differences (both in structure and behaviour) prevent the optimum development of forensic technologies and techniques for these environments. The explicit definition of these differences and the compilation of particular tools are necessary. However, there are common tools that automate extraction but, as expected, certain components or modules do not operate in specific situations.

Therefore, one of the fields that can experience the greatest growth is computer forensics in Linux, combining and unifying tools, techniques and processes for any Linux system.

6.3.2 Automation

Another area with development potential and great utility is the automation of the forensic process. The extraction of all forensic artifacts, correlation and analysis of evidence is a long and meticulous process, depending heavily on the results and conclusions of all previous steps and their proper execution.

It is, therefore, not surprising that the automation of this process is of great potential. Not only facilitating the analysis in a universal way, independently of the operating system, but an automation of the parsing of logs, extraction of key artifacts or correlations.

While it is true that there are tools that act as automatic extractors, generating logs of the results, their performance is only a support. With the current calculation capabilities and read/write speeds it would be possible to improve these processes with the help of current techniques such as Artificial Intelligence.

Bibliography

[1] NIST. *Data Leakage Case*. URL: https://www.cfreds.nist.gov/data_leakage_case/data-leakage-case.html (visited on 2019).

[2] Ci Song et al. *The washing away of wrongs: forensic medicine in thirteenth-century China*. 1. University of Michigan Press, 1981.

[3] Microsoft. *How NTFS Works*. URL: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781134(v=ws.10) (visited on 2019).

[4] Microsoft. *Master File Table*. URL: https://docs.microsoft.com/es-es/windows/desktop/DevNotes/master-file-table (visited on 2019).

[5] ForensicWiki. *New Technology File System (NTFS)*. URL: https://forensicswiki.org/wiki/New_Technology_File_System_(NTFS) (visited on 2019).

[6] Yudi Prayudi and Azhari Sn. "Digital chain of custody: State of the art". In: *International Journal of Computer Applications* 114.5 (2015).

[7] Eoghan Casey and Gerasimos J Stellatos. "The impact of full disk encryption on digital forensics". In: *ACM SIGOPS Operating Systems Review* 42.3 (2008), pp. 93–98.

[8] The Sleuth Kit. *Autopsy*. URL: https://www.sleuthkit.org/autopsy/ (visited on 2019).

[9] Patrick Kral. *Incident Handler's Handbook*. Tech. rep. SANS Institute, 2012.

[10] Paul Cichonski et al. “Computer security incident handling guide”. In: *NIST Special Publication* 800.61 (2012), pp. 1–147.

[11] Simson L Garfinkel. “Digital forensics research: The next 10 years”. In: *digital investigation* 7 (2010), S64–S73.

[12] Brian D Carrier and Joe Grand. “A hardware-based memory acquisition procedure for digital investigations”. In: *Digital Investigation* 1.1 (2004), pp. 50–60.

[13] ISO/IEC. *Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence*. Standard ISO/IEC 27037:2012. Geneva, Switzerland: International Organization for Standardization, 2012.

[14] Rick Ayers Sam Brothers Wayne Jansen, Rick Ayers, and S Brothers. “Guidelines on Mobile Device Forensics”. In: *NIST Special Publication* (2014), pp. 800–101.

[15] D. Brezinski and T. Killalea. *Guidelines for Evidence Collection and Archiving*. BCP 55. RFC Editor, Feb. 2002.

[16] ISO/IEC. *Information technology – Security techniques – Guidelines for the analysis and interpretation of digital evidence*. Standard ISO/IEC 27042:2015. Geneva, Switzerland: International Organization for Standardization, 2015.

[17] Karen Kent et al. “Guide to Integrating Forensic Techniques into Incident Response”. In: *NIST Special Publication* (Jan. 2006).

[18] Microsoft. *Volume Shadow Copy Service*. URL: `https://docs.microsoft.com/en-us/windows-server/storage/file-server/volume-shadow-copy-service` (visited on 2019).

[19] Mari DeGrazia. *SQLite Deleted Records Parser*. URL: `https://github.com/mdegrazia/SQLite-Deleted-Records-Parser` (visited on 2019).

[20] Atenea CCN-CERT. *Atenea Challenges*. URL: `https://atenea.ccn-cert.cni.es` (visited on 2019).

[21] Microsoft. *Microsoft Security Bulletin MS17-010 - Critical*. URL: https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010 (visited on 2019).

[22] FireEye. *SMB Exploited: WannaCry Use of "EternalBlue"*. URL: https://www.fireeye.com/blog/threat-research/2017/05/smb-exploited-wannacry-use-of-eternalblue.html (visited on 2019).

[23] *CVE-2017-0143*. Available from MITRE, CVE-ID CVE-2017-0143. URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143 (visited on 2019).

[24] Rapid7. *EternalBlue Metasploit Module*. URL: https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/smb/ms17_010_eternalblue.rb (visited on 2019).

[25] MITRE. *WannaCry MITRE Software*. URL: https://attack.mitre.org/software/S0366/ (visited on 2019).