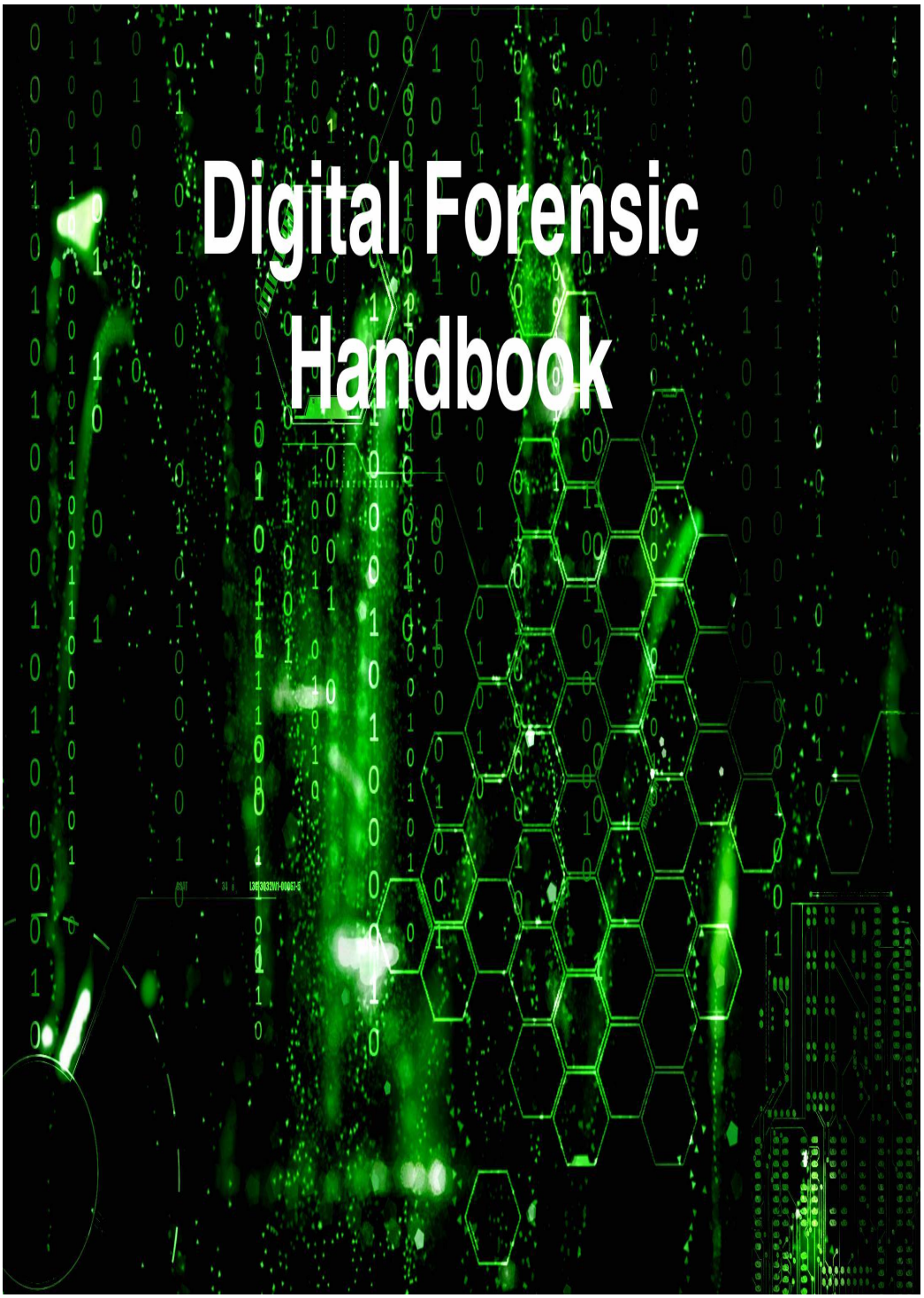

Digital Forensic
Handbook

Digital Forensic Handbook

Hochschule Ulm
University of Applied Sciences

By : Rahul Tak

Contents

List of Figures

Abstract

As growing explosively Internet, Computers and modern technologies are slowly taking control, as is the number of crimes that are committed with or against computers. In response to the modern crimes taking place, the field of Computer Forensics or Digital Forensics has emerged. Information Gathering, examining electronic evidences are becoming an type of issue in sky rocketing number of conflicts and crimes. Digital Forensics involves not only collecting data but also to recover the lost information from such a system to prosecute a criminal. With the growing number of computerised crimes called as "Cyber Crimes" it is important for IT professionals to understand the technology that holds a part of Digital Forensics. This paper will discuss more about Digital Forensics, tools, legal aspects, procedures, tasks involved, services and skills needed by professionals.

Crimes are getting more committed with the increasing sophistication of computers, different types of Cyber attacks like DOS, Ransomware, Viruses etc are just leaves of a huge tree of attacks that are taking place against other computers. There is a big necessity of IT managers understanding Digital Forensics. Digital Forensics is about analysing, collecting information and presenting in a readable format as an evidence in front of the court. Data on Hard Drive, fingerprints, transaction history, log files can be the forms of evidences that can be present in the court to prosecute.

Ensuring acceptance by the courts, procedures have to be adopted and presented which are not at all same as the Biological Forensics governed by Medicine, Science and the Law. Using the Digital evidences that with the already set up long, tried and tested scientific methods of the forensic community can help in learning and solving the cases.

Chapter 1

Introduction to Forensics

The new emerging and day to day changing technology brings us new, more sophisticated, high performance systems but as we now "A coin has two sides" this era of new technologies brings up also dangers, threats, loopholes etc which can be exploited by an attacker. Nothing can be 100% secure so we can learn from the mistakes, incidents and improve constantly. The biggest threat faced by industries, people and businesses is that from cyber attacks and threats, can also be called as cyber terrorism if they are in large magnitude and result of which can be felt on both regarding cost and human emotions. Then the question arises:

How did it happen ?

How can this be prevented ?

Depending on the severity of the incidents it can take up to 6 months or even a year to find the answers of these two questions. This is where the role of forensics come into play, For Example: Any remnants left after the cyber attack incident can be very carefully collected and examined by IT professionals and then from this point **WHO? WHEN? WHAT? WHERE? HOW?** questions can be answered.

Digital Forensics also known as computer forensics, e-discovery, cyber forensics, digital discovery, data recovery, data discovery, computer analysis etc. is a technical process that uses analysing techniques to gather information from electronic devices such laptop, desktop, hard disks, tapes etc for evidences. The process of locating, extracting, and analysing types of data from different devices, which specialists then interpret to serve as legal evidence in the court of law. In other words

Ít is the discipline that combines the elements of law and computer sci-

ence to collect and analyse data from computer systems, networks, wireless communications and storage devices in a way that is admissible as evidence in a court of law."

Digital Forensics process can be compared with the general biological forensics as they have the almost the same process for gathering data and analysing it, with a proper presentation in court of law.

1.1 Why this report on Digital Forensics?

Report on Digital Forensics was written in my Bachelorś Degree of Computer Science with specialisation in Information Security. This report intends to answer the question related to Digital Forensics regarding its procedures, processes involved, skills, legal aspects or about the forensic hacks. The first part of the report was written by me which includes the introduction, goals, legals aspects, process part of forensics, it took a whole amount of effort and hard work to complete the task. Although the huge effort was there but I enjoyed learning about digital forensics and got into deep with every step.

1.2 Goals

The goal of forensics is to find facts about an incident and via these facts to recreate the truth of an event that took place in the past. The forensics expert will present his discovered results on the basis of remnants of the event that have been left on the system. These remnants are more renowned as *artifacts*. *Artifacts* are the traces which are left behind due to activities and events.

"With contact between two items, there will be an exchange" stated by Locardś exchange principle is the underlying principle of all digital forensics. A person using a computer system will definitely leave traces of that activity on the system as there was an contact, there was an exchange. The goal is to retrieve the data from the traces left and interpret as much information about it as possible. Simple access to the system leads to small changes in the registers of the processors and on the other hand more complex actions leads to deeper impressions on the system both leaving traces behind. Even the act of cleaning the traces leaves traces like in a real life example of a burglar getting into a house unknowingly leaving behind the footprints.

The task of a forensic expert is to present the evidence in a presentable manner in the court of the law and not to prove these assertions. The task is to uncover artifacts that indicate the hypothesis to be either valid or not valid.

1.3 Objectives

A brief objective is to provide the evidences collected both in direct and indirect ways from the digital media. The keyword is "usable as evidence in court of law "**What kind of roles does computer plays in digital forensics?** Answer to this will help us understand the main objectives of Computer Forensics.

- *As contraband or fruit of the crime*

  Eg: Criminals having the copies of the software without the permissions of the software company is an violation of copyright law. In this case the computer where the software is installed is an contraband or a fruit of the crime.

- *As an instrument for the Attack*

  Eg: Criminal using computer to attack another computer on the internet using computer as an instrument.

- *As a tool which is incidentally used in the crime*

  Eg: Computers can used by criminals as to store account information or communicating with their accomplices and they are a part of a crime unintentionally.

- *As a tool both instrumentality to the offence and a storage device for evidence*

  E.g. A criminal uses a computer to attack another computer through the internet and stores the information and data he gets on the hard disk to share with each other.

Conclusion after understanding the role of the computers which helps us to get the main objective is as follows::

- Providing evidences which could be found directly. Eg: Data found on the digital media, pirated CDś, unauthorized access record to the other computers, and digital photograph which could prove guilt of a criminal.

- Providing evidences which could be found indirectly. This includes two parts. The first one is evidences obtained by analysing data obtained from the digital media, e.g. the calling record stored in a mobile phone which indicates a suspect has communicated with another one who has already been proven to be guilty. The other part is evidences that have been deleted but are still possible to be recovered using certain tools and technology.

1.4 The Digital Forensics Process

This section describes the rules, procedures of digital forensics. It can basically be divided into three categories: *acquisition, analysis* and *presentation.*

- *Acquisition* refers to the collection of the data which is to be examined. Physical hard drives, USB sticks, optical media, storage cards from digital camera, mobile phones, embedded devices chips, folders, text files etc can be used to collect data for any artifacts. The process of acquisition should include making copy of the original data and the examination process should be performed on the copy files. It should also maintain the records of all actions taken with any original media. *"Authenticity of the Original Media should be maintained".*

- *Analysis* refers to the processes dealing with the actual examination of the media which involves " *"identification, analysis and interpretation* of the data. Identification is recognising the important data and reducing this set of data to items or artifacts of interest. This leads us to examine the artifacts with different kind of analysis like file system analysis, file content examination, log analysis or statistical analysis. Lastly based on the analysed report the examiner will interpret results based on his experience, expertise and experimentation.

- *Presentation* is the process to share the results to the interested parties. This consists of writing a report of action taken by the examiner, artifacts discovered and the meaning of those artifacts.

1.5 The Crime Scene

On a crime scenario, experts from general forensic and the experts from digital forensic world have to work hand in hand. As soon as they are at the crime place, the first thing to be done is to seize the things, lock the place and to implement the process like 4 eye principle. They also ensure that the things on the crime is not set up by the criminals intentionally. If on the crime scene there are digital evidences, they are taken special care only by trained specific IT forensic team. Prior to this the forensic expert must take permissions from the relevant client to analyse the evidences.

Once the required permission is granted, the specialist starts the external examination of the media. The expert should also not change the original state, i.e. if an external storage medium such as a USB stick is connected, this should also not be removed directly from the system. Pictures, documentation of the crime scene and also the markings of the objects lying on the crime scene is very important as it will help to prove later how the crime took place.

The systems and electronic objects are being examined at the crime scene and as an expert, he/she has to make a decision quickly whether to continue to work on the real system "keeping in mind that the data might be lost" or to do a post analysis which is performed on a separate system with an image of the hard disks and RAM memory. The image file of the system or USB etc is an important asset to provide as an evidence in the court of law as the target system is the actual evidence and must be therefore remain unaltered.

Crime Scenario::

- 4 eye Principle :: Leads to Data Integrity
- Pictures and Documentation of the crime scene
- To allow only authenticated persons
- Securing the evidences
- Noting down all the actions taken on the place :: To prove it in the court

Step by step procedure::

- Seizing the place of the crime then secure the computers and digital media which might have evidences. Since the special characteristics

of digital data, it is more vulnerable and easier to be modified and destroyed. So it should be preserved very carefully and could only be accessed by authorised experts.

- Search for data existing in the digital media. The data can be of various types like log files, voices, text files, documents etc. It also distributes in different forms, e.g. files stored directly on the hard disk, have already been deleted but still acquirable, encrypted or hidden from unauthorised users.

- Dispose data collected. Tools can be used by the expert to collect deleted, decrypt information that has been coded, or reveal contents that have been hidden from the normal access.

- Analysing information. This process needs not only technical skills but also efficient tools, and experience is a must requirement for this process step.

- Creating a report which presentable and readable by a non-technical person keeping in mind of all the law aspects.

1.6 Classification of evidences

After the evidences are collected, now the forensic expert has to analyse the scenario and do the classifying of clues that were found based on how the attacker penetrated into the system and how the incident occurred. There can be many types of information classification.

1.6.1 Based on network

Network based attacks are easily recognisable and examined but there are many other network-side hints. For example: If there is an unusual network input and output traffic occurs, this might be an internal attack in which the attacker uses the victim system to hack third parties.

If there is a firewall placed between the internet connection and the companyś system, a higher number of firewall breaches may indicate an attack. Proxy logs is an good example to look at as an intruder uses the internet access port to send the data on the internet. [Cf. Alexander Geschonneck, p.50 f.]

1.6.2 Based on server

Attacks on servers, the server-side hints may indicate that there was an attack on the server like Denial-of-Service etc. In this case, the expert has to have good knowledge with the server and the its operations.

- Repeating of the unknown process of the same name –¿ indicate that a loop is running like "brute force"
- Unknown data with special rights stored in an unusual location and with an unusual name
- Unusually high system load
- Unusual logins at suspicious times
- New and unknown users with special rights [Cf. Alexander Geschonneck, p.51 f.]

1.7 The Forensic Technician

The goal of the forensic expert is to find out the criminal of the case he/she is working on. The main tasks involve:

- The crime scene has to be frozen, completely seized, that is the evidences should be collected as soon as possible and without any alteration.
- Chain of custody known as the continuity of the evidences, that means to have all the information for all that has happened to the evidences between its original collection and its appearance in court, preferably unaltered.
- All procedures used in examination should be audit able, that means an investigator from the other side in a case should be able to track all the investigations carried out by the prosecution expert.

For more complicated cases, involvement of different networking or malware like logic bomb or virus suspected , use of special forensic experts tools and experts are needed. Special training is also required. The key features of forensic technician is as follows:

- Methodology of approaching, record keeping
- Differentiation of relevant and irrelevant data
- Collection of evidence in accordance with the law
- A sound knowledge of legal procedures
- Access to and skill in the use of appropriate utilities
- He should also have psychological experience : helps in proving fast in the court
- Very good knowledge in technical areas such as file systems, networking systems

  It is important, among other things, to be able to assess and deal with the client, because the client could also be a part of the case and may give false information.

1.8 Forensic Tools List

There are many tools for performing different digital forensics process. Most of the tools are based on UNIX environment as nowadays most of the servers and systems runs on UNIX environment.These tools are really helpful and easy to use which makes the work of the forensic expert very easy, effective, automatic and complete.

- SIFT The SANS Investigative Forensic toolkit is an Ubuntu based tool helps in conducting an in-depth forensic or incident response investigation.
- ProDiscover Basic is an investigation tool, helps in analysing and reporting of the evidences found on the disk.
- Volatility a framework helps in for incident response and malware analysis.
- The Sleuth kit helps in having an in depth analysis of file systems.
- CAINE Computer Aided Investigation Environment is Linux Live CD that contains a wealth of digital forensic tools.

- DEFT is also an Live CD helps in Incident Response, Cyber Intelligence and Computer Forensics scenarios.

1.9 Services Offered by the technician

The most important thing in the investigation of a digital crime and the case that is to be presented in the court is the evidences that are being found and the type of investigation steps that are involved. Therefore it is very important for an client and the court to hire a digital forensic professional which can provide authenticated services. No matter how careful they are, when intruders try to steal electronic information like customer data, databases etc. they leave behind traces of their activities. A computer forensic expert does more than turn on a computer, searching for important files. The expert should be able to successfully perform complex evidences recovery procedures with the skill and expertise that leads credibility to the case. The experts should be able to perform and provide services as follows::

- Data seizure :: collecting evidence at a crime scene given by the client in accordance with the law
- Duplication of data and preservation of data
- Data recovery :: Find evidence in hidden areas of memory
- Document searches :: Carving all deleted files
- Media conversion
- Expert witness services :: To testify as an expert and appraiser before a judicial jury
- After the investigation and archiving of the evidence, the experts assist in reconstruction
- Standard Service
- On-site Service
- Emergency Service
- Priority Service
- Weekend Service

1.10 SAP Model

Many companies, organisations and government institutions have an Incident Detection Team which through certain preparatory measures, employee training awareness and other strategies, tries to uncover and correct unusual behaviour in the IT system as quickly as possible. The so-called first responder, is the person who first notices the anomaly and initiates further steps by reporting it to other external investigative authorities such as police and IT forensics companies. If it turns out that the anomaly is an attack, the IT experts go through a procedure to secure concrete evidence. There are many different models to do this. One of the best known models is the (SAP) model **Secure Analyse Present** which is used as a general overview due to its high abstraction level.

1.10.1 Secure

The digital crime scene and task of IT forensics described should strictly preserve the properties of the evidence, on the basis of admissibility in the later course. In the securing phase, the first step is to close off the crime scene, apply the four-eye principle and log every step.

The data and evidence is then collected. [Cf. Alexander Geschonneck, p.68 f.]

The data found must now be analysed in the next step.

1.10.2 Analyse

In the analysis phase, the track found must be thoroughly checked. Intermediate results must always be critically questioned in order to find avoidable gaps in the argumentation chain.

The relevant evidence data is separated from the remaining unusable data. There are two different techniques to do this. Either a live response or post-mortem analysis. The first technique is performed while the system is running, while the other technique is only performed by means of an image, i.e. a memory duplicate. The extracted data must now be viewed in a new light. This is a big challenge for the IT forensic expert, because he has to be exactly know what he is doing, what laws he could violate and what exactly he has to look for to get a conclusive result. Therefore it is very important to train and educate the staff at all times. [Cf. Prof. Dr. Dreo, Gabi et al.., p.13-14][Cf. John R. Vacca, p.224]

1.10.3 Present

The evidence collected in the securing and analysis phase should now be recorded and prepared for the court hearing for the target group. Suitable for various bodies and legal representatives with little technical knowledge, the results are now presented. The final result of the investigation should include the following points: Information about

- The identity of the intruder, hacker
- The scale of the attack
- Motivation and intention
- Reasons and method of implementation
- time of the crime

[Cf. Alexander Geschonneck, p.68 f.]

1.11 Relation to Classical Forensics

Classical and digital forensics differ fundamentally based on the type of crime committed and the type of criminal involved. In classical forensics humans

or objects like guns are basically the reason for the investigation and on the other hand in the digital forensics, data is the main reason of investigation. Both forensics go hand in hand once there is an involvement of both humans and electronics at the same crime scene. Data is the most important thing in digital forensics and the term for collecting data can be compared to the term of collecting fingerprints from the murder scene. We can compare taking fingerprints from the murder scene with collecting data, hidden data or anything leading to help finding the criminal. The main goal of the both types of forensics is the same only they differ in the type of process they use. Once the different process are done performing, they are presented in front of the court. The evidences are treated similarly by law, both the forensics follow procedures, rules and regulations for the presentation of the evidences with maintaining the authenticity, accuracy and completeness of the evidences.

1.12 Documentation of the investigation

Documentation is one of the most important parts of the investigation, as because without proper, credible and authenticated documentation the case can really complicated and it can be really hard fighting with the other party in the court. It is therefore important to have a look at certain rules.

Like every report that is to be presented in the court has to be written neutrally, objectively and without emotions. Document must be very detailed, specifying details about the crime scene, pictures, what has been done on the evidences, what processes are being done?

1.13 Field set check list for Forensic expert

The forensic expert should always be prepared for all types of cases, so his field set should include the following:

- Camera with sufficient memory and battery
- Bootable USB with Cain or Kali Linux
- Various connection cables (network cable, USB cable,...)
- Software tools for collecting and analysing data

- Write Blocker, Pens, forms
- radio-controlled clock
- Various storage media with sufficient storage space
- packaging materials such as plastic bags for evidence
- Stickers for labelling
- WLAN adapter for WLAN sniffing
- Latex Gloves

  [Vgl. Kuhlee, Lorenz, S.5]

1.14 Challenges, Problems and Limitations of Digital Forensic

1.14.1 Challenges faced

- Constant technological change, with the result that forensic analysis methods are constantly changing[Cf. John R. Vacca, Chapter: Fundamentals p.24].
- To find out the real evidence that can stand up in the court of law and its legal/provable.
- To be able to explain evidence and analytical methods in an understandable and simple way.
- Finding out the role of the computer. Is the computer the target, an instrument or it stores evidence.[Cf. John Vacca, Chapter: Fundamentals p.6].
- Conflict between accuracy, carefulness and time pressure[Cf. John R. Vacca, Chapter: Fundamentals p.7].

1.14.2 Problems faced

- First problem is with time, as it changes and so does information systems are changing.[Cf. John R. Vacca, Chapter: Fundamentals p.19]
- Data is invisible to humans and must be made visible with the help of tools. These tools can be challenged in court if they are not verifiably error-free. [Cf. John R. Vacca, Chapter: Fundamentals p.19]
- Methods for proving can always be challenged as the means of proving technology are changing rapidly.
- Facts of the data to be examined. Is the data authentic?
- To prove reliability, completeness, free of contamination and falsification? [Cf. John R. Vacca, Chapter: Fundamentals p.21 f.]

1.14.3 Limitations

- Privacy concern as it limits the scope of investigation.
- Legal disputes at a technical level are partially settled in court. As these are very complex and complicated[Cf. John R. Vacca, Chapter: Fundamentals p.20].

1.15 Legal Aspects

1.15.1 Criminal Law

Criminal law comprises all legal norms that determine the content and scope of state criminal powers. The first paragraph of the Criminal Code regulates in general when an offence can be punished. At the time when the offence is committed, there must also be an appropriate law that identifies the offence as a criminal offence. Even if the act is morally unjustifiable but there is no law against it, it is not a crime. Forensics professionals should know certain extracts from the criminal code in order to know which offences can actually be reported as criminal offences.

Furthermore, not all possible technical examination methods are legal. For this reason, the Criminal Code must also be taken into account in investigations to prevent forensic crimes being committed in the course of an investigation.

Section 34 Necessity

Forensics investigators must also ensure that they **comply with legislation in their investigations**. Criminal law makes no distinction between a forensic investigator (doing things without permission) and a criminal. However, criminal law contains Section34, which defines the state of emergency. This describes a situation in which a crime may be committed under certain circumstances in order to avert a danger.

However, the hazard is precisely defined. Furthermore, a balance must be struck between the danger and the interest to be protected. It must be determined that the interests to be protected predominate significantly. If all these points are met, an appropriate means may be used to avert the hazard. If an offence is committed as a result, this is not an offence.

Section 202a Data Espionage

Similar to the secrecy of correspondence, there is also a penalty for spying out data. If a document is opened by a person who is not intended for that person and is secured against inspection, it commits an criminal offence, Section 202 a. Violations of the secrecy of correspondence are punishable by a fine and a custodial sentence of up to one year.

The same principle can be applied to data that may need to be analysed by a forensic expert. First of all, data is information that is stored or transmitted electronically, magnetically or otherwise not perceptible. These can be files on a hard disk, but also data that is transported over a network. Now Section 202a, regulates under which circumstances a criminal offence is committed during the inspection of data. As soon as a person does not have the permission to view the data and has to overcome access protection, he or she commits a crime.

Overcoming access security means, for example, overcoming an encryption or a password. Even if the person works on behalf of a client, he or she is liable to prosecution. Violations of Section 202 a are punished with up to three years imprisonment.

Section 202b, 202c Phishing and data espionage

These sections Section 202b interception of data, Section 202c preparation of spying and interception of data and Section 202d data monitoring. This enables an expert to detect crimes and prevent them from being committed even through carelessness. According to Section 202b, intercepting data from a non-public network or spying out electromagnetic emissions from data processing systems is already an criminal offence. Therefore, the data traffic of an external network must not be read and analysed without the required permissions. In addition, Section 202c already defines the possession of programs with the purpose of spying out and intercepting data as a criminal offence.

Trading with illegally obtained data is also a criminal offence and falls under Section 202d Paragraph. As soon as the data gets itself or someone else or made available in any other way with the aim of enriching oneself or others or harming someone, a criminal offence is committed. This also means that the acquisition of this data for forensic investigations is a criminal offence, as a third person is paid for the purchase and thus enriched.

Section 203 , 204 Secrets of other persons

Experts use undisclosed secret data in their work, which inevitably contains secrets in the legal sense. Likewise, the disclosure of third-party data content to clients or judicial bodies is part of forensic activities. With regard to the disclosure or exploitation of foreign secrets, Section 203 and 204 are thereby are really important. As of Section 203 "a secret belonging to the personal sphere of life or a trade or business secret [...]". Certain persons are obliged to maintain secrecy on the basis of their profession in accordance with Section 203. A disclosure of secrets without permission is always punishable. Those who work for these groups of persons in accordance with Section 203 para. 3 StGB are equally obliged to maintain confidentiality. A forensic expert who inspects data carriers for a lawyer would be treated in the same way and would therefore be obliged to maintain confidentiality.

The exploitation of secrets is further defined in Section 204, in which the unauthorised exploitation of secrets is punishable. For this to happen, however, there must be a breach of the duty of confidentiality pursuant to Section 203. It should be noted that the disclosure of secrets as well as the exploitation of such secrets according to Section 203 and 204 is punishable

even after the death of the person concerned.

Section 303 Criminal Damage, Data Tampering

Digital forensics aims at changing existing data in order to find further data hidden in it. Although forensic methods usually never changes the original data, forensics professionals should be aware of the criminal consequences and the occupational risks associated with their work.

Accordingly, data in accordance with Section 202 on the violation of the secrecy of correspondence are to be understood analogously to the document and its content. In contrast to documents, the term "only those data which are stored or transmitted electronically, magnetically or otherwise not directly perceptible" is used in accordance with Section 202a. Deleting, suppressing, rendering unusable or modifying data is data modification. Illegal data modification in the sense of Section 303a is punishable by law. According to Section 303a even the attempt to change the data failed or prevented is punishable.

If the data change in this sense is aimed at data processing, which is "of essential importance for an external company, an external company or an authority," such a data change is punishable under Section 303b. If a data processing system or a data carrier is destroyed, damaged, rendered unusable, removed or changed as a result of the data change in accordance with Section 303b, the penalty will be higher. Even the attempt according to Section 303b is punishable.

Nevertheless, the illegal actions described in Section 303a and 303b are not automatically prosecuted. These are only initiated upon criminal complaint by reporting the injured party. Exceptions are cases in which criminal prosecution by the public prosecutorś office can also be initiated without a criminal complaint due to the public interest. Forensics professionals should always be aware that they work as external employees for a "Client company", and therefore there is always the risk that data changes can have legal consequences, even if unintentionally.

1.15.2 Corporate Law/ Right of Co-determination

Section 91 Organisation, Accounting

German Stock Corporation Act (Aktiengesetz, AktG)

Section 91 (1) of the German Stock Corporation Act stipulates that the Management Board is obliged to keep commercial books. This is interesting for the experts if the investigation is on a financial level and/or the experts wants to get a rougher picture of an alleged criminal or needs other additional information. Likewise in Section 91 under (2), the Management Board has the duty to avoid risks that could impair the continued existence of the company by means of appropriate monitoring methods. Among other things, this means acting "forensics ready".

Right of Co-determination

Handelsgesetzbuch (HGB) und Betriebsverfassungsgesetz (BetrVG)

According to Section 87 of the BetrVG, the right of co-determination applies in companies: "The works council has a say in the following matters, unless there is a statutory or collective agreement: the introduction and use of technical equipment designed to monitor the behaviour or performance of workers[...]" This means as already in chapter: What "Forensics Ready" means is that the works council has the right of over all internal monitoring.

1.15.3 Section 22 Personal Rights

"Portraits may only be distributed or publicly displayed with the consent of the person depicted. In case of doubt, consent shall be considered given if the depicted person received remuneration for having himself depicted. After the death of the person depicted, the consent of the of the person depicted. Relatives within the meaning of this Act are the surviving spouse or partner and the children of the person depicted and, if neither a spouse or partner nor children are present, the parents of the person depicted.

The marked text passages show that the IT forensic expert quickly finds himself in an area of personal rights. He should therefore always pay attention to the correct handling of the data found and avoid unauthorised actions.

1.16 Standards and Best Practices

ISO 27001:2013

Excerpt: Information security management system – Requirements

A.16 Information security incident management		
A.16.1 Management of information security incidents and improvements		
Objective: To ensure a consistent and effective approach to the management of information security incidents, including communication on security events and weaknesses.		
A.16.1.1	Responsibilities and procedures	Control: Management responsibilities and procedures shall be established to ensure a quick, effective and orderly response to information security incidents.
A.16.1.2	Reporting information security events	Control: Information security events shall be reported through appropriate management channels as quickly as possible.
A.16.1.3	Reporting information security weaknesses	Control: Employees and contractors using the organization's information systems and services shall be required to note and report any observed or suspected information security weaknesses in systems or services.
A.16.1.4	Assessment of and decision on information security events	Control: Information security events shall be assessed and it shall be decided if they are to be classified as information security incidents.
A.16.1.5	Response to information security incidents	Control: Information security incidents shall be responded to in accordance with the documented procedures.
A.16.1.6	Learning from information security incidents	Control: Knowledge gained from analysing and resolving information security incidents shall be used to reduce the likelihood or impact of future incidents.
A.16.1.7	Collection of evidence	Control: The organization shall define and apply procedures for the identification, collection, acquisition and preservation of information, which can serve as evidence.

Figure 1.1: Information security management system

Response to information security incidents A 16.1.5

- A documented procedure should be followed to respond to security events.
- Collecting evidence as soon as possible after the occurrence.
- Performing a security forensic analysis.
- The existence or details of a security event should be discussed with all persons who have the right to know.
- Acting with security weaknesses, may have contributed to the incident or cause.
- Once the incident has been processed successfully, it should be formally closed and recorded.

Collection of Evidence A 16.1.7

The organisation should agree and apply procedures to identify, collect, acquire and present information that can be used as evidence. These procedures should also match different types of media, devices, or different device statuses.

The procedures should be as follows:

- Chain of evidence
- Securing evidence
- Personal evidence
- Define roles and responsibilities of the people involved
- Competence of the personnel
- Documentation

Chapter 2

Forensic Hacks

2.1 Digital Traces on Linux

2.1.1 Introduction

Talking about Linux, it is almost everywhere, from phones, refrigerators, supercomputers, home applications or even as backbone of the internet. Although for now it is not so common for the home computers but definitely in the coming years, it going to take over everything.

Usually, most of the servers and supercomputers are running on Linux Operating Systems and more and more users are shifting to use different Linux Distributions. Being free and open source, many companies are opting to choose Linux over Windows (which is costing them licence fees).

We are going to perform a series of forensic steps to find out the configuration of a machine running on a Linux distribution. Which includes:

- Determining Present Linux Distribution
- Taking a look at the Partition Structure of the Memory 1. SYS V architecture 2. BSD architecture
- What software is running?
- What Network services are running?
- Determining Network-configuration

As a Forensic expert, knowledge of different operating systems is a must. Most of the forensic tools are based on a Linux environment and utilising knowledge with Linux based tools will make complicated things very easy.

In this Hack we are going to show; what exactly needs to be done? Which steps to follow? and How to proceed further when given a Linux machine for performing forensic examinations.

2.1.2 Determining present linux distribution

At the very start of each forensic examination, you need to be able to tell which distribution of the Linux operating system is present. Due to a high degree of standardisation in the UNIX operating system domain there are a lot of similarities between different Linux systems but despite that, there are two major Linux architectures: BSD and Sys V. After the birth of the UNIX operating system in 1969 the Californian Berkeley University further developed this software architecture and released 1977 the UNIX distribution: Ḃerkeley Software DistributionḂSD. Sys V, spoken Ś́ystem five; was one of the first commercial distributions of UNIX, developed by AT and T in their prestigious Bell labs. In the course of time more and more systems were developed basing on these two architectures but although there were quite some standardisation efforts, for example by POSIX, ISO or IEEE, differences in commands or paths for log files are still commonly observed.

Sys V:

- Debian
- Ubuntu
- RedHat
- HP-UX
- Oracle Solaris

BSD:

- OpenBSD
- FreeBSD
- Mac OS X

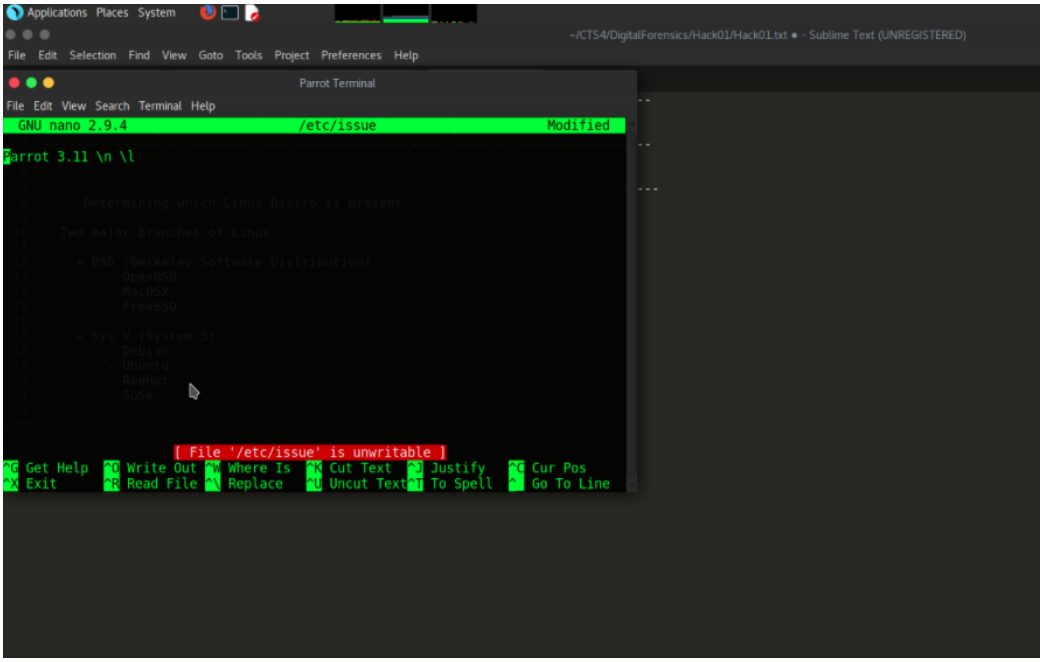


Figure 2.1: Linux Distribution

In an ongoing forensic investigation where you have access to an image of the to be examined system, the Linux distribution can be determined by having a look in the following files:

cat /etc/issue or cat /etc/issue.net

This íssuefile contains a message or a system identification which is displayed before login, by default this contains the name of the currently installed Linux distribution. But by writing random characters into this file, a system administrator, for example, could permit the acquisition of this information. If no knowledge could be gained by these files, further investigations need to be carried out. Another way of determining the two different distributions is by look for software architecture specific differences like: - name of the device of the root partition: Sys V: /dev/sda1 ; BSD: /dev/disk0s1 - the password files: Sys V: /etc/passwd OR /etc/shadow BSD:/etc/passwd OR /etc/master.passwd an extensive list of differences can be found on :: http://unixguide.net/unixguide.shtml

2.1.3 Getting an overview of the partition

After successful determination of the present Linux distribution it is highly advisable to uncover the structure of the storage media at hand. Thus its necessary to know the layout of this very storage media, the file system type, the partitions and where these partitions are mounted. If we have this information in an forensic examination we can narrow down the area to be forensically examined. To determine the disk properties we need to differentiate again between Sys V and BSD architectures.

Sys V: For Sys V architectures the storage media can be evaluated using the fdisk utility with elevated privileges.

sudo fdisk -lu

fdisk is a terminal based program which can create and manipulate partition tables. Partitions, which are just simple block devices divided into one or more logical disks, are allocated in the partition table of this storage device, usually in sector 0. By reading / writing to this partition table, fdisk is able to achieve its purpose.

Another way of determining the partitions of the storage media at hand is using mlls, which is part of the Sleuth-Kit. mmls also lists the content of the partition table, but specifies additional size information making it easy to use ́ddór any similar tool to image partitions. mmls is very similar to fdisk -lu, but also shows unused sectors on the disk, BSD partition tables and accepts forensic disk image files making it very attractive for forensic purposes.

mmls -t dos disk.dd

BSD: After discovering what kind of Linux distribution the machine is running on and seeing the SysV architectureś partition table. Letś now look at how partition table look like in a BSD architecture. This will help us to perform further steps on the server knowing clearly about the partition of a particular type of architecture.

In principle the partition of BSD systems are not that different from the Linux or Windows systems, therefore it is also based on DOS - partition table. To find out how the partition table in an BSD architecture looks like we can use the same commands as what we did in the last Hack about SYS V architecture.

Tools:: BSD architecture based machine like MacOS Terminal

First of all we check what all partitions are there on our Mac OS machine:

command: diskutil list

```
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                 DOS-Partition                   *256.1 GB   disk0
   1:                         EFI EFI                     209.7 MB   disk0s1
   2:                 Apple_APFS Container disk1          255.9 GB   disk0s2

/dev/disk1 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +255.9 GB   disk1
                                 Physical Store disk0s2
   1:                APFS Volume Mac OS                   144.1 GB   disk1s1
   2:                APFS Volume Preboot                  22.7 MB    disk1s2
   3:                APFS Volume Recovery                 517.8 MB   disk1s3
   4:                APFS Volume VM                       5.4 GB     disk1s4

/dev/disk2 (disk image):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     Apple_partition_scheme                         +24.2 MB    disk2
   1:        Apple_partition_map                          32.3 KB    disk2s1
   2:                  Apple_HFS Flash Player             24.2 MB    disk2s2
```

Figure 2.2: Partition Table

As we now know the types, names and sizes of all the disk running, we can proceed further by using the command.

Command sudo mmls /dev/disk0

```
DOS Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Safety Table
001:  -------   0000000000   0000000039   0000000040   Unallocated
002:  Meta      0000000001   0000000001   0000000001   GPT Header
003:  Meta      0000000002   0000000033   0000000032   Partition Table
004:  000       0000000040   0000409639   0000409600   FreeBSD(0xA5)
005:  001       0000409640   0500118151   0499708512   Mac OS
006:  -------   0500118152   0500118191   0000000040   Unallocated
```

Figure 2.3: BSD Architecture

We look closely at the FreeBsd (0xA5) with the command.

sudo mmls -o 40 /dev/disk0

As now we have all the individual partitions listed which makes our evaluation of the disk much easy.

2.1.4 Determining installed softwares

It is very important to have a look at the installed software as it will show what all kind of softwares are installed, are they installed by the administrator or hacker, are there any back doors running, what time and date they were installed, where they were installed, is encryption available etc.

In Linux there are two options to install different software one is using packet manager and the other is installing directly from the sources of the software. To know more about the software running, the packet manager is itself used to keep track of all the details of the installed softwares.

The two most common packet managers available for Linux are Red Hat Packet Manager (RPM) and Debian Packet Manager (DPKG).

Lets start by checking all the running softwares on the system. Seeing the list we can say what softwares are installed, are there any softwares running without purpose. Command dpkg -l

rpm -qa

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                                  Version                                 Architecture Description
+++-=====================================-=======================================-============-
ii  0trace                                0.01-3                                  amd64        A traceroute tool that can run within an existing TCP
connection.
ii  aapt                                 1:7.0.0+r33-1                           amd64        Android Asset Packaging Tool
ii  acccheck                             0.2.1-3                                 all          Password dictionary attack tool for SMB
ii  accountsservice                      0.6.45-1                                amd64        query and manipulate user account information
ii  ace-voip                             1.10-1parrot0                           amd64        A simple VoIP corporate directory enumeration tool
ii  acl                                  2.2.52-3+b1                             amd64        Access control list utilities
ii  adduser                              3.117                                   all          add and remove users and groups
ii  adwaita-icon-theme                   3.28.0-1                                all          default icon theme of GNOME
ii  afflib-tools                         3.7.16-3                                amd64        Advanced Forensics Format Library (utilities)
ii  aglfn                                1.7-3                                   all          Adobe Glyph List For New Fonts
ii  aircrack-ng                          1:1.2-0~rc4-4                           amd64        wireless WEP/WPA cracking utilities
ii  airgeddon                            8.01+parrot0                            all          a multi-use bash script to audit wireless networks
ii  albatross-gtk-theme                  1.7.4-1                                 all          dark and light GTK+ theme from the Shimmer Project
ii  alsa-firmware-loaders                1.1.3-1                                 amd64        ALSA software loaders for specific hardware
ii  alsa-tools                           1.1.3-1                                 amd64        Console based ALSA utilities for specific hardware
ii  alsa-utils                           1.1.3-1                                 amd64        Utilities for configuring and using ALSA
ii  amap                                 5.4-4                                   amd64        next-generation scanning tool for pentesters
ii  amd64-microcode                      3.20171205.1                            amd64        Processor microcode firmware for AMD CPUs
ii  android-framework-res                1:7.0.0+r33-1                           all          Android platform framework resources
ii  android-libaapt                      1:7.0.0+r33-1                           amd64        Android Asset Packaging Tool - Shared library
ii  android-libandroidfw                 1:7.0.0+r33-1                           amd64        Android utility library
ii  android-libbacktrace                 1:7.0.0+r33-2                           amd64        Android backtrace library
ii  android-libbase                      1:7.0.0+r33-2                           amd64        Android base library
ii  android-libcutils                    1:7.0.0+r33-2                           amd64        Android utils library for C
ii  android-liblog                       1:7.0.0+r33-2                           amd64        Android NDK logger interfaces
ii  android-libunwind                    7.0.0+r1-4                              amd64        libunwind for Android
ii  android-libutils                     1:7.0.0+r33-2                           amd64        Android Utility Function Library
ii  android-libziparchive                1:7.0.0+r33-2                           amd64        Library for ZIP archives
ii  anonsurf                             2.6+parrot2                             all          AnonSurf
```

Figure 2.4: Packages

To get every detail of the linux commands there are Ḿan pagesẃhich explains the details about the particular command with the usage as well the understanding of the command. There are ḿan pagesf́or every command and that might not be a good idea to read every bit of those pages for a small question. To handle this event of action we have a special command Ápropos ẃhich searches the manual pages for a special keyword or

regular expression. Each of the manual pages has a short description and ápropossśearches for the keywords.

eg:

apropos mount

In this command ḿountẃill be used as a keyword and ápropos ŕeturns all the man pages including that term.

What to do if the software was installed using source filesŤo find out about the softwares installed manually using source file we used commands like ẃhereis, which, locateẁhich shows the path of the installed software. The only thing to notice is to make sure that the Program Path for the relative software in the PATH variable (echo PATH).

```
user@rz-linea:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games
user@rz-linea:~$ whereis xclock
xclock: /usr/bin/xclock /usr/bin/X11/xclock /usr/share/man/man1/xclock.
1.gz
user@rz-linea:~$ which xclock
/usr/bin/xclock
user@rz-linea:~$ locate xclock
/usr/bin/xclock
/usr/share/man/man1/xclock.1.gz
user@rz-linea:~$
```

Figure 2.5: Source Path

Finding more details about the software using timestamp to determine date and time of installed software. There are three distinct types of values are stored for each of software as defined by the POSIX standard. Each file has three values

- The time of last data access. (atime)
- The time of data modification. (mtime)
- The time of file status last changed. (ctime)

We made a sample test file to check all the stats.
touch test stat test

```
user@rz-linea:~$ stat test
  File: 'test'
  Size: 0           Blocks: 0          IO Block: 4096   regular empty file
Device: 813h/2067d  Inode: 657740      Links: 1
Access: (0644/-rw-r--r--)  Uid: (83672/     user)   Gid: (10513/domain
users)
Access: 2018-05-01 23:15:07.970992146 +0200
Modify: 2018-05-01 23:15:07.970992146 +0200
Change: 2018-05-01 23:15:07.970992146 +0200
 Birth: -
```

Figure 2.6: Status

In Linux there is a command ́historyó see all the commands a user used recently. This is a very useful to know what things have been done on the terminal which will help us in analysing who dealt with the machine at hand.

Checking all the softwares like this will answer our questions about the version, user, owner, who installed it Hacker or administrator; when it was last accessed etc which would help in analysing our server programs and to get into deep.

2.1.5 Determining running network services

Nowadays, the usage of network services at private homes is steadily increasing, from the use of smart phones to smart light bulbs everything is connected to some sort of networks. But not only the private sector is heavily relying on network services but also the corporate and federal sector, ranging from small enterprises to global players like Google and co. This huge responsibility sometimes does not make it possible to conduct forensic examinations on disk images (dead systems) but on running, live systems. Furthermore there is quite a big difference in legality if illegal contents are not only stored but also shared. The most frequently used network services for illegally sharing files are SSH, HTTP, FTP and BitTorrent.

In general, network services can be divided into two categories, standalone network services and superdaemon network services.

Standalone network services

A typical standalone service is ssh, the secure shell daemon. Here, the client sets up a socket by looking for a free port on client side and then connecting to the servers port 22, therefore the secure shell daemon sshd needs to be running in the background. The config file for the ssh daemon is located, like almost all configs, in /etc. Usually the service starts automatically after booting, which can be verified by the syslog, but if it is started manually, we need to have a look in .bash history.

Command ls -l /etc/rc*.d/* — grep -i ssh

```
lrwxrwxrwx 1 root root  13 Mar 31 12:55 /etc/rc2.d/K01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root  13 Mar 31 12:55 /etc/rc3.d/K01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root  13 Mar 31 12:55 /etc/rc4.d/K01ssh -> ../init.d/ssh
lrwxrwxrwx 1 root root  13 Mar 31 12:55 /etc/rc5.d/K01ssh -> ../init.d/ssh
```

Figure 2.7: sshd

If we need to take a look into the .bash history command, we can do this by using the history command. The history command records each line inputted to a terminal at any time in the above mentioned .bash history file. To see if a given service was manually executed we have to examine this file.

To determine if the target machine is running a web server, we also take a look in the /etc directory. Here we search for installations of an apache / apache2 server, which stores its config under: - /etc/apache2/apache2.conf /etc/apache2/ports.conf

Super daemon network services

Whereas network services like ssh are running all by themselves, the super daemon network service isńt and the client needs to request the super daemon to start. The super daemon then wakes up the network service and establishes a connection to the client. The super daemon on Linux is called inetd or xinetd which is inetdś successor. Like for most other programs the configuration for these daemons can be found in their respective directory in /etc. - /etc/inetd.conf - /etc/xinetd.conf

Another typical example for a super daemon network service would be telnet. Before starting, the telnet service can ask a security service, called the

TCP ẃrapper, if the requested IP address is blocked. The TCP-wrapper does this by looking into the following files, specifying if a connection is allowed or not: - /etc/hosts.allow or /etc/hosts.deny

2.1.6 Determining network services

After having a closer look to the network services running on the server, its very important to know how our network configuration part works.

IP address (Internet Protocol) assignment of these address can be done in two ways using static or dynamic DHCP (Dynamic Host Configuration Protocol) protocols. Static Configuration means configuring and assigning IP address to the server and other machines manually. We can do this by going to the location :

/etc/network/interfacesánd then configuring our desired addresses.

eg:

auto eth0 iface eth0 inet static address 192.168.0.100 netmask 255.255.255.0 gateway 192.168.0.1

DHCP

auto eth0 iface eth0 inet dhcp

DHCP is a network management post used on TCP/IP networks whereby DHCP server dynamically (automatically) assigns an IP address and other network configuration to each device on a network so they communicate with other IP networks. In home and small companies this done by router in the network.

To check the basic configuration of the DHCP ¿ /etc/dhcp.conf ‹

This link here will explain about How to install, configure and manage DHCP in Linux server. cf. https://www.brennan.id.au/10-DHCPServer.html

DHCP server Persistence is the ability to save the lease information and to provide the client a IP address which does not conflict with other devices on the network, even after the devices are rebooted. It basically stores all the information on the flash memory and provides two functions ¿ IP address uniqueness ¿ Ease of use (Automatic restoration)

Determining IP address

Determining IP address of a live system: To determine the ip address of running server we use commands.

ifconfig -a netstat -in

Netstat

To check all the TCP connection running on our system n̈etstatd̈isplays network connections through the TCP (Transmission Control Protocol), routing between the connection in form of tables, including the number of network protocols running on the network interface and statistics of the network.

eg:

netstat

This command will show all the above properties running on the system.

Services

To start, stop and restart different services on the Linux machine there is a command service which is used to run a System V init script. All the scripts related to the command are stored in the /̈etc/init.dd̈irectory. All scripts stored in the directory can be start, stop and restarted by the śervicećommand.

As we now know most basic information about our Linux machine, we can go further in the process by applying other methods for analysing and digging deeper into the system.

2.2 File Carving

2.2.1 Introduction

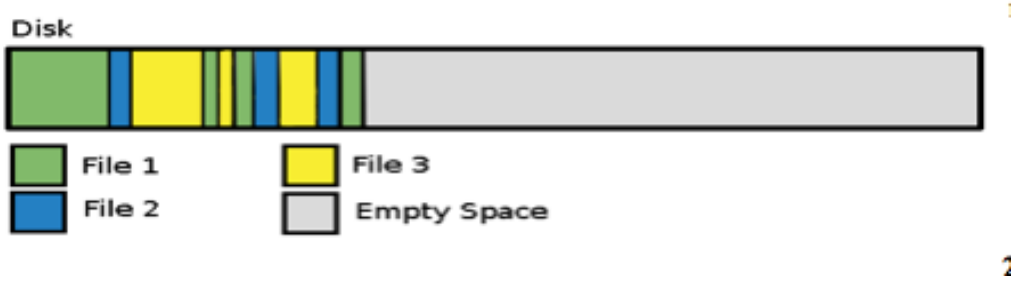
Data Carving is about identifying and extracting file types out of undifferentiated blocks, raw data. This is done by analysing file formats. This paper discusses two techniques of data carving. First, we will start with some basics about file systems and fragmentation. Later we will explain the methods of byte-based data carving and block hashed data carving. The first one focuses on searching for data byte by byte, comparing the file structures and extracting the found files. This is a very simple method and limited to non-fragmented data. The second method is restricted on detecting the presence of a specific file by comparing hashes, however itś not affected by fragmentation. Of course, the examples used in this paper are very straightforward examples to understand the basics of the underlying carving methods. Further development of these examples can minimise carving time and increase the accuracy of the results.

2.2.2 File Systems

The file system is a structure for storing and organising computer files and their metadata on the hard drive, memory card, USB, etc. File systems locate the files in blocks. Blocks are always occupied in whole, even if only one bit is stored in it. If the size of a file is smaller or equal to the size of a block, itś stored in one, single block, otherwise, the file occupies more than one block. But if the next block is already occupied, the file system must find the next free block. These splitting of blocks belonging to one file is called fragmentation. Of course, larger files are more likely to be fragmented.

FAT

FAT stands for File Allocation Table. In particularly, Fat is used for USB sticks and external hard drives, not only used with Windows OS. There are some different types of FAT: FAT12, FAT16, FAT32. The Cluster size is 16KiB3. FAT is storing data in a very simple way, saving data in a row and causing a lot of fragmentation.



NTFS

NTFS stands for New Technology File System. It is particularly used for the inbuilt hard drive but can also be used for external hard drive exclusively used with Windows OS. The Cluster size is 4KiB3. Information about each file, as the corresponding clusters, is stored in a file included in the MFT (Master File Table)4. NTFS arranges data smarter than FAT. It allocates buffer space, preventing that enlarging files will immediately cause fragmentation.

EXT

Extended filesystem is the default file system in Linux. Block Sizes can be 1KiB, 2KiB, 4KiB, 8KiB. Ext saves data in an even more intelligent way than NTFS. The data is scattered all over the disk to minimise the fragmentation caused by editing and enlarging files.

Conclusion

Talking about Windows OSś, FAT is more prone to fragmentation than NTFS. As we will see later, carving for files is not possible on fragmented disk images. This means it is more likely to successfully carve data from an image of NTFS than from an image of FAT.

But no matter what OS is being used, a lot of reading and writing will sooner or later cause fragmentation.

2.2.3 Simple File Carving

Header/ Footer Carving Hack 34

Header/Footer Carving is one of the easiest ways of file carving. With this method, we can identify the specific types of file headers and footers and then carve out the data between these headers and footers. This method is which is basically file-system independent, as file signatures and structures are the same, but it doesnt́ work in case the files:

- do not have a standard footer signature
- are fragmented
- beginning is no longer present

Note: The footer mode of Doc File is NEXT. This means Header + all data up to and excluding the footer.

The footer mode of the pdf file is REVERSE. This means Header + all data up to the last occurrence of footer within maximum file size.

Example:

We just focus on the process of carving an unfragmented .jpg file as this method does not work for fragmented files. In the example, Iḿ going to work with the 11-carve-fat.dd data which is a FAT32 system and available in public/Schaefter directory. We can recover some .jpeg files by using some commands in Linux and the header/footer method which I have already introduced above. In case you work with Windows you can use Cygwin to run Linux commands on your command line. **Step 1:** As we know any

F i l e Extension	Maximum file size in Bytes	Header	Footer
JPG	200000000	FF D8 E0 or FF D8 FF E1	FF D9
PNG	20000000	89 50 4E 47	49 45 4E 44 AE 42 60 82
PDF	5000000	25 50 44 46 2D 31 2E	25 25 45 4F 46
DOC	10000000	D0 CF 11 E0 A1 B1 1A E1	57 6F 72 64 2E 44 6F 63 75 6D 65 6E 74 2E
GIF	5000000	47 49 46 38 39 61 4E 01 53 00 C4	21 00 00 3B 00
ZIP	10000000	50 4B 03 04 14	50 4B 05 06 00
XLS		D0 CF 11 E0 A1 B1 1A E1	FE FF FF FF 00 00 00 00 00 00 00 00 57 00 6F 00 72 00 6B 00 62 00 6F 00 6F 00 6B 00
PPT		D0 CF 11 E0 A1 B1 1A E1	50 00 6F 00 77 00 65 00 72 00 50 00 6F 00 69 00 6E 00 74 00 20 00 44 00 6F 00 63 00 75 00 6D 00 65 00 6E 00 74
BMP		42 4D	
DOCX		50 4B 03 04	50 4B 05 06 (PK..) followed by 18 additional bytes at the end of t h e file
HTML	50000	<html>	</html>
PST	500000000	21 42 4E A5 6F B5 A6	

Figure 2.8: Table

JPEG file starts with the header ff d8 ff followed by either e0 or e1 and ends with the footer ffd9. E1 just means this is a JPEG image and E2 Digital camera JPG using Exchangeable Image File Format (EXIF). So, we can search for the JPEG headers and footers. We start with searching for the headers, with:

xxd 11-carve-fat.dd — grep ffd8

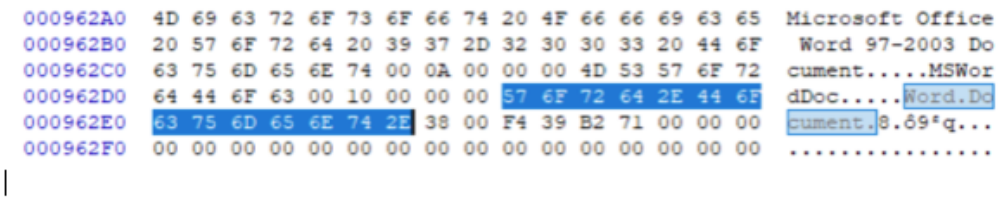


Figure 2.9: Footer

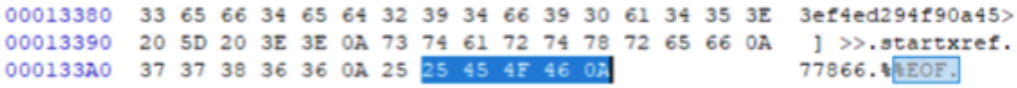
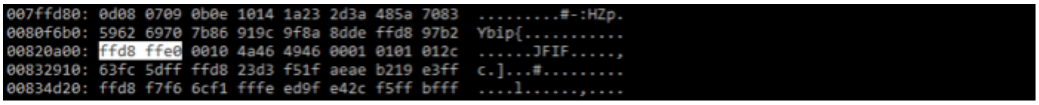


Figure 2.10: Footer

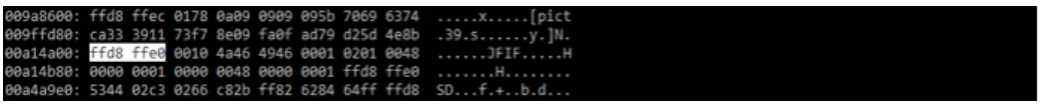
This will lead to the following result:



Over all we can find 4 possible .jpg file headers:

1. Picture: Header: 00820A00
2. Picture: Header: 009A0A00
3. Picture: Header: 00A14A00
4. Picture: Header: 009A8200

In this example Iḿ just going to show you how to recover the third jpeg file image.



Step 2: We need to convert the offset into a decimal number. There are two ways to do this:

Hint: the bc command only understands uppercase letters, so you need use Áínstead of á:

```
$ printf "%d\n" 0x00a14a00
10570240
```

```
$ echo "ibase=16;00A214A00 | bc
10570240
```

Important: We dont́ care about the offset at the beginning of the line. Thus, we should do a calculation to know the offset of the header ffd8. In this case, we should do: Header: 10570240 +0 =10570240 Same we will do for the footer but including the 2 bytes of the footer.

Step 3: Now we need to find the footers, skipping those before the header offset. We writer: *xxd -s 10570240 11-carve-fat.dd — grep ffd9*

```
00a165c0: ff00 ffd9 0038 4249 4d04 2100 0000 0000  .....8BIM.!.....
00a2cde0: ffd9 0000 0000 0000 0000 0000 0000 0000  ................
00a30940: 0580 8b99 d2c8 01d9 ba27 ffd9 a890 eb55  .........'.....U
00a9e5f0: 93f5 f6d5 feba 7ab5 1f59 50c8 3227 ffd9  ......z..YP.2'..
```

Converting the Offset to decimal and calculating the right offset (including ffd9), and we get 10669538.

Step4:

We need to calculate the file size, by simply subtracting the header offset from the footer offset.

```
$ echo " 10669538-10570240" | bc
99298
```

We get the resulting file size of 99298 bytes

Step5:

Now we can finally extract the .jpg file with the command dd (data duplicator). We simply need to tell the system, what the original file we want to carve from is, the header offset, and the number of bytes we want to extract.

```
$ dd if=11-carve-fat.dd of=shark.jpg skip=10570240 bs=1 count=99298
99298+0 Datensätze ein
99298+0 Datensätze aus
99298 Bytes (99 kB, 97 KiB) kopiert, 0,552542 s, 180 kB/s
```

Figure 2.11: dd

After all steps above there should be a new jpg-file in the current directory. Opening it we will see the picture of a shark.

Block - hashed carving Hack 35

Explanation Block hashed carving is a technique to detect the presence of a specific target file, even if the file has been fragmented, partially overwritten or slightly modified with the attempt of hiding. While file-based hashing calculates hash values of the whole file and cant́ detect the file, if only one bit has been modified, block-based hashing means creating hashes of data blocks and searching for similar blocks and e.g. recognising 9 out of 10 similar Blocks, 90 percent of the original file. Block-level hashing is the most basic scheme for determining the similarity of binary data. The technique generates and stores hashes for blocks of chosen size (usually the size of one block/cluster). Block hashes of two different files can be compared and by counting the number of matches found, a percentage of similarity can be calculated. The hashed sectors need to be aligned with the file system allocation blocks so that the sector hashes will then be aligned with the file block hashes. However, deletion or insertion at the beginning of the disk image file can lead to the different division of the blocks and thus, different calculated hashes leading to no matches. Another disadvantage is the increasing requirement of storage space due to fixed hash value lengths. For example, using the hash-calculation MD5 will always result in a hash value of 128 bit, even if only 1 byte has been put in. Unfortunately, this is no efficient method to search for evidence, due to its low evidential value. No matter how the hashes are being calculated, there is always a potential for hash value collision, especially for smaller block sizes. But even detecting all blocks, using larger block sizes only means that this file was once saved on that device, without the possibility of determining the time.

Example The following steps explained, are based on the blockhash.sh script described in the book Computer Forensik Hacks, Hack 35, p.104. We will be working on the file 12-carve-fat.dd and try to find the haxor2.bmp file (cf. http://dftt.sourceforge.net/test12/index.html), here called test.bmp, we just extracted, using the carving tool Autopsy 4.6.0

Step 1:

Starting the script, we need to provide the device image you want to search in, the target file you want to search for, and the block size. As we know, we are working on an ext2 file system, we will choose the block size of 1024.

Step 2:

Now the number of blocks in both files are calculated, by:

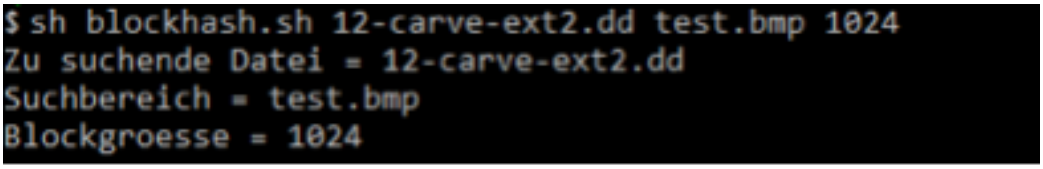


Figure 2.12: Script

let firstlength=firstsize/blocksize let secondlength=secondsize/blocksize

resulting in 126325 blocks for the disk image file and 160 blocks for the .bmp file

Step 3: These lengths are used to iterate over each block, calculating the MD5 hash values and storing them in two files. Here, the example of the first file for-loop:

for ((i=0; i ¡ ((firstlength)); i++)); do dd if = first bs = block size skip=i count =1 md5sum awk print 1 ¿¿ first.hashes

Step 4: Both files, including the hash values, now can be compared. We count the matches in one file and calculating the result in a percentage by deviding it by the length of the other file:

hits1=(grep−c−ffirst.hashes second.hashes) firstpercent=(echo"scale = 2; (hits1/secondlength ∗ 100)"|bc − l)

So, we will get following output:

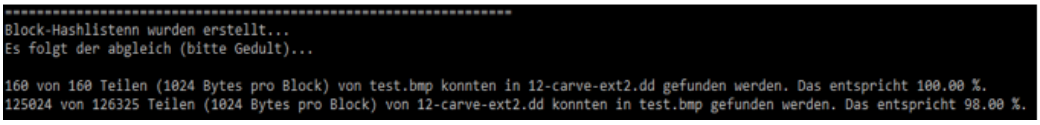


Figure 2.13: Hash Values

We were able to find all of the .bmp files 160 hashes in our image file, which indicates that the .bmp file was saved on the hard drive at some time. As we see we have found 100 percent of the bmp file in the image file. The other way around, we even found 125042 hashes of the image file in the .bmp file. It is a little irritating that we have found 98 percent of our image file in the .bmp file even though the image file is much bigger. This is a good example of data collision. Examining the image hash file, we can see that there are 124864 hashes of blocks of 0s that are included in the .bmp file hashes twice. Thatś why these hashes are matched with the same .bmp

hashes repeatedly. Adding those 124864 to the 160 hashes of the .bmp file we get the resulting 125024 matches.

To give it another try, we can now change the .bmp file within the drive image file, by changing one single byte, using a hex editor. This could happen due to partial overwriting or be an attempt to hide the file. Doing the same procedure again will now lead to the following result:

Figure 2.14: Hex Editor

In this case, we were able to find 159 matches, 99 percent of the .bmp file in the image file. Only one Block including the modified byte cant́ be matched. Still, we get a clear idea, of how possible it is, that this file was once stored on the hard drive.

2.2.4 Glossary

1. **Cygwin :** is a Linux-like environment for Windows providing the UNIX API to run UNIX commands on the Windows command line11. Installing Cygwin gives access to many standard UNIX utilities that can be used from either the provided shell or the windows command prompt.

2. **Autopsy :** is a graphical interface to the Sleuth Kit, a collection of command line tools to scan disk images. It is used to simplify the analyses of UNIX and Windows disks and file systems. It can analyse disk images of either raw/dd or E1 format, local drives or a folder of local files. Features of Autopsy are timeline analysis, web artifacts, email analysis and many more. The feature we used to carve the ext2 drive image for a .jpg file is file type sorting, which is finding and grouping all images and documents. Moreover, Autopsy can create some types of reports about these procedures, that can be used for investigation.

3. **Hex editor :** A hex editor is used to view and edit binary files. Hex editors can visualise raw data of files, unlike other programs, that in-

terpret the data for you. Typically, there are three view areas in a hex editor: the address area, the hexadecimal area and the character area. The Process of editing a file using a hex editor is called hex editing.

2.3 Internet Artifacts with Mozilla Firefox

2.3.1 Introduction

Internet artifacts refer to the data saved by the browser on the userś history. They are important for digital forensics, because they can be used to make a timeline of the userś events on the browser and the Internet in general. In the case of Mozilla Firefox the history data is stored in SQLite files. SQLite files are used to store databases. These files can be found in the Firefox Profile folder. The path to the folder depends on the operating system:

Operating System	Location
Windows XP	C:\Documents and Settings\%username\Local Settings\ Application Data\Mozilla\Firefox\Profiles
Windows Vista/7/10	C:\Users\%username%\AppData\Roaming\Mozilla\Firefox\ Profiles
Linux	/home/$username/.mozilla/firefox/Profiles
OS X	/Users/$username/Library/Application Support/Firefox/Profiles/

Figure 2.15: Table

This folder can also be directly accessed through the Firefox browser itself. Simply type ábout:profilesínto the search bar and click the Śhow in Finderbútton. The SQLite files could be open with two kinds of programs. First, a command line tool called SQLite3 can be used. This command line tool is available for Mac, Linux, and Windows. It can be downloaded here: https://sqlite.org/download.html. To open a database using this command-line tool simply type śqlite3 full file path to the database file. If the śqlite3¿prompt is seen, then the database has been opened. The .tablesćommand can be used to see all the tables in the database, .schemaćan reveal all the fields in the database, and .helpćan be used to get a full list of the available commands. However, in order to get information required from the database, an SQL query will be needed. [Alt11]

The SQLite files could also be read by a program using a GUI interface. One such program is called sqliteman. It is available for Windows and Linux at:https://sourceforge.net/. A sqlite database is opened using Ctrl-O (or going File Ópen) in this program. The same SQL queries used in sqlite3

must be used in sqliteman. Unfortunately, a version of this program is not available for Mac users.

Formhistory

First, the Firefox profile has the formhsitoryṡqlite database. This database contains all the data that the user has entered into forms. This could include usernames, emails, addresses, search queries, etc. It would not include passwords as those are stored somewhere else. After opening a database, an SQL query is needed to get information. In this case run the command:

The command should retrieve information from the $moz\_formhistory$ table, this is why it is named in the FORM section. After SELECT all the fields that are wanted have to be named.

ID: This is the number used by the database to differentiate entries. Since a single entry could be very long an id number will help tell when an entry stops and when the next one starts. This number should be sequential.

Fieldname: This defines the type of data being stored in the entry. Possibilities include username, email, and search bar history.

Value: This is what the user actually typed into the form.

timesUsed: The number of times the user has used this entry.

firstUsed: The time the user first used the entry. All time related entries in these database are stored in someting called PRTime. UNIX Epoch time is the number of seconds that have passed since January 1st, 1970. PRTime is the UNIX time in microseconds. So this query will first convert PRTime unto UNIX time and them convert this UNIX time into the more readable Year-Month-Day Hour-Minute-Second format. This is done with the date time function. [Alt11].

lastUsed: The last time the user used this entry. It is converted using the date time function.

From the above example, a digital forensics specialist could find out that, that in a form field for either an username or an email the user entered the email ábbasbushehri2@gmail.comónce on April 7th, 2018 at 13:46:34.

Downloads

The downloads.sqlite database has information on files downloaded using Firefox. This includes the name of the file, where it was downloaded from, where it was downloaded to, when the download started, and when it ended.

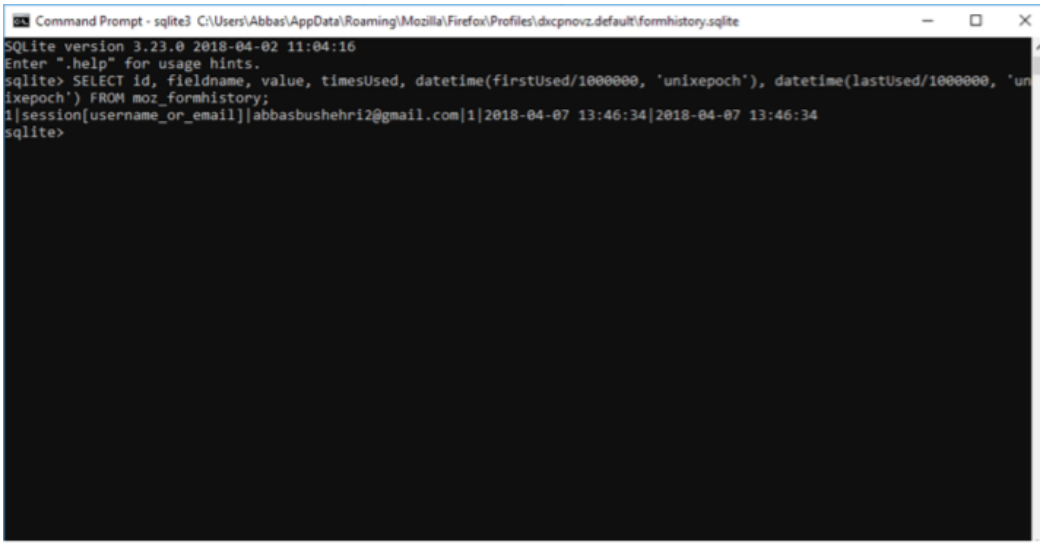


Figure 2.16: Last Used

This database does not contain information on files that were downloaded through an add-on or the cache, only files handled directly by the Firefox download manager. Use the following query:

The query is used to gain the following information from the *moz_downloads* table:

ID: The number used by the database to differentiate entries. This number should be sequential.

Name: The name of the file downloaded.

Source: The full URL or filename of the fileś origin.

Target: The full file path of where the file was saved.

tempPath: If a file was opened instead of downloaded then this entry will show where the file was temporarily stored. This typically happens when opening large PDF files.

startTime: The time when the download was started. This time is converted using the date time function. If the download was canceled and then restarted, then this value will be updated.

endTime: The time when the download was completed, paused, or canceled. It is converted with the date time function. A restarted download will update this value.

State: If the value is zero then the download is in progress. If itś 1 then

the download is complete. If itś 3 then the download stopped. If itś 4 then itś paused.

Referrer: The URL that directed the browser to the downloaded file.

currBytes: Current number of bytes that have been downloaded.

maxBytes: The size of the source file in bytes.

prefferedApplication: Application that was used to open the file. This is typically Firefox unless the user decides to open the file and selects an application to open it with.

Cookies

The cookies.sqlite database stores all the Firefox cookies. The cookies could tell when was the last time a user visited a site, whether they were logged in or not, and whether the site set or requested the cookie. Use the following query:

The query gets the following information from the *moz_cookies* table:

ID: The sequential number used by the database to differentiate entries.

baseDomain: The base URL value. For example, with the URL ẃebmail.hs-ulm.de, ́hs-ulm.dewould be the baseDomain value.

Host: The full name of the website.

Name: The name of the cookie. This field can sometimes tell what information the cookie is holding.

Value: The value of the cookie. This field is usually in a hash format and not readable. However, some cookies are readable.

creationTime: When the cookie was first made. It is converted using the date time function.

lastAccessed: When the cookie was last used. It is converted using the date time function.

Expiry: When the cookie deletes itself. It is converted using the date time function. The fact that a cookie expires is another reason why a digital forensic specialist should only work with copies of the original data, as a copy could be changed over time.

In the above example, it can be seen that the 523rd entry in the *moz_cookies* database is a cookie from the newnotcenter.com domain. The cookie is for remembering whether or not the Firefox search extension is enabled according to the **Name**. The **Value** says that it is set to true. The cookie was only accessed once as the **creationTime** and **lastAccessed** are the same. Finally, the cookie has no expiry date.

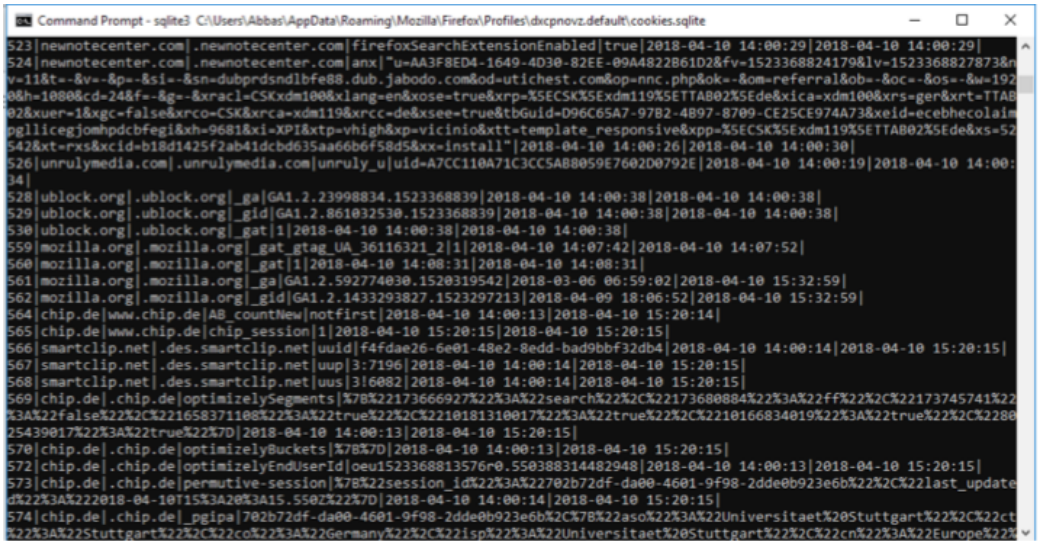


Figure 2.17: Cookies

Places

The places.sqlite database has the most information related to the userś activity. It contains all the websites the user visited along with the time they visited them. The sites are stored in the *moz_places* table while the time is stored in the *moz_historyvisits* table. So the query will have to match the entries in one table to the other one. Also, the time is recorded in PRTime once again so the date time function will be used. The following query will not only list the sites with the time, but also convert the time into a readable format:

This is what a complete places timeline would look like.

Cache

The Firefox cache stores the images, scripts, and other parts of a website that has been visited. So if the same website is opened then it will load faster. Cache is not stored in a SQLite file. In fact, it may not be stored in the usual Firefox profile. Instead, itś stored in:

Another program has to be used in order to read the cahce. In the case of Windows, download MozillaCacheView from https://$www.nirsoft.net/utils/mozilla_cache_viewer.h$ When this program is opened it will automatically read the current contents

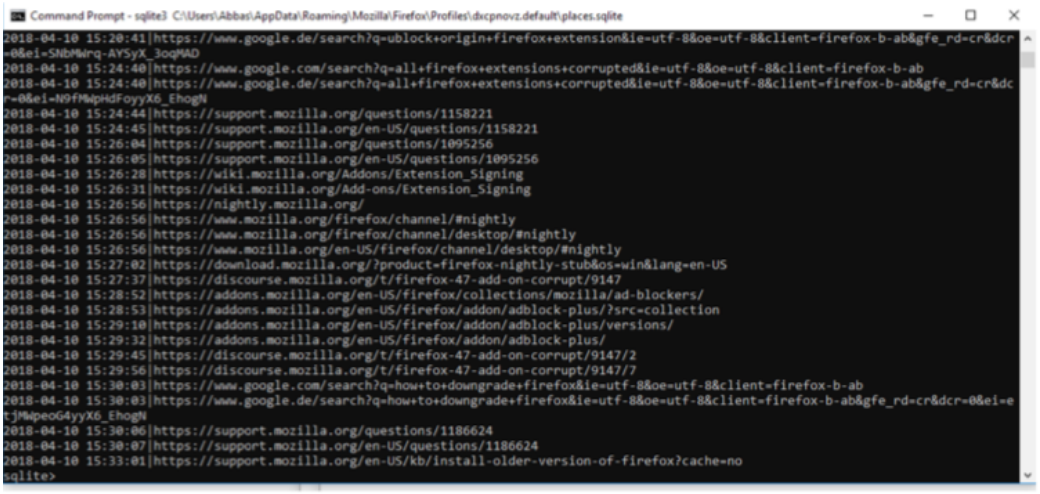


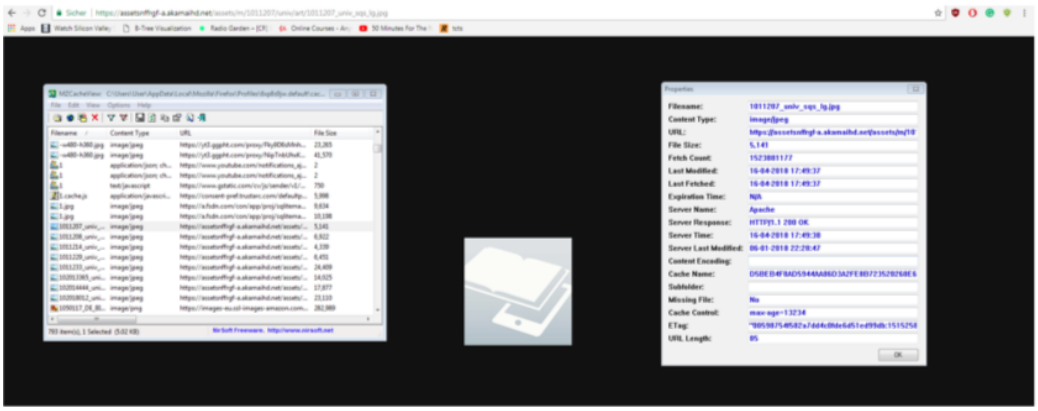
Figure 2.18: Places

Operating System	Location
Windows XP	C:\Documents and Settings\%username\Local Settings\ Application Data\Mozilla\Firefox\Profiles
Windows Vista/7	C:\Users\%username%\AppData\Roaming\Mozilla\Firefox\ Profiles
Linux	/home/$username/.mozilla/firefox/Profiles
OS X	/Users/$username/Library/Caches/Firefox/Profiles/
Windows 10	C:\Users\%username%\AppData\Local\Mozilla\Firefox\Profiles

Figure 2.19: Cache Stored in

of the Firefox cache. Since a digital forensics specialist should only work with a copy of the profile instead of the original, make sure to load the cache directly from the folder of the copy.

In the above example, one of the images from the cache is opened. In the imageś URL is in the data. The URL is then searched to get the original image. The original image is showed in the centre. If a website was modified or removed, then the cache would be the proof of what it contained.



Saved Session Data

Whenever the Firefox browser is closed then a file called sessionstorejs is created. This file stores information on all the closed tabs and windows of the browser. When the browser is reopened it will read the file in order to restore the tabs and windows, before then removing the file. Sessionstorejs is a JSON file. While they can be viewed with any text editor they would be hard to understand. So a program like Json View, which can be downloaded here:

http://www.softpedia.com/get/Programming/File-Editors/JSON-Viewer-Mitec.shtmlm

The above example shows how the URLs from the closed tabs could be retrieved.

Bookmarks

Firefox stores the bookmarks data in the places.sqlite database. The relevant data is stored in two different databases moz_places and moz_bookmarks. Using the SQL query:

moz_bookmarks.title: The name the bookmark was saved as. moz_bookmarks.dateAdded: The time the bookmark was made. It is converted using the date time function.

moz_bookmarks.lastModified: The time the bookmark was last changed. It is converted using the date time function.

moz_places.url: The URL link to the bookmark.

moz_places.title: The name of the website the bookmark links to. If the

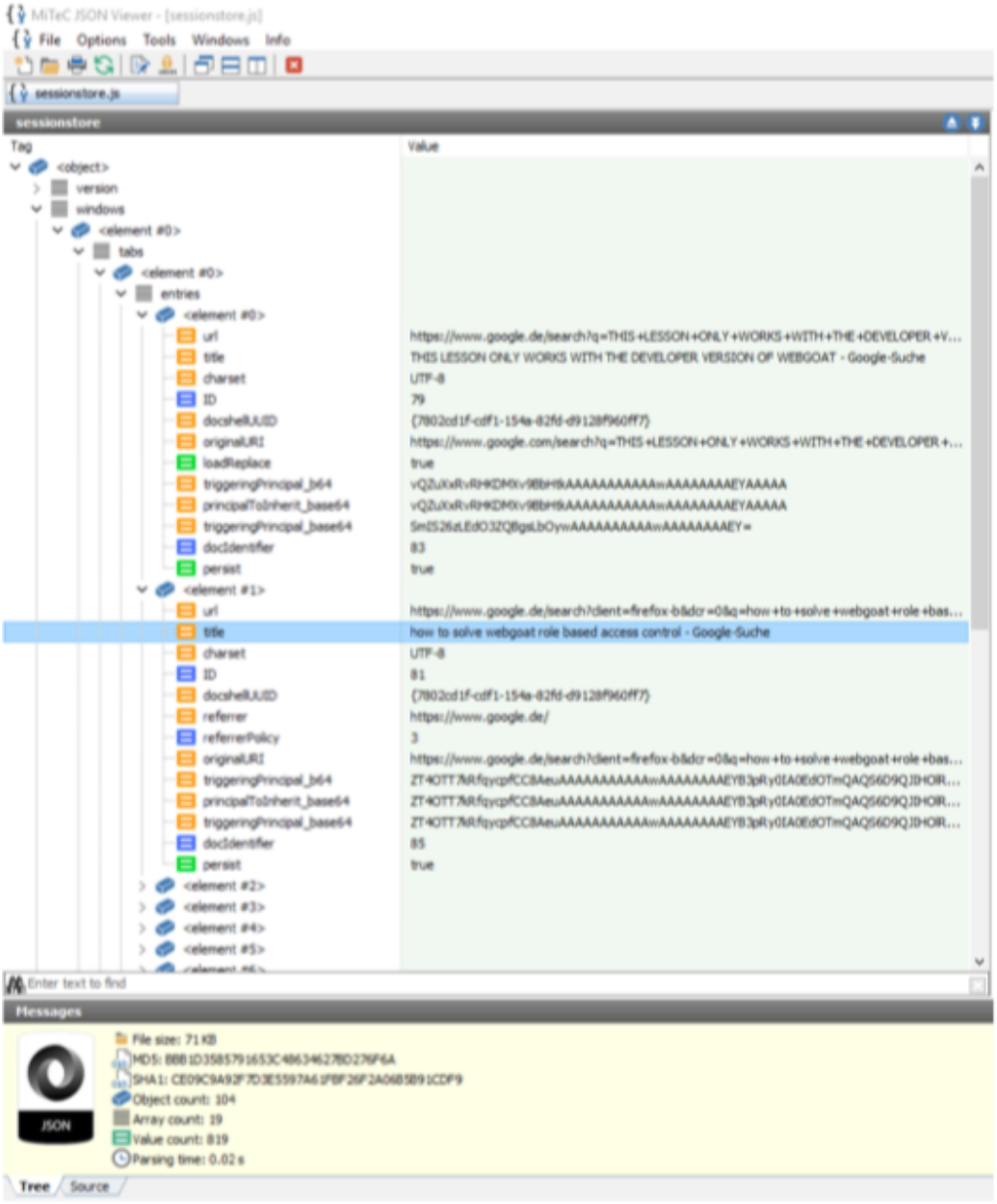


Figure 2.20: JsonView

user doesnt́ specify a name for the bookmark then this field is equal to the *moz_bookmarks.title*.

moz_places.visit_count: The number of times the bookmark was used.

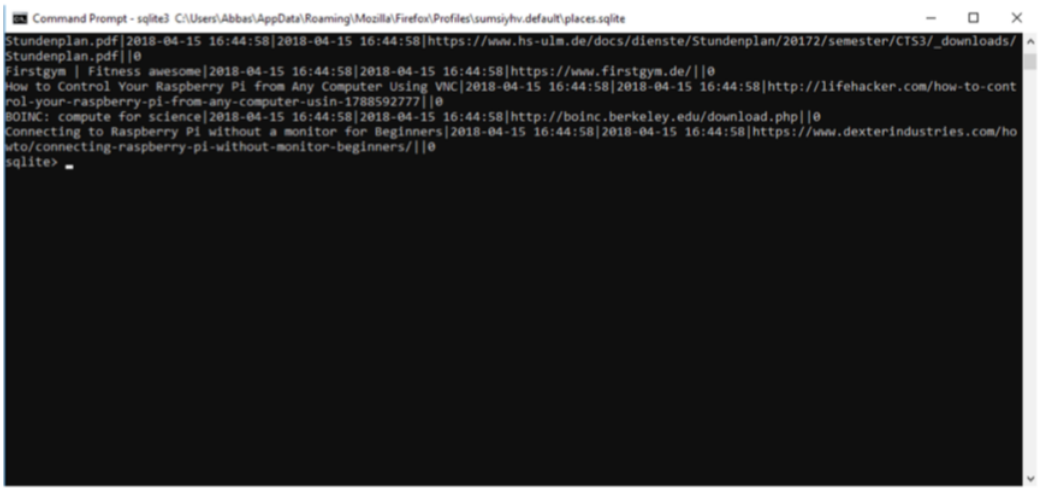


Figure 2.21: Bookmarks

Extensions

Firefox stores the extensions data in a JSON file called extensions.json. It will contain data on what extensions the user has, when they were downloaded, and whether they are enabled.

In the following picture, it can be seen that the web extension uBlock is installed. It could also be found that the extension is enabled.

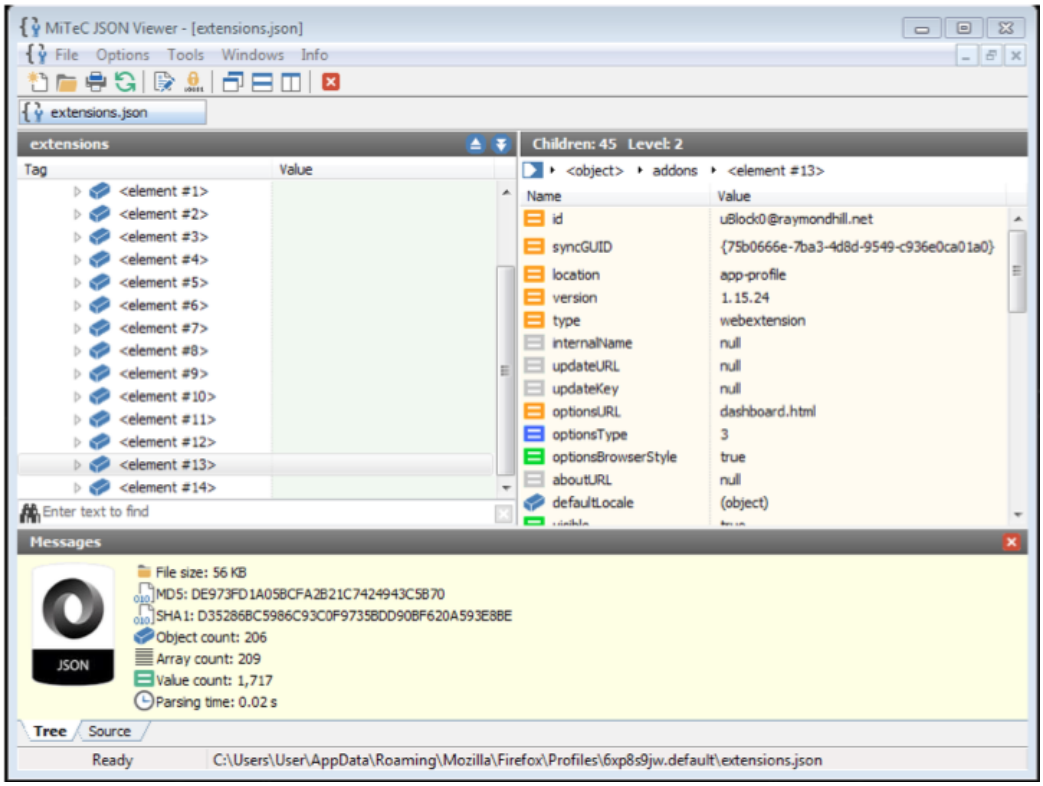


Figure 2.22: Extensions

2.4 Email Forensics

2.4.1 Introduction

Many cyber crimes have involved the use of emails, either as the means or the evidence of the crime. Emails could contain evidence of many types of crimes such as:

- Domestic violence
- Cyber-harassment
- Extortion
- Embezzlement
- Fraud

- Identity theft
- Child exploitation and abuse
- Terrorism
- Drug dealing
- Gambling
- Intellectual property theft
- Organised crime

2.4.2 Basics of email

There are two types of email systems:

- Client/Server Email:The client sends or receives the emails.The server stores the messages until the user retrieves them. In this system, the emails are downloaded onto the userś computer.
- Web-based Email:The email account has to be accessed through a Web browser. The emails are stored in the email service providerś server.

Email systems use a variety of protocols:

- SMTP(Simple Mail Transfer Protocol):It is part of the TCP/IP, which is the primary protocol for sending messages through the Internet. SMTP is responsible for sending emails over either a network or the Internet.
- POP3(Post Office Protocol 3):POP3 is used to read the email.It stores the emails in a single folder until the user downloads them. After an email has been downloaded, it is deleted from the server by POP3. However, a user could choose to keep the emails on the server after downloading for a period of time. This means the investigators should not ignore this server even if the user already downloaded the email.

- IMAP (Internet Message Access Protocol):IMAP is used to read to retrieve and read emails, like POP3. Unlike POP3, IMAP gives the user the option to store emails on various folders on the server. POP3 is still more widely used than IMAP.

An email is made up of a domain name and a username. The domain name is everything after the @ symbol, while the username is everything before it. For example, in the email éxample@yahoo.com; éxampleís the username while ýahoo.comís the domain name.

2.4.3 Parts of an email

An email contains a body and a header. The body is the actual content of the message. The header is either in the condensed version or the full version. The type of header field used will determine the data the investigator can retrieve.

Condensed Header:

- **From:** This field contains the senderś email address, which could be faked. It could also contain the senderś name, which could also be faked. The reason they could both be faked is because SMTP does not verify email headers.
- **To:** This contains the receiverś address, and possibly their name. Once again this could have been spoofed.
- **Subject:** This field could be blank. It could also contain misleading information.
- **Date:** This includes the date, day of the week, time, and time zone. This field is recorded by the senderś computerś clock. However, it is not accurate if the senderś clock is not set correctly.

Full Header:

- **X-Originating-IP:** This field reveals the senderś IP address. An IP address can either be static or dynamic. A static IP address is a permanent address for a specific computer. A dynamic IP address is used by a computer for a period of time. When the time is up, the IP address is placed back into a pool of available iP addresses, where any computer could end up taking it.

- **Received:** This field is in the format ́from[IP address] by [server name] with [Internet protocol], day of the week (first three letters), date [format: day month (first three letters) year], at [time (format is hour:minute:seconds)] time zone. Some email systems do not include the IP address of the sender. Also, emails can contain more than one Received field if the email goes through several servers. This is because a server is responsible for creating this field. Multiple Received fields can reveal whether or not the senderś IP address is faked. The investigator just has to check if the location next to the word ́byís the same as the location next to the word ́fromín the Received field below it. If they do not match, then the senderś IP address has been faked.

- **Return-Path:** This is where the email should be returned to if it could not reach its destination. If this does not match the address in the From field, then the sender faked their address.

- **MessageID:** This contains the name of the server and a unique string that the server assigned to the message. This string can be used to track the message.

- **Received-SPF:** The receiver of an email puts the email through an SPF query. This query checks if the sending server is allowed to send an email to the receiverś domain. If the result is ́failt́he message is rejected. If the result is ńeutralór ṕasst́hen another spam filter decides if the message should be counted as spam.

- **Authentication-Results:** This field makes a recommendation to the user on the validity of the messageś origin and content.

- **Content-Type:** This indicates the type of data in the message, such as text, audio, video, or images.

- **X-Mailer:** This specifies the email system used to send the message. Examples include Microsoft Outlook and Verizon Webmail.

2.4.4 How to conduct an email investigation

The first step is obtaining the email. The computer forensics investigator should first make a copy of the digital evidence. The copy should of course include the full header and any attachments. Keep in mind that even if

the receiver deletes the email, it can still be possibly found in the sendeŕs computer. Even if it́s deleted there, it can be found in the backup tape of a network server, or in a computeŕs temporary files, or in a computeŕs unallocated space.

The emaiĺs body and header should be searched for evidence. The investigator should also check attachments, people who have received copies of the email as secondary recipients, and people to whom the message was forwarded. In the header, the most important information is the IP address. The PING command can be used to check if the IP address is accessible (ie: ṕing 72.14.204.444́).

A query tool known as WHOIS allows a computer forensics investigator to find out the contact and location information of the owner of an IP address. To use WHOIS, the investigator should type the IP address retrieved from the email into this query tool, and the tool then retrieves information about the ISP. A static IP address is easy to trace back to the computer. To trace a dynamic IP address, the investigator should also provide the date and time the criminal used the IP address. Domain names can also be used for a WHOIS search. However, in this case the investigator should know exactly what they are looking for, because many different results could come from this query. [Mar15]

2.4.5 Problems encountered by investigators

- **ProxyServer:** Criminals may use proxy servers to hide or mask their IP addresses. If someone uses a proxy server, that useŕs identity is not revealed because the proxy server gives its own identity instead.

- **Tor:** It is a communication system that allows people to communicate without losing their privacy.Instead of the message taking a direct route, the data packets on the Tor network take a random pathway through several relays that cover your tracks so no observer at a single point can tell where the data came from or where it is going.

- **Avoidance:** With this technique, a useŕs actions are displaced to times and places where the surveillance is assumed to be absent. For example, Al-Qaeda used this technique to distribute its propaganda videos. Websites and message boards were used to distribute these videos. Different websites uploaded the videos, then the videos would remove themselves

after a period of time, and then they would be uploaded on different websites. This technique made it hard for authorities to track where the videos were being uploaded form.

- **Piggybacking:** When there is surveillance, the information that needs to pass through undetected can be attached to a legitimate object. One way this can be accomplished is with steganography where the data could be hidden inside a sound or image file. In this case, only someone with the appropriate software can see the hidden message.

- **Blocking move:** This is when the individual either blocks access to the communication or renders parts of it unusable. For example, encryption is considered a blocking move. This is because only the cipher text is sent over the communication channel, and it is not usable to a third party. The intended recipient should have the decryption key.

- **Pizzini :** Small slips of paper, either hand written or type written, that are used for communication in order to avoid the surveillance of telecommunications and electronic communications.

2.5 John the Ripper - Breaking passwords

2.5.1 How passwords are stored?

The first important thing for cracking passwords, is to know how they are stored. Normally passwords are never stored in their clear format, because when attackers get access to the server or database where the passwords are stored they had it immediately. To get a higher security the passwords are stored as hash codes. The hash code is like an encryption for the password. These codes get calculated with a complicated cryptographic algorithm depending on which encryption/hash algorithm is used. Popular hash algorithms are for example SHA-1, MD5, SHA256/512. The advantage is that all these encryptions are like a one-way encryption, this means you can calculate the hash code of a string but not the string from the hash code. When you try to login with a password the system generates the hash code of the string and compare it with the hash code from the database for example.

Password Salts

Because the hardware in our computers becomes faster and faster the method to store passwords as hashes gets less secure. To increase the security of passwords people are using password salts in combination with hashes. A salt is a random generated string which gets attached to the normal password. This means that it will be saved as a different hash than the normal password. The random string will be different all time, even if the two passwords are exactly the same.

	Password	Hashed Value
No Salt	this1sAg00dPASSword!!	a5a5baa0c16166260e9ef8a48dbde112
Salted	6789o3uigtbgeat7this1sAg00dPASSword!!	53cffc58904a10b9dcc40345433862dc
Salted	v8734ihv6!nre432this1sAg00dPASSword!!	28b8f782262a890b4d730f8001d23bd5
No Salt	love	b5c0b187fe309af0f4d35982fd961d7e
Salted	12bg55tygsdf4gvi9yrdslove	65c96e15930d34dd9a9ce916b81fb044
Salted	879rughq2ebt5dfxcasedlove	a35436c0e0f2821db2703c1983a641ab

Figure 2.23: Password Types

2.5.2 Algorithms behind passwords

In the recent years password security is becoming more and more important. As the hardware in our computers becomes increasingly more powerful it also becomes increasingly easier to guess passwords by using some forms of brute force methods(brute forcing = trying all combinations which are possible). When looking at the industry we can see that passwords are usually can be stored either hashed or in plain text like described in the first part.

So now that we know how the passwords are usually stored we have to look at some math to determine how weak or strong our passwords are and what is possible when we talk about guessing passwords with software.

These days any computer equipped with the right software tools can easily guess billions of passwords per second and super computers that are used by the NSA can guess more than a quintillion passwords a second.

From mathematics we know that the formula for computing all possible passwords with a certain number of characters is: (number of characters in the alphabet)$\hat{}$(password length) so a simple password like ́helloẃould fall into a category where we have 26 characters in the alphabet and a length of 5. This means that there are $26\hat{5} = 11\ 881\ 376$ possible combinations for such types of passwords. A person trying to crack such easy passwords with a computer that can guess around 50 million passwords per second would only need around 2 seconds to crack the password. If we, however, consider passwords that use the lower and upper case characters, numbers and some special characters we end up with an alphabet that consists of 80 characters. If the password is 10 characters long then the person trying to guess the password by the means of brute force would need to wait for around 6800 years. At this point we can be happy that our passwords are secure but when we consider the fastest supercomputers that can try quintillions of passwords per second even this strong password becomes available to easily crack and this password would be cracked in around 10 seconds (if we consider a speed of 1 quintillion guesses per second).

These kinds of computing speeds are still not available to the general public so thatś why we have to choose different methods of trying to crack passwords.

2.5.3 How John the Ripper works?

John the Ripper uses three different options to crack a password. These are the śingle crackḿode, the ẃordlistḿode and the íncrementalḿode. By trying to crack the password with the f́ormatt́ool, john the ripper is trying all these three options. First the śingle crackḿode than the ẃordlistḿode and when this does not work - the íncrementalḿode. Itś also possible to start the modes manually.

How the modes work in detail:

Single Crack Mode

Śingle crackḿode: This is the fastest mode and the mode with which john the ripper is started. By using the śingle crackḿode john the ripper is trying all strings which are saved on the system. For example, user names or names of directories.

Word list mode

wordlist mode: The wordlist mode is the most important mode. In this mode john is using a list which includes words/strings and tries these words. John is calculating hash codes from these strings then compares them with the hashes of the password. Because passwords are normally saved as hash codes and itś not possible to calculate the string from the hash code John has to do this. This increases the chance of guessing the password since most people only use non-unique passwords like ṕassword; 1́234567890; etc..

Incremental mode

incremental mode: In the incremental mode john the ripper is trying to crack the password with the help of brute-forcing. This means trying all possible combinations of the password. It is the slowest mode and that is the reason why john the ripper uses it as a last resort option. The problem is that it is theoretically possible to crack all passwords but in virtue of the slowness of the process it is practically not always possible to crack a strong password because it would take years to crack it. It is possible to set some filters, for example, only lower case letter and numbers, or something else. For setting a filter you have to write: john –incremental = your preferred filter password hash.

Following filters are available: digits(numbers(10)), upper(uppercase letters(26)), lower(lowercase letters(26)), lowerspace(lowercase letters + space(27)), uppernum(uppercase letters + digits(36)), lowernum(lowercase letters + digits(36)), alpha(lower + uppercase letters(52)), alnum(lower + uppercase letters + digits(62)), alnumspace(lower + uppercase letters + digits + space(63)), ascii(all printable ascii characters(95)), lm_ascii(for LM hashes)

2.5.4 Installation

In this tutorial we are going to use a pre - compiled version of john the ripper to make the experience easier for others who want to try this hack and it is okay to do so as we are only trying to use this for learning purposes.

We use a unix based machine in this example and we start by downloading a compiled version of John the Ripper:

http://openwall.info/wiki/john/custom-builds

After downloading and unzipping the file we have a directory with all the files of John the ripper. In terminal we navigate to the folder with cd and then do:

cd run

Now we are in the run folder and with
'ls
' we can see from which file formats john the ripper can pick out the hash code:

2.5.5 Getting access to locked pdf

To crack a password of a pdf we are going to use pdf2john file. We have a password protected pdf on the desktop called x.pdf. We first need to generate a hash for this file by running:

Now we have a file called output.txt on our desktop and itś contents looks like this:

*/Users/kgudzius/Desktop/x.pdf: pdf 4*4*128*-1084*1*16*7e0b1cd3e44b11475d1ed17d1fe9aaed*
*6afcd4730bf56564000000000000000000000000000000*32*c82b70f2fb3fd5ab153e814*
497c7655594abb457cfca41f61df4a3f0da9fe060

Now all we have to do is to find the password by running the command:

And we have the password of the pdf in front of us :

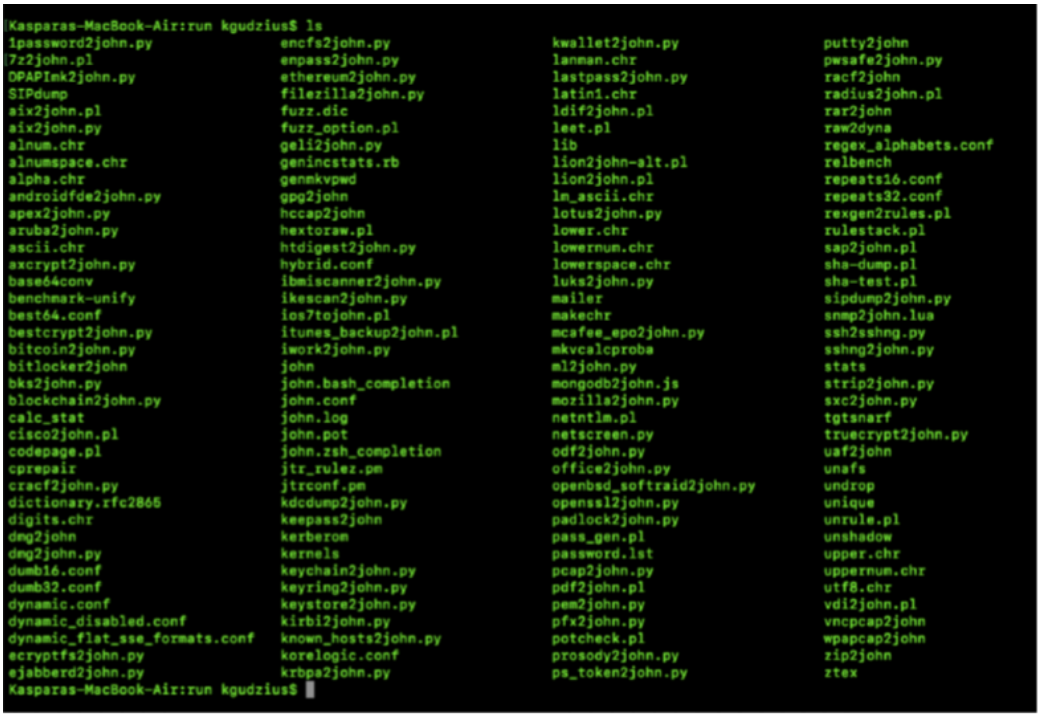


Figure 2.24: Files

```
./pdf2john.pl (directory of the file to be cracked) > (output file)
```

our example:

```
./pdf2john.pl /Users/kgudzius/Desktop/x.pdf > /Users/kgudzius/Desktop/
output.txt
```

Figure 2.25: Opening File

2.5.6 Cracking system passwords

In this example we are going to crack the system passwords using the tools that John the Ripper provides us with. This could be any other hashed password list but we are going to use the system passwords as an example that everyone can try out themselves. First we need to un shadow the system password file:

unshadow /etc/passwd /etc/shadow /root/Desktop/mypassword.txt

```
./john --format=pdf (output file)
```

our example:

Figure 2.26: Output file

```
./john --format=pdf /Users/kgudzius/Desktop/output.txt
```

Figure 2.27: Command

```
Kasparas-MacBook-Air:run kgudzius$ ./john --format=pdf /Users/kgudzius/Desktop/output.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (/Users/kgudzius/Desktop/x.pdf)
1g 0:00:00:00 DONE 2/3 (2018-04-11 15:13) 1.515g/s 62200p/s 62200c/s 62200C/s 123456..franklin
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 2.28: Password Cracked

Then we have to decrypt the hash code file. We are going to do this by using only the wordlist mode.

```
john --wordlist=/usr/share/wordlist/sqlmap.txt /root/Desktop/
                                    mypassword.txt
```

Figure 2.29: Wordlist mode

That is the output of this command. We can see the decrypted password. The command finished in about 15 minutes.

We can see the cracked password - "hello" !

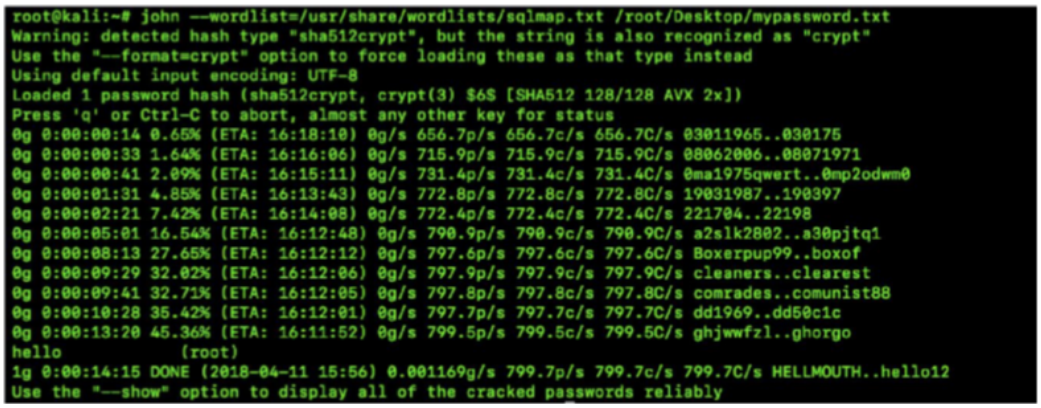


Figure 2.30: Cracked Password

2.6 Remote Imaging with net cat

2.6.1 Objective

This hack aims to access and transfer data (e.g. hard drive disk images) remotely via network. The main focus of this procedure is to configure a device to send information like disk images or backups to another machine. This mechanism is especially important to obtain data for analysing purposes from a storage system where a physical access is not possible.

Basically there are two different approaches. This first hack (12) uses a simple TCP/UDP protocol established via netcat. It must be taken into account that these connections are not encrypted and thus not safe. In contrast to this procedure, Hack 13 deals with a secured transmission by setting up a SSH connection.

2.6.2 Netcat

Netcat is a simple network tool to read and write data across a network using TCP and UDP. Itś also referred as the Świss Army knifećause of itś wide range of functionalities. Some of these includes transferring files, port scanning, establish back doors and port listening. This Hack describes the basics of Netcat and how data can be transferred and backed up over a networks.

Installation on Windows and Linux

Linux: On many Linux-Distributions Netcat is already preinstalled and can be used with the path-variable ńc. Should this not be the case, it can be installed on debian systems with the package manager using the following command:

sudo apt-get install netcat-openbsd

Windows: To run Netcat on a Windows system and use it without installing itś possible to download a portable version in the link below. Instead of the ńcćommand on Windows, the tool can be used by replacing it with ńcatėxe directly from the folder.

Downloadlink: http://nmap.org/dist/ncat-portable-5.59BETA1.zip

2.6.3 Banner grabbing for OS-fingerprinting

Banner grabbing is a technique to gather information about the operating system, the brand and the version of a service or application. Many services identify themselves with a so-called Banner when establishing a connection. This Banner contains information about the host and its service.

If an attacker knows which operating system and which software and services are used, he can search for known vulnerabilities to exploit them. Therefore, it makes sense for an attacker to pretend a connection request to get this information. On the other side this allows administrators and forensics to retrace what applications are running on the server and how an attacker gained accessed to a system. Also it allows to stay aware about the versions of the running software on the servers.

Acquiring the header

To grab this Information itś already enough to send a simple HTTP-Request to the Target-Host. First, we have to establish a connection to the target host with the specified port. This can be easily done with the following command:

nc host port

For our example we want to connect with the HS-Ulm web server on the HTTP-Port 80.

nc www.hs-ulm.de 80

The next step is to specify the type of the request. If we only want the Header, we had to type ***HEAD / http/1.0*** followed by two carriage returns.

As result, we will receive all the Information that the Header contains as seen in the following illustration.

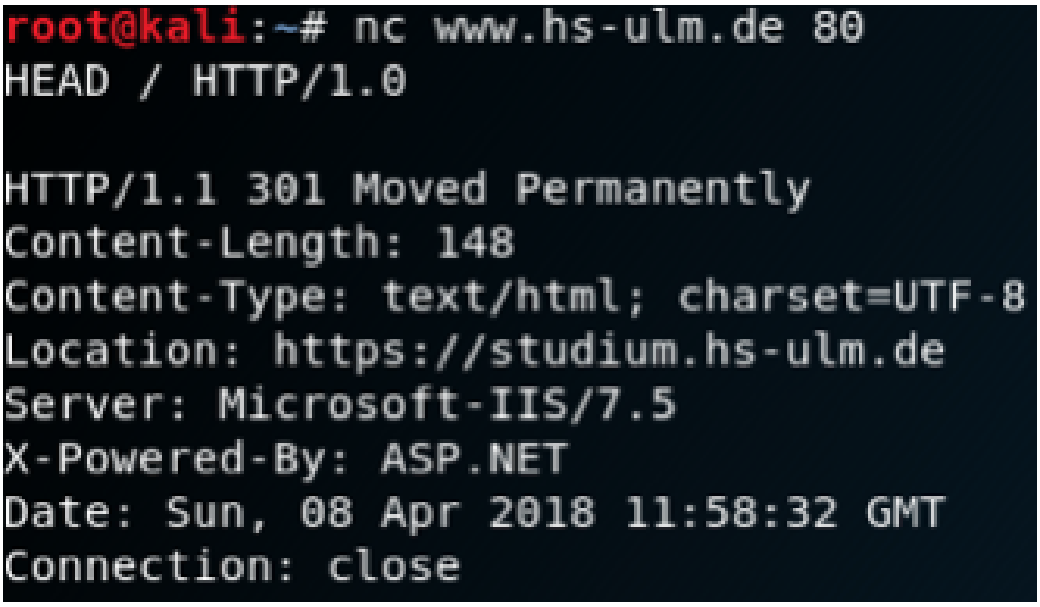


Figure 2.31: nc

Analyse the response

If we evaluate this information and looking for the Server line, we can see that the web server is running an application with Microsoftś IIS version 7.5. This information is already enough to take a quick search in a vulnerability database to find some vulnerabilities.

Furthermore, we can see that the server is also running an APS.NET extension. In some cases, we could also get some more information, but on purpose to prevent an attack, some of these entries can be deactivated.

2.6.4 Open a remote shell

The functions of Netcat are not only limited to Information Gathering. It also allows to get access to a command shell, execute a program of your choice or install a persistant backdoor. However, the establishment of a permanent backdoor for forensic investigations should only be taken with care, because these backdoors could also be used by an attacker.

Instructions

To set up a connection to a windows client via telnet over port 23, we can run the following command on the client:

Ncat.exe ĺ ṕ 23 t́ é cmd.exe

Parameter	Abbreviation	Description
--listen	-l	Bind and listen for incoming connections
--source-port	-p	Specify source port to use
--telnet	-t	Answer Telnet negotiations
--exec	-e	Executes the given command

Figure 2.32: Info Table

This command opens a local command shell on a Windows-Client, which can be used remotely to execute commands on the target machine. Otherwise, on Linux-Clients we had to replace cmd.exe with /bin/bash.

By running the command below we can access the target machine from any client in the network. Itś only necessary to replace the target-ip with IP-address of the host. For the connection thereś no authentication or anything similar required.

telnet target-ip

Once we have established the connection we are able to control the client and execute commands. The special feature of this function is, that the port can be changed to any port number to hide the connection or tunnel trough a firewall.

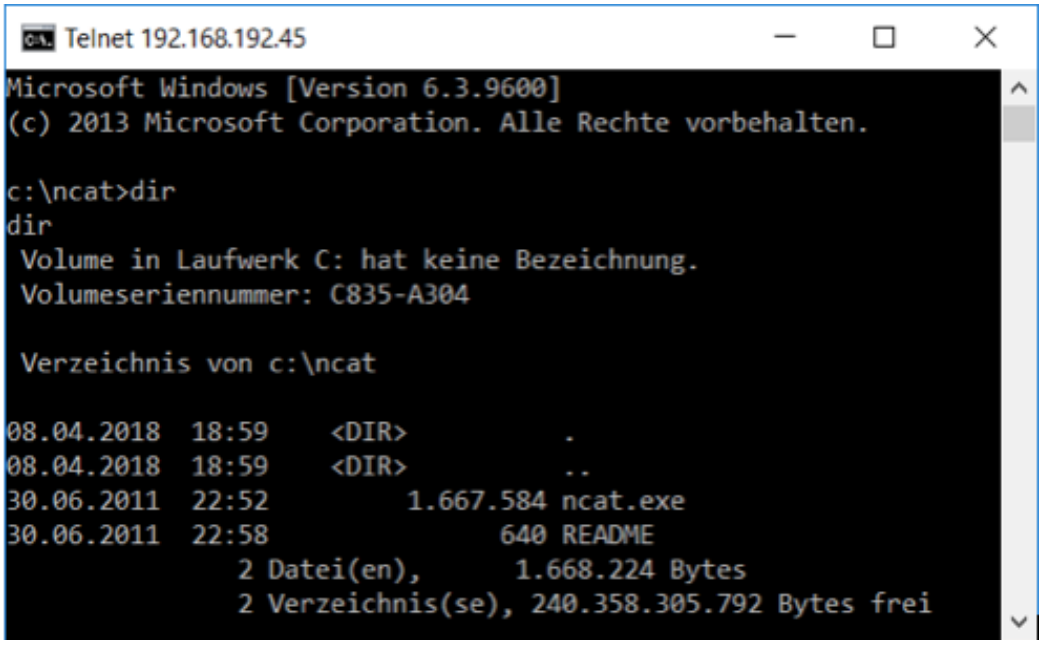


Figure 2.33: Telnet to execute commands

2.6.5 Transferring files

Today it is often too unsafe transferring data with Netcat cause of a lack of encryption. Please note therefore that if confidential data should be transferred, the usage of Hack 13 is recommended. However, it is often used for troubleshooting of web server, FTP servers or other services. Also, it offers great opportunities in the case, that the physical access to the target machine is quite difficult or if the transmission via USB 1.0 or USB 2.0 is to slow. The greatest advantages of Netcat file transfer mechanisms are the speed, simplicity and the portability.

Due to the fact that not all data transmission necessarily includes confidential information, it is in some cases quite appropriate to transmit them unencrypted over a network.

Instructions

Destination host:

First we need to instruct Netcat on the destination host to listen for an incoming request on a randomly chosen port. The command below initiates

netcat to retain listening on port 4711 until it receives a request for a transfer of file.txt.

```
nc -v -w5 -l -p 4711 > file.txt
```

Parameter	Abbreviation	Description
--listen	-l	Bind and listen for incoming connections
--source-port	-p	Specify source port to use
	>	Redirects the output of the command to file
--verbose	-v	Print out Messages
	-w 5	Wait 5 seconds after transfer to close connection

Figure 2.34: Listen on Port 4711

Source Host:

With the command below we can start the data transfer from the source host to the target host. In this process, the content of the file will be written into the standard input of the left command.

```
nc -v -w5 <target-ip> 4711 < file.txt
```

Parameter	Abbreviation	Description
	<	Use File as Standard-Input

Figure 2.35: Data transfer from source

2.6.6 Cloning of hard disks and partitions

During a forensic investigation, itś extremely important to write as less as possible on the investigated medium. Especially in a live analysis, this could be problematic, because each command can have an effect on the result. The advantage of Netcat is, that it can transfer data immediately from one machine to another over the network without being temporarily stored on the source machine.

Disk dump

The Disk Dump tool can be used to create bit-accurate copies of hard disks, partitions, or file. This tool is already included in the most linux-distributions and can be used with the path-variable ́dd:

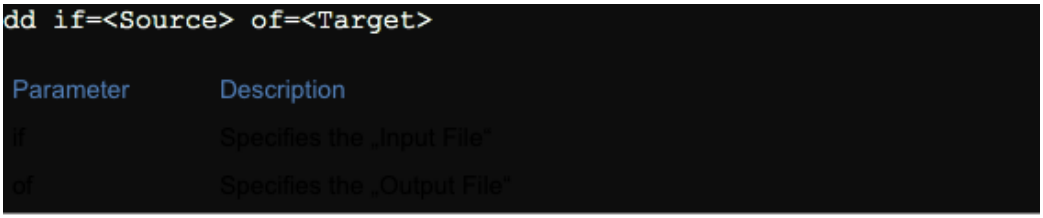


Figure 2.36: Disk dump

Instructions

The use of Disk Dump in conjunction with Netcat offers the opportunity to create a copy of a hard disk or partition and transmit it directly over the network to the forensic workstation. With the command below we can set the server on the listening mode:

nc ĺ ṕ ¡port¿ — dd of=¡Path¿/Image.dd

Example:

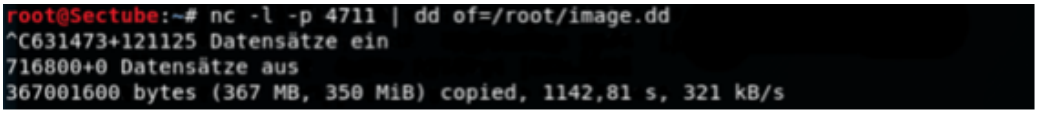


Figure 2.37: Server Listening

In this process the standard output of Netcat is written by the Piper-Operator into the standard input of the Disk Dump Tool. To identify the source-device we can use the command fdisk lu, that lists us all partitions of the system. For a more detailed explanation please refer to 13, Use-Case 1, Instructions. The transfer can be started with the command below:

dd if=¡device¿ — nc ¡ip-adress¿ ¡port¿

Conversely to the command on the server the standard output of Disk Dump is written in the standard input of netcat.

Example:

```
root@kali:~# dd if=/dev/sda1 | nc 192.168.192.38 4711
716800+0 records in
716800+0 records out
367001600 bytes (367 MB, 350 MiB) copied, 1118.66 s, 328 kB/s
```

Figure 2.38: nc

2.6.7 Port scans

The best way to understand how an attacker gained access to a system is to understand the motivation and the approach of various attack scenarios. Therefore, the use of port scanners can provide insightful information about existing vulnerabilities and open ports, which where exploited by an attacker.

Netcat can be used as a simple port scanner. In comparison to nmap, the tool does not provide as many options for this purpose, but it is enough to find out which ports are open in a system. For the execution of a port scan it is already sufficient to specify the target IP-address and the area that should be scanned:

```
nc -v <ip-address> -z <port-range (FROM-TO)>
```

Parameter	Abbreviation	Description
–verbose	-v	Print out messages
	-z	Just scan for listening daemons, without sending data

Figure 2.39: nc

Example:

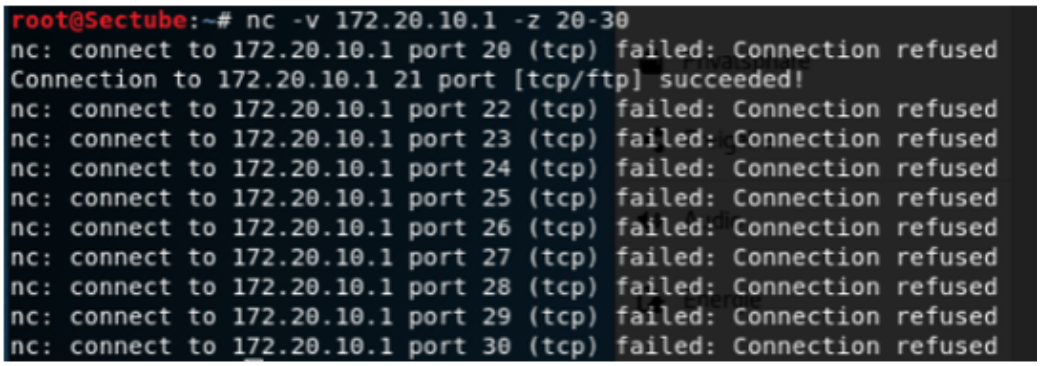


Figure 2.40: Netcat

2.7 Remote Imaging with encryption

2.7.1 Objective

This hack provides data transmission from a remote machine in a secured and encrypted manner by using the SSH protocol. We focus on two different approaches depending on what kind and size of data should be obtained. If an entire partition needs to be secured its advantage is to use the Secure Shell Filesystem (sshfs, see Case 2) and directly mount the target partition into your system.

2.7.2 Secure copying via ssh

SSH

SSH, also called Secure Shell, is a secure alternative to telnet and is used to establish secure connections using RSA Public Key Encryptions. For the forensic process it allows a simple and secure way to transfer entire hard disks and partitions.

Preparation

This Hack can only be used on Linux systems and assumes that the computer which should be backed up is started with a Linux Live CD/DVD. The installation of a SSH-Server on the forensic workstation allows that a client

can create and transmit a backup over the network. This can be easy installed by the package manager with the command below:

sudo apt-get install opensshserver

Due to the fact that the SSH Client is already part of almost all Linux-distributions, this step is already sufficient that a client can be connected with the server. However, it is important to ensure that the SSH-Server is running. The list below gives an basic overview about the administration of the SSH-service:

Command	Description
sudo service ssh status	Displays the status of the service
sudo service ssh start	Launches the service
sudo service ssh stop	Terminates the service

Figure 2.41: SSH service

To simplify and improve this process itś recommended to install a few more packages on the client:

```
sudo apt-get install pv gzip buffer
```

Parameter	Description
pv	The Pipe Viewer allows to monitor the progess of the filetransfer.
gzip	Gzip offer the opportunity to compress the transmitted file.
buffer	Ensures continious data flow and increases the data throughput

Figure 2.42: Installing packages

Instructions

Before we can start the transmission we need to identify the partition on the source host that should be transmitted to the target-host. The command

shell based tool fdisk allows us with the parameter lu to list all partitions on a data medium.

```
root@kali:~# fdisk -lu
Disk /dev/sda: 238.5 GiB, 256060514304 bytes, 500118192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe2bb9626

Device     Boot  Start       End   Sectors   Size Id Type
/dev/sda1  *      2048    718847    716800   350M  7 HPFS/NTFS/exFAT
/dev/sda2       718848 500115455 499396608 238.1G  7 HPFS/NTFS/exFAT
```

Figure 2.43: Tool fdisk

As we can see in the output, there are two devices available that can be chossen to back up. Similar to 13, Use-Case 4, we can use the Disk Dump Tool ́ddt́o create the image, with the difference that the host has to authentificate himself on the ssh-server of the target-machine by providing the username, the password and the ip-address of the forensic workstation. To select the source-device we can specify it by the device name. The ć́at-command is used to write the copied data into the image-file.

In this regard, it must be noted that the Parameters pv, gzip and buffer are optional and not necessary. However they are preferable to increase the data throughput and minimise the transmission time.

Example:

```
root@kali:~# dd if=/dev/sda1 | pv | buffer -s 64k -S 10m | gzip -c | ssh forensi
c@192.168.192.38 "cat > sda.dd.gz"
forensic@192.168.192.38's password:
716800+0 records in69KiB/s] [                                    <=>           ]
716800+0 records out
367001600 bytes (367 MB, 350 MiB) copied, 1522.71 s, 241 kB/s
 350MiB 0:25:22 [ 235KiB/s] [                                    <=>           ]
     358400K,        235K/s
```

Figure 2.44: dd command

During the transmission we can already find, for our example in the home-directory of the forensic -user on the target-machine, the created image-file with the name sdaḋdġz.

2.7.3 Secure copying via sshfs

FUSE

Filesystem in User space (also called FUSE) is a module for Unix-Systems, that allows the user to integrate own filesystems. The peculiarity of this principle is the fact, that also unprivileged users are able to mount filesystems. The mounted filesystems can then be navigated as if it were the own.

sshfs

sshfs is a tool for multiple platforms (Windows, Linux etc.) that is used to share files and directories on a remote machine This allows to add a directory tree of a remote machine to mount it on the ssh-server.

dc3dd

These tool is based on the code from dd. In contrast to dd it allows hashing with multiple algorithms (MD5, SHA-1, SHA-256 and SHA-512) to create a digital fingerprint. This allows to retrace if a image was manipulated during a investigation and allows to verify its authenticity.

2.7.4 Preparation

Just like the last use-case we need to install the OpenSSH-Server on the forensic workstation with the command:

sudo apt-get install opensshserver

On the client side we have to install the sshfs tool:

sudo apt-get install sshfs

2.7.5 Instructions

Before we can start to back up the client we need to start the SSH-Service on the target-host and create a directory, where the image is to be stored. This can be easily done with:

mkdir /Image

With this command we have created a Directory, named Image, in the home folder of the user. The next step is to create a directory on the client in which the Image-Directory of the Server can be mounted. In our example we

created a folder fuse in the home directory, that is named after the Filesystem in User space.

mkdir /fuse

By the execution of the subsequently command on the client, we can mount the directory Image. All files that weŕe moving now into the fuse-folder in the client are transmitted to the Image-folder on the Server.

Example:

```
root@kali:~# mkdir fuse
root@kali:~# sshfs forensic@192.168.192.38:Image/ ~/fuse
```

Figure 2.45: sshfs

To transmit a copy of an entire hard disk or partition we can use the tool dc3dd, which needs the source-device and the target-path as parameter. By the specification of a hash-algorithm we can create a hash-value, to retrace the authenticity of the image, in the same process.

```
dc3dd if=<source> of=<target-path>/image.dd hash=<algorithm>
log=<target-path>/log.txt
```

Figure 2.46: Tool dc3dd

After the transmission is completed we will get a logfile, that contains the details about the transmission and the created hash-value.

Example:

At the end of the backup process we need to unmount the remote directory on the client with the command fusermount and the parameter u (for unmount).

fusermount u /fuse

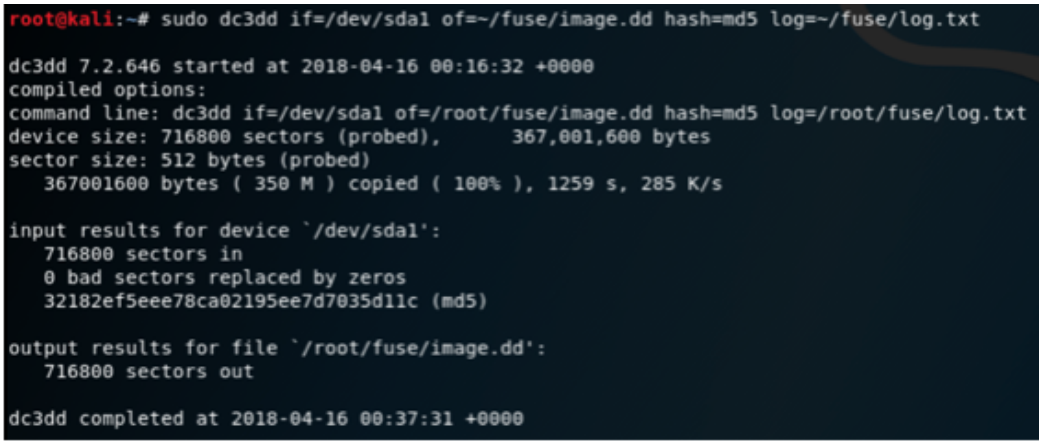


Figure 2.47: Created Hash Values

2.8 Data Recovery and slacker

2.8.1 abstract

Data recovery is a field of digital forensics. Often people delete by mistake files on computer. Sometimes files can be really important company documents. Also, hardware and software failures can result in loss of import data. Hackers can intentionally delete files to cover up their tracks or to damage a targeted company. In all of these cases, digital forensics experts are called. These experts know how to recover potential data or files. Special software can be used to extract data from hard drives. This approach from a software side only works when physical drives are still properly working.

When a physical drive is no longer detected by the computer data recovery is very complicated. In these cases, special hardware experts must be called and software alone is often no longer able to recover the data. Depending on the different type of physical drive it may be harder to recover data. Mounted flash drives display the data to the operating system in the same way as a magnetic storage such as hard disk drives(HDD) but how they store data physically works totally different. Magnetic storage devices are easier to analyse because it is easier to unmount disk physically inside the HDD. On these disks, the data is located. Often times only the technology to read and write to these drives is damaged.

When a flash memory is not detected by the hardware it is often difficult

to recover those because the chips are inside a circuit board. The memory chips need to be soldered out and this can be quite a tricky process. Additionally, flash drives constantly need to update the cells to create new empty cells. The operating system tells the SSD with the "TRIM" command to make certain blocks ready to use. In this process, old data is often finally cleared so the empty sectors are made available for reuse.

2.8.2 New Technology File System

Today many computer run a Microsoft Windows operating system. The default file system of Windows is NTFS since 1993. With Windows NT 3.1 the first version of NTFS was released. Another common file system that was developed by Microsoft is exFAT which fixes a lot of the old problems with the older FAT systems including FAT, FAT12, FAT16 and FAT32. For example, the maximum file size of FAT32 is 4GB. However, exFAT is not a journaling file system and will still be often used on flash drives such as SD-cards or USB-Sticks. NTFS is standard, however if mounted on Linux systems often additional drivers need to be installed. NTFS is journaling file system. A journaling file system is a layer of protection if errors or file corruption appears. For example, if a power failure happens and the system was not shut down properly data can be corrupted. However, a journaling file system keeps track of the write processes and if a failure occurs it can revert to the older state of a file. Because NTFS is not open source most of the functionality and information about the file system was reverse engineered over the years.

Figure 2.48: Structure of NTFS

File properties

Existing Files in windows can easily be inspected with a right click in the windows explorer. It displays all kinds of information about the file. A file

can have attributes like write protected, encrypted and etc. When a file is deleted it still contains this kind of information at an offset on the master file table. Another great feature is the support of access control lists(short ACL).

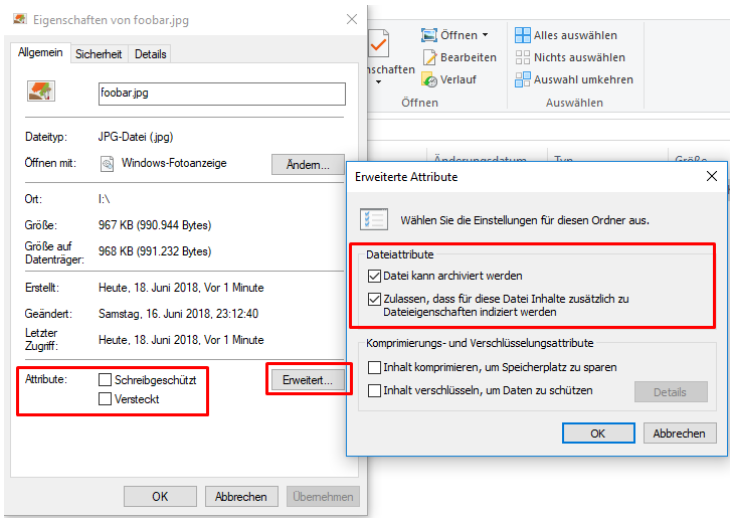


Figure 2.49: File attributes in Windows files

Master file table

The NTFS file system has a master file table which stores all kind of meta information of every file. The metadata consists of data like the name of the file, last edited date, size, first cluster address, etc. Every file has an entry inside the master file table. Each record has a fixed size of 1024 bytes. The most interesting information when recovering files is the allocation flag. This flag is located at the offsets 22 and 23 of each record. This flag contains information about the status of a file. A file can either be deleted or allocated. With this information recovery tools like Recuva can browse through the entry of files and essentially recover old files. So when deleting files on NTFS they are just marked as unallocated. However, when deleted marked files are overwritten by the operating system it is really hard to recover the data and sometimes impossible. Using a hex editor we can actually search for master file entries manually. Each entry starts with the master file table signature which is "FILE0" IN ASCII, this can be seen in 2.50. The first 16 entries

are system reserved. The first entry is actually the master file table itself. $MFT means master file table. The second entry is $MFTMir the master file table mirror. The $MFTMir is used to restore files when error occur because it stores a copy of a file before a change happens. A record start with a 48 byte header section as seen in figure 2.51.

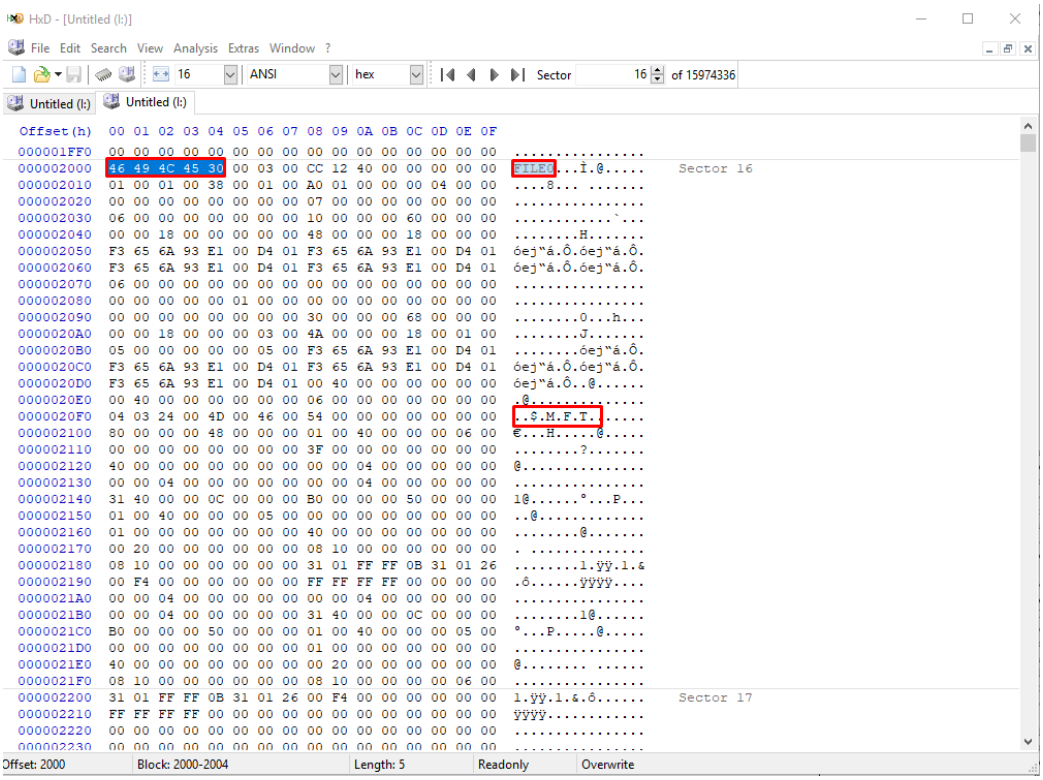


Figure 2.50: Hex editor showing the first entry of the master file table

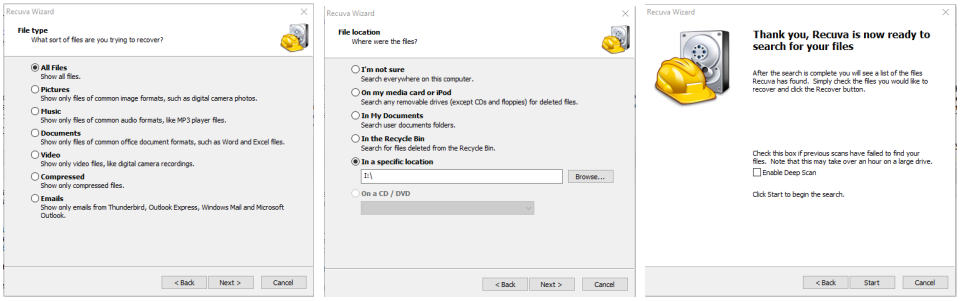
Data recovery with free software

Recuva Recuva is a free software written in C++ to recover data from hard drives. It works best with NTFS and has a graphical interface. It is only supported on Windows operating systems. It is possible to specify file types and the file location as seen in figure 1, which file types are supported are listed in table 1. Recuva needs a location to scan for files, the wizard easily let is you select the location as seen in figure 2.52b. The software offers two search modes:

Offset	Size	Description
00	Char 4	MFT record signature
04	Word	Offset to update sequence
06	Word	Number of entries in Fixup array
08	LongLong	$LogFile Sequence Number (LSN)
10	Word	Record usage number
12	Word	Hard link count
14	Word	Offset to first attribute
16	Word	Flags: 01 00 record in use, 02 00 directory
18	DWord	Actual size of MFT entry
1C	DWord	Allocated size of MFT entry
20	LongLong	File reference to the base FILE record
28	Word	Next attribute ID
2A	Word	
2C	DWord	MFT record number
30		Fixup values and Attributes

Figure 2.51: MFT Record header layout

- normal scan
- deep scan



(a) Select file type in recuva (b) Select where to look for files (c) Choose between normal and deep scan

Figure 2.52: Recuva Wizard

The normal scan looks inside the master file table for deleted, corrupted and overwritten files. This scan is very fast. By default, Recuva will use the normal scan as seen in figure 2.52c. , if the drive was formatted or damaged most of the time the normal scan is not sufficient. In that case, the mode deep scan might work. Deep scan does file carving and because of that, it is a lot slower. File names are also not recovered by the deep scan.

Table 2.1: Supported file types for deep scan

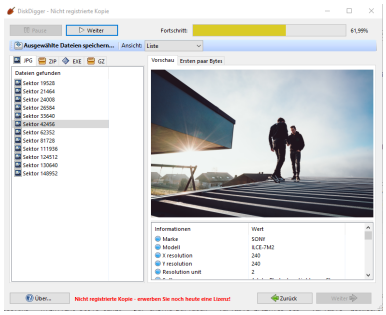
Graphics	BMP, JPG, JPEG, PNG, GIF, TIFF
Microsoft Office 2007	DOCX, XLSX, PPTX
Microsoft Office (pre-2007)	DOC, XLS, PPT, VSD
OpenOffice	ODT, ODP, ODS, ODG, ODF
Audio	MP3, MP2, MP1, AIF, WMA, OGG, WAV, AAC, M4A
Video	MOV, MPG, MP4, 3GP, FLV, WMV, AVI
Archives	RAR, ZIP, CAB
Other file types	PDF, RTF, VXD, URL

Source: https://www.ccleaner.com/docs/recuva/using-recuva/wizard-mode/the-deep-scan-option

Testdisk Unlike the other tools, Testdisk only comes with a command line interface. This software is also designed for partition recovery. This tool can help to recover damaged boot drives.

Diskdigger This is free software for private use only. This tool does not need an installation and also is able to scan DD-formated images.

Figure 2.53: Diskdigger



Glary Undelete Another free software which also supports to search for a lot of different file types. This software has the overall best filter options. Nonetheless, this software does not offer the carving method. This software also comes with some adware.

Conclusion There are a lot of different tools to recover deleted data. Mostly all of them offer the utility for a normal scan that only checks the master file table. And they also offer another mode that uses carving to scan for certain files types. They may differ in speed but the results are mostly the same.

2.8.3 Slack

Hard drives consist usually of 512 byte sectors. Depending on the configuration cluster sizes can be different. The cluster size is the minimal size the hard drive allocates for a file. Standard cluster size on NTFS is 4096 byte. That means 8 sector combined are one cluster. Since the smallest file size has to be 4096 bytes a smaller file that usually would only take 1 byte still reserves the full cluster. 4095 bytes are unused. That remaining space is called slack. There are 2 different kinds of slack spaces. For example, the file takes up the first sector, however only the first byte. The last bytes in this sector are written with null bytes. This is called the RAM-slack. Former random RAM memory was written at the end of the sector. Unfortunately, this is not secure since the RAM can consist of sensitive data. In this case, the last 7 sectors are drive slack. They contain data that was written on these sectors before. Figure 2.54 shows that file A writes its content to 6 sectors of the cluster. The last 2 sectors contain whatever data was stored before. File B overrides the cluster where file A was stored. However, B is smaller than A and for that reason, 2 sectors with data from A still exist.

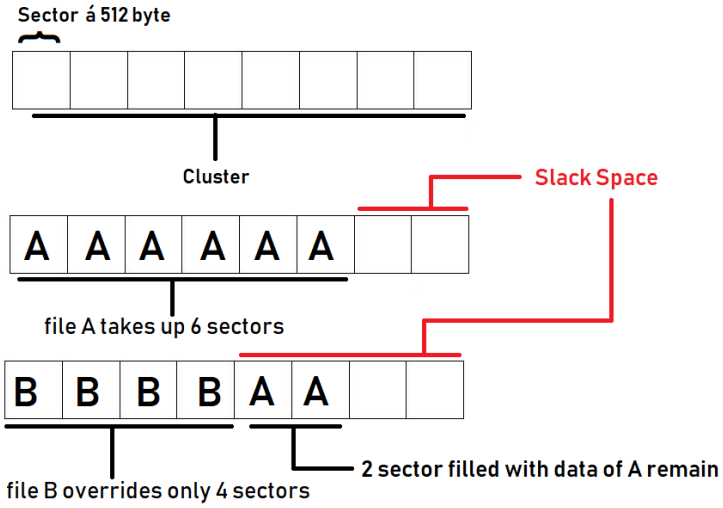


Figure 2.54: Slack space

Digital forensics expert should know about the slack area and how to evaluate it. The drive slack may contain information about a person doing however, data must be assignable to a person. That is not the case with drive slack. For that reason, the information inside the drive slack is not

valid in court, but it can help for further investigation since it is possible it contains some password keys. Hence the data is in most cases never complete analyzing the data can become quite difficult. A criminal could also use the slack space to hide certain data. Locating such data proves to be extremely time-consuming.

2.8.4 Alternate data streams in NTFS

In Windows XP, Microsoft introduced alternate data streams. Whenever a file in NTFS is created the meta informations are stored inside the header of the $MFT. The header inside the $MFT is followed by the attributes. One of these attributes is the $DATA section which stores the file data. Typically files only have one data stream that is the unnamed data stream. The name is because the default data stream has no name as seen in figure 2.55. The colon operator is used to attach a data stream to a file. So by default windows loads the empty data stream as the file data.

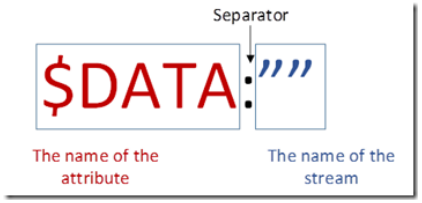


Figure 2.55: Accessing streams in NTFS[?]

Alternate data streams do not show up in the directory listing and they do not increase the file size of the original file. So data can be hidden in those streams. With the knowledge from figure 2.55 we can write and read data streams. In figure 2.56 we will create a file and write some text to it. As we can see it actually has a size of 34 bytes. The total free space on this drive is 8.150.261.760 bytes.

We can actually use the notepad to open such a stream and write some content to it. In figure 2.57a a secret message is written to the alternate data stream called secret. The secret data would not be visible if we printed out the content as shown in figure 2.57b. However, we can use notepad again to view the content of the alternate data stream. The additional parameter /r is necessary to list data streams in the current directory. After creating the alternate data stream the actual file size is still the same. We can see that

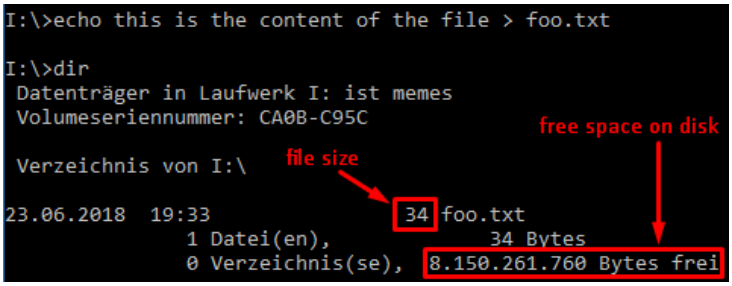
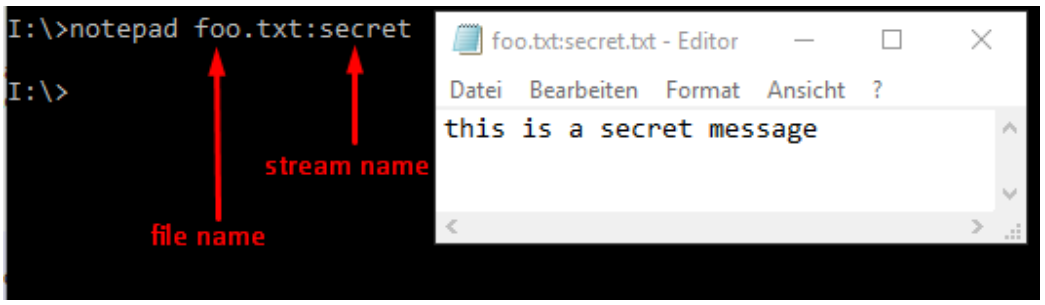
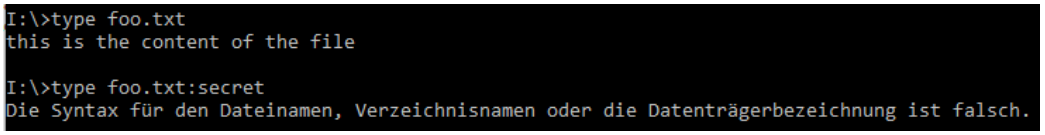


Figure 2.56: Create a file from the windows command line

when we compare the allocated file size of foo.txt on figure 2.56 and 2.58. The free space of the drive also did not change.



(a) Writing hidden data



(b) the command type cannot interpret alternate data streams

Figure 2.57: Analyzing alter data stream in command line

Not only files but also hidden software could be hidden in such an alternate data stream. It is possible to attach a stream to a directory. Most anti virus software will scan these streams, for example Malwarebytes Anti-Malware scans for and removes unwanted ADS (as Rootkit.ADS) [**?**].
With Windows 8 or higher, you can use the Powershell to create, list and view streams [**?**]. You can use *Set-Content* and *Get-Content* to write to a file or view the data of the file. With *Path* you can define what file or directory you want to look at. The *Stream* object will use alternate data streams instead of the default one.

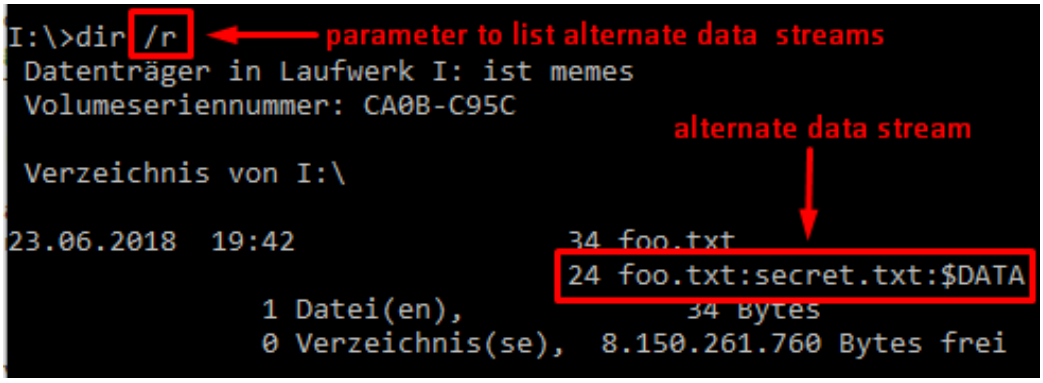


Figure 2.58: List alternate data streams

2.9 History of actions

2.9.1 Introduction

This hack deals with the history of actions from a user, which means a digital forensics expert can find tracks and evidence of possible illegal activities. In order to understand how to extract that informations, a baseline understanding of the technology to provide a good foundation for how and why prefetch files contain certain data is needed.

What are prefetch files?

Windows prefetching started with Windows 2003 Server and Windows XP and is still used on Windows 10. Prefetch is a feature, that stores specific data about the applications a user runs in order to help them start faster. Furthermore prefetch is a algorithm that helps anticipate cache misses, and stores that data on the hard disk for easy retrieval [1]. There are three different types of prefetch files:

- **boot trace:** the purpose of this kind of prefetch files is to speed up the operating system when itś being started or rebooted.
- **application:** the intent of this prefetch files is to speed up the time it takes for Windows to load certain applications, which includes software like cmd.exe, notepad or other third party applications.
- **hosting application:** the last type of prefetch files records the trace activity of certain programs that are used to spawn system process,

such as DLLHOST.exe, RUNDLL32.exe and so on. Windows needs to keep track of the different processes, that were started by applications, which is why they are categorised separately as hosting applications

All types of prefetch files are located in the prefetch directory, which normally can be found under CWindows independent from the running Windows operating system. Do not get confused by the file names because, their naming convention is unique for each type of prefetch files. The most common prefetch file types are the application prefetch files, which filenames contains the application name followed by a thirty two bit hash or number represented in hexadecimal, which is an indicator for the location of the application and finishes with the ?.pf? extension [2].

With all this informations a forensics expert may ask now what is the forensics value of the prefetch file?

Forensics value of prefetch files

Prefetch files can sometimes answer the vital questions of digital forensic analysis: **who**, **what**, **when**, **where**, **why** and sometimes even **how**. Furthermore there are two different perspectives a investigator can examine informations from prefetch files, the content itself or the creation of the existence in the prefetch directory.

Firstly let?s have a look at the content of prefetch files. If you are interested how analysing tools for prefetch files work, you can take a first look at the content using an hex editor.

You can download this tool at: **http://www.mitec.cz/Downloads/HEXEdit.zip** In addition you may need further information how to translate these hexvalues in actual metadata.

http://www.mitec.cz/Downloads/WFA%20Guidance.pdf

Of course you wont́ use a hex editor to examine the data, which are of forensics value for you, but it is a good insight for understanding the inner workings of analysing tools, which will be used in the later section of this document.

No matter which way you going to look at the prefetch files, it will be pretty obvious what informations are of value for your forensics examination. The metadata contains the application name, executed location and different timestamps for instance the last runtime. These artifacts might answer the ?what?, ?where? and ?when? of an incident.

The second half of prefetch files are written in plain text, but can be challenging to read. Tools can organise this content, making it easier to

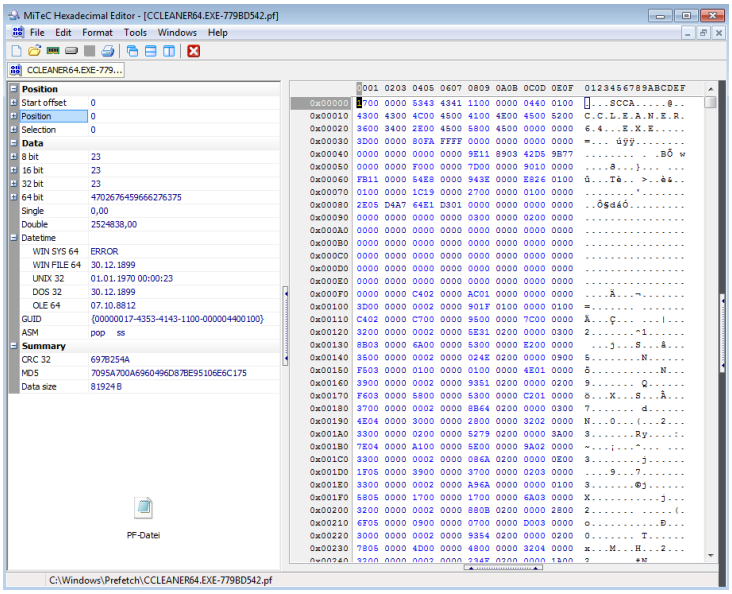


Figure 2.59: hex editor overview

read and to identify artifacts of interest. Scanning all locations, where an application was running may reveal hidden or obfuscated directory locations, for instance a TrueCrypt volume. TrueCrypt is a software, that has the ability to hide directories from view, finding this path listed in a prefetch file can provide a data source that might not otherwise be identified. By just browsing the contents of prefetch files it is possible to identify an obfuscated directory.

Often, hackers will hide tools in plain sight in unusual directories in the System32 folder. The System32 directory is a folder that contains many programs used by the operating system. Most users do not browse this directory. The full directory path in the prefetch file can also provide any user accounts, this could reveal a temporary account used for malicious activity by showing programs that were executed sometime in the past by a potential unauthorised user. This may answer the ?who? question for a forensic exam, or at least narrow the scope[2].

Lastly analysing the full paths in the prefetch files can show that an application was accessed from an external storage device. With that information you are able to compare the last access time in the prefetch file with the timestamps in the USBStor registry key. If you identify matching timestamps the

USBStor registry key entry will contain a serial number of the device. This can lead to other devices that need to be seized and analysed. Identifying unaccounted USB storage devices and applications or files accessed on those USB devices might help in answering the ?what? and ?why? questions [2].

The next section of this document will be focused on the tools you can use for your forensics examination of prefetch files.

2.9.2 Tools

Windows File Analyser

The Windows File Analyser, as mentioned in the book Computer Forensik Hacks decodes and analyses Prefetch-Files which are saved in the folder Prefetch, located within C:/Windows. These files contain interesting information about forensic analysis. This utility is very user-friendly and therefore easy to use. [5]? Download: *https://www.mitec.cz/wfa.html*

This program doesnt́ need any installation, thus you just can run it. After opening just select File Analyse Prefetch. Then pick the prefetch-folder of your choice and the Windows File Analyser accomplishes its duty.

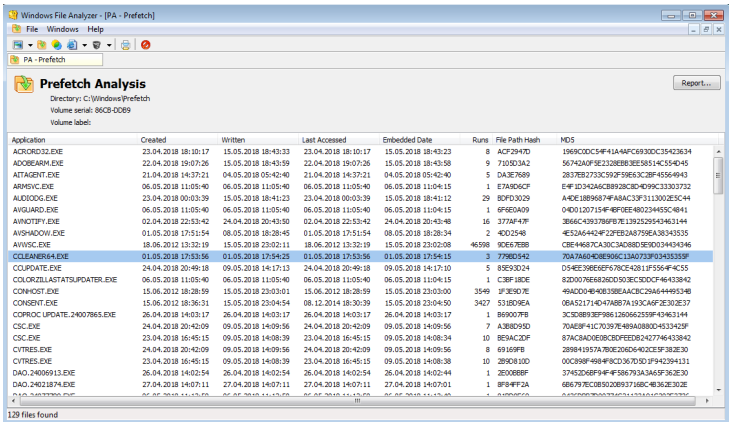


Figure 2.60: Windows File Analyser overview

Unfortunately, Windows File Analyzer can only be used by Windows OS ending with Windows 7 and Windows Server 2008. Since Windows 8 or newer isn?t supported by WFA, there have to be other programs to come into use. For example: WinPrefetchView.

WinPrefetchView

WinPrefetchView is used to read the informations which are stored in those prefetch-files. With this program you are able to display these informations onto your screen. There you can see clues when which applications were run last and what files were loaded by them. Furthermore you can see which files are loaded on Windows boot.

How to use winprefetchview?

Download: **http://www.nirsoft.net/utils/winprefetchview.html**

After downloading, you get a zip file you need to extract. This zip-file contains a .exe-file you only have to run. There is no installation needed. You can also run this .exe-file on a external USB-flashdrive to keep your system untouched. Usually, winprefetchview automatically selects the original windows prefetch-folder, but within the application you are able to select another target-folder.

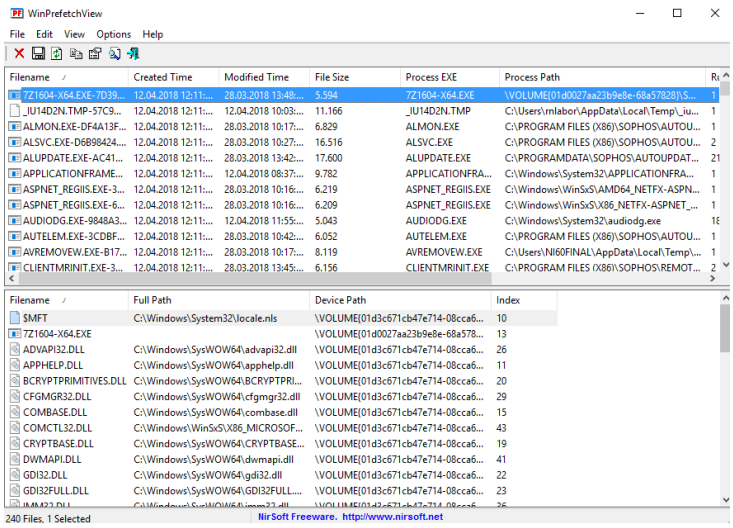


Figure 2.61: WinPrefetchView Overview

Here you can analyse a single file. To do so select one file and click on File and Select Properties (alternative: select one file and press Alt + Enter)

In using winprefetchview with the command interpreter (e.g. CMD or Power shell), you now are able to access the advanced options of the tool, for example selecting a certain prefetch directory for your analysis. To use another directory type in the following command

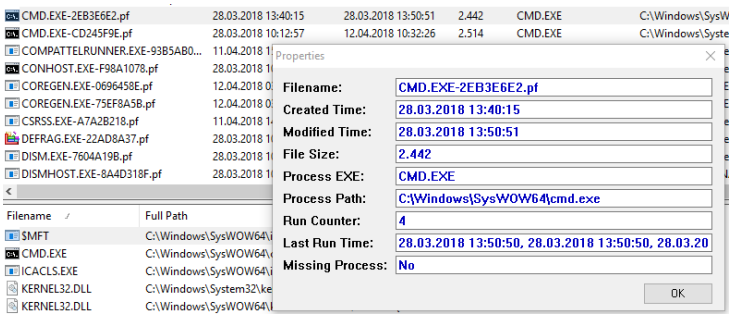


Figure 2.62: WinPrefetchView properties

prompt: [path]/WinPrefetchView.exe /folder [path]/[directory name]

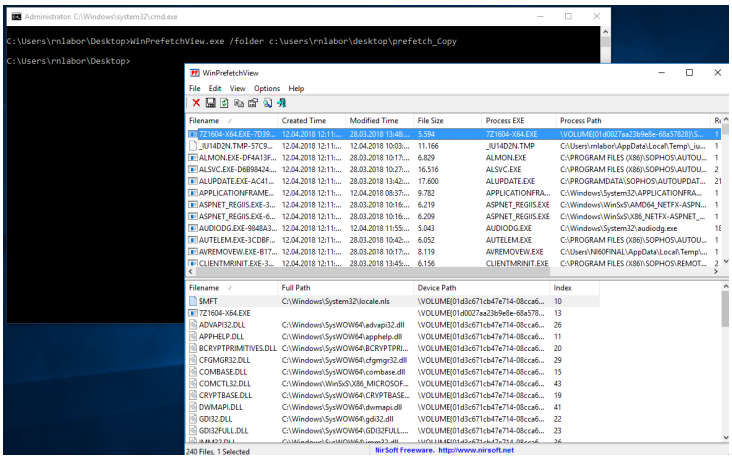


Figure 2.63: WinPrefetchView cmd usage

There are some ways how you can export the prefetch-file informations within the command line.

Save it in a normal .txt-file

prompt :[path]/WinPrefetchView.exe /folder [path]/[directory name] /stext [path][filename]

You can also save it in a. html-file, .xml-file or in a tab-delimited text file to get a better overview.

To simplify your searching, you can sort you findings. In this case, we sorted our findings by the "Run-Counter" (i.e. how often a application has been started) and saved it into a .html-file:

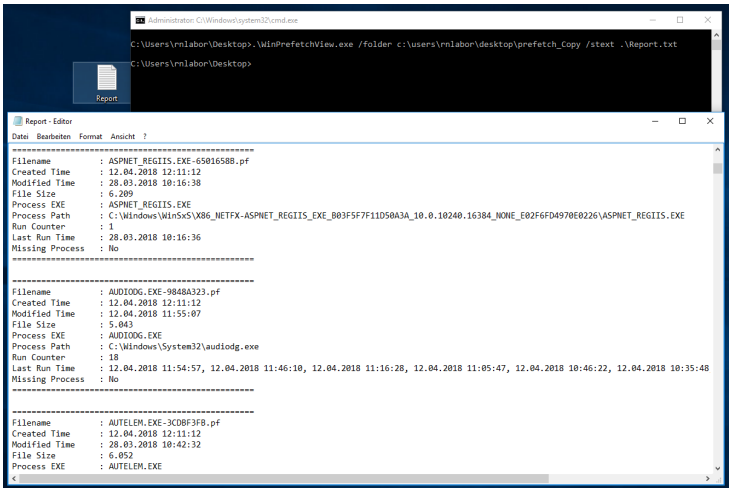


Figure 2.64: WinPrefetchView cmd usage

prompt: [path]/WinPrefetchView.exe /folder [path]/[directory name] /shtml [path]/[filename] /sort ” Run Counter”

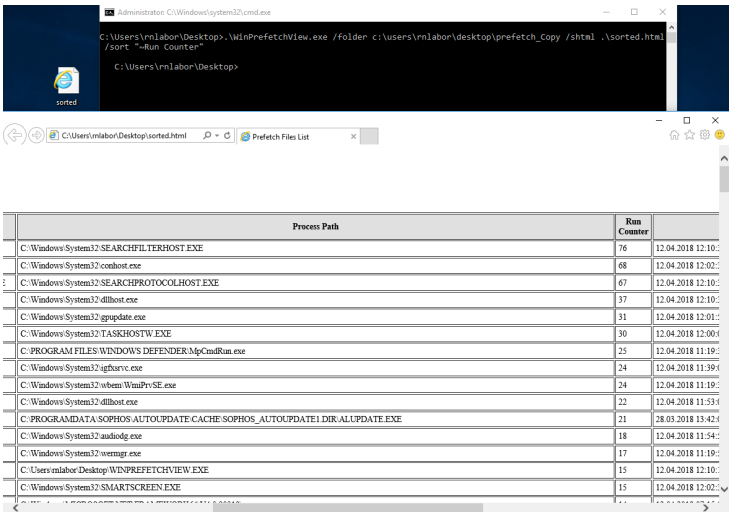


Figure 2.65: WinPrefetchView cmd usage

If you have a clue, e.g which application was used to commit a crime, you can specifically extract one certain prefetch-file to analyse its loaded files and the last run time.

prompt: [path]/WinPrefetchView.exe /folder [path]/[directory name] /sverhtml [path]/[filename] /prefetchfile [path]/[filename]

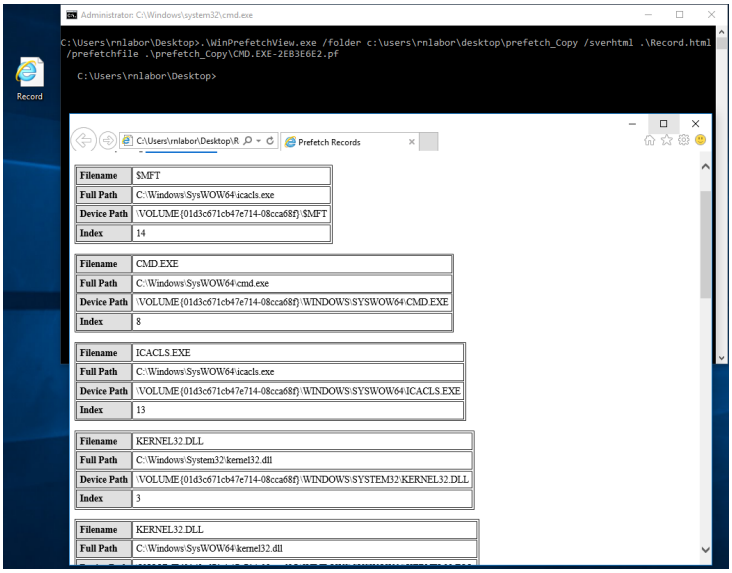


Figure 2.66: WinPrefetchView cmd usage

Of course there are more commands for this utility, but this is not necessary to show every single command. Here is a list which commands can be used to analyse the prefetch-files:

/folder <Folder>	Start WinPrefetchView with Prefetch folder from another instance of Windows operating system.
/prefetchfile <Filename>	You can use this command-line parameter with the other save commands (/shtml, /stab, and so on) in order to export the records of specific .pf file into text/html/csv file, for example: WinPrefetchView.exe /shtml "C:\temp\records.html" /prefetchfile "C:\windows\Prefetch\NTOSBOOT-B00DFAAD.pf"
/stext <Filename>	Save the list of Prefetch files into a regular text file.
/stab <Filename>	Save the list of Prefetch files into a tab-delimited text file.
/scomma <Filename>	Save the list of Prefetch files into a comma-delimited text file (csv).
/stabular <Filename>	Save the list of Prefetch files into a tabular text file.
/shtml <Filename>	Save the list of Prefetch files into HTML file (Horizontal).
/sverhtml <Filename>	Save the list of Prefetch files into HTML file (Vertical).
/sxml <Filename>	Save the list of Prefetch files into XML file.
/sort <column>	This command-line option can be used with other save options for sorting by the desired column. If you don't specify this option, the list is sorted according to the last sort that you made from the user interface. The <column> parameter can specify the column index (0 for the first column, 1 for the second column, and so on) or the name of the column, like "File Size" and "Filename". You can specify the '~' prefix character (e.g: "~Created Time") if you want to sort in descending order. You can put multiple /sort in the command-line if you want to sort by multiple columns. Examples: WinPrefetchView.exe /shtml "f:\temp\Prefetch.html" /sort 2 /sort ~1 WinPrefetchView.exe /shtml "f:\temp\Prefetch.html" /sort "~Modified Time"
/nosort	When you specify this command-line option, the list will be saved without any sorting.

Figure 2.67: WinPrefetchView cmd commandlist

2.10 Metasploit

2.10.1 Introduction

Metasploit is a very strong security testing tool, as it is a very feature rich platform. The framework is a complete platform for performing vulnerability testing and exploitation, loaded with thousands of exploits and hundreds of payloads.

Performing an exploit using Metasploit will normally lead to either a Remote shell to the target computer, which is a remote terminal connection or a Meterpreter shell, offering many programs and utilities that can be run to gather information about the target computer or control devices like the webcam or microphone.

Metasploit is handy tool for forensic analysts, as it can provide a remote shell to a target computer that can be used to obtain information without changing the data. It?s possible to download files, such as log files from the target system, as well as recovering data without direct access. However, this purposes that the analyst has the owner?s permission to do so.

2.10.2 Installing Metasploit

Metasploit is preinstalled on Kali Linux. Especially on Kali Linux 2.0 the Metasploit terminal can easily be opened by clicking the Metasploit framework button on the Quick Launch tab. Using another OS Metasploit framework can be accessed by downloading the corresponding installer on

http://www.rapid7.com/products/metasploit/download.jsp. As Metasploit is detected by AV software as malicious this could cause problems, thus the AV software should be disabled before installing Metasploit. Firewalls can also interfere with the download, detecting the framework as malware, hence the local firewall should be disabled first. To start the msf prompt simply navigate to the Metasploit directory and start the console.bat under Windows or type ./msfconsleúnder Linux.

2.10.3 Definitions

Exploits

Exploits are the attacking methods. Metasploit has a wide collection of exploits of different OSs, that can be used for many safety and penetration tests. On the other hand, this large number of exploits can be used to intrude into other devices. Exploits concentrate on Software and Hardware vulnerabilities on the respective system. It?s important to carefully chose the exploit depending on the target OS and configure the corresponding options.

Payload

Payload is the executable code, that is run once the system has been intruded. Examples of payloads are different kinds of shells, such as the Meterpreter, that autonomously build up a reverse connection to the attacking machine. Metasploit contains many different types of payloads collected in a Database, that can be listed using the command ?show payloads?. All payloads are handled by the Multi Handler, no matter what architecture or connection type is used.

Msfvenom

The msfvenom command can be used to create a payload file. It combines both tools, msfpayload and msfencode. Msfpayload is compromising the

process of selecting a payload, set the payload and set all necessary options, msfencoder re-encodes the payload to hide it from VS. By using msfvenom re-encoding and embedding the payload can be done by one single tool at once. Msfvenom has many options that can be used along with this command, shown by ?msfvenom -h?. An example of this is shown in the Hack, later.

2.10.4 Basic steps

Picking an exploit

Metasploit contains about 1500 exploits and is frequently being expanded. To see all exploits type show exploits in the msf prompt.

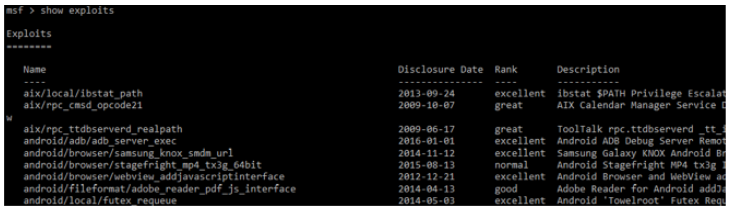


Figure 2.68: Show exploits

Another way to search for an exploit is using the search command. In this way you can specifically search for exploits for example depending on the target platform. Typing "help search" will show the options available. To know more about one specific exploit the command ?info? together with the exploit path can be used to display the information. For example, having a closer look on the adobe_pdf_embedded_exe exploit on windows will show the available targets of this exploit, the options that can be set and a brief description about what this exploit does. So, this exploit is embedding the payload into an existing PDF file.

To select a chosen, exploit the use command followed by the path of the exploit must be run.

As we have now set up the exploit we need to set the options.

Setting exploit options

To see what variables can be set for the chosen exploit use show options. This will show the names of the module options, their current values, a short description and if they are required or not.

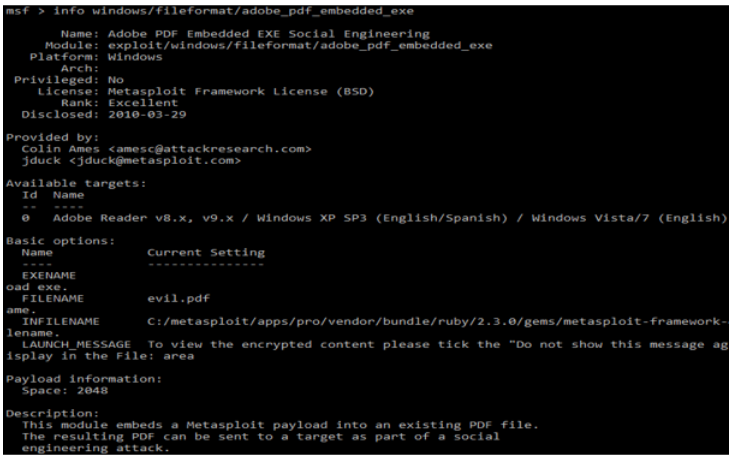
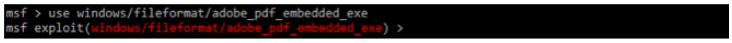


Figure 2.69: Exploit info



In this example we can see, that the FILENAME is evil.pdf. As this is obvious we should think about changing that. To set a specific variable the set command needs to be used, followed by the option name and the value.

Figure 2.70: Setting exploit

Picking a payload

To list all the possible payloads that can be selected, use the show payloads command. Most of them have the standard layout of $< OS/SHELL_TYPE/CONNECTION_TYP$
.

Now a payload depending on the desired shell type such as remote shell or Meterpreter and the different ways of how the payload communicates with

Figure 2.71: Picking payload

the attacking machine needs to be selected. Usually a reverse_tcp connection is preferred, as the outgoing connection initiated by the target system won?t be blocked by its firewall, whereas an incoming connection most likely will.

Using the set command the chosen Payload can then be set.

Setting payload options

Payloads have options that can be set, the same way we set the exploit options before. Using the show options command again will now show an additional section with payload options.

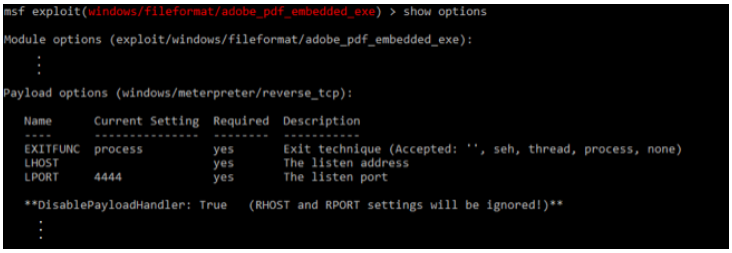


Figure 2.72: Setting payload

Apparently the required LHOST option is not set now. To set this value the set command must be used.

Running the exploit

After all the needed options are set, the exploit can be run using the command exploit.

2.10.5 Create payload for meterpreter session on Windows 10

In this Hack we want to gain access to a windows 10 target device using a reverse tcp connection to start a Meterpreter session. Therefore, we will be using the msfvenom command. So, we will start by opening the msf prompt and using the msfvenom command together with the needed options to create our payload file.

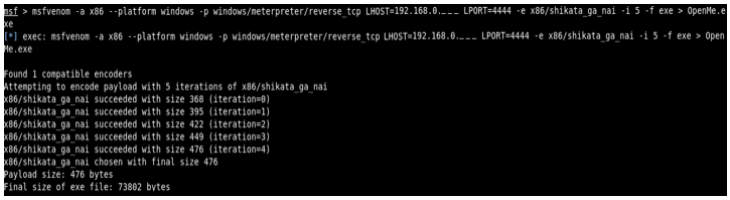


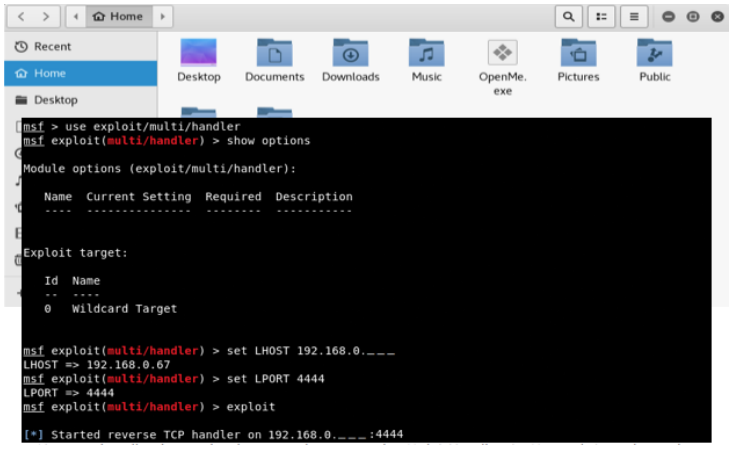
Figure 2.73: Payload

As this command is combining the functions of creating and encoding the payload we start with specifying the payload. The architecture is set to x86 with -a followed by the target platform specified by ?platform. Next, we need to choose our Payload specified by -p. As we want to have an extended access to the target system, we will be using the Meterpreter shell. Also, we want to set up a reverse tcp connection. Now, we need to specify the options of the chosen payload which is the LHOST and the LPORT. As msfvenom is not just embedding the payload but also re-encoding it we need to select an encoding mechanism with -e. A list of all encoding mechanisms in Metasploit can be accessed with the msfencode -l command. In this case we will be using the x86/shikata_ga_nai encoder. Finally, we need to select the output file format, here exe and the name of the output file.

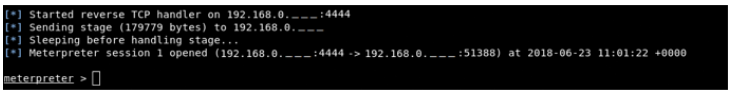
After executing this command, we will see, that it has created an .exe file named OpenMe.exe.

Now, to handle the payload we need to start the Multi Handler in Metasploit and set the needed options which are LHOST and LPORT. Using the exploit command, we can start the handler, waiting for an incoming connection.

Now we can send our payload file to the victim either by email or attaching it to a pdf. We must make sure to appear trustful enough so that the file will be opened. Once the .exe file has been run, nothing will happen on the target



system, but in the background a connection to our system will be established and the Meterpreter shell will be started giving us fully access to the target system.



Now we can use all the utilities offered by the Meterpreter shell. For example, we can edit files, download files from the target system on our own system or access hardware components such as the microphone or webcam.

To take a snapshot from the webcam we simply need two commands. First, we need to find out if there is a webcam to be used with the command webcam_list. Then we can use the command webcam_snap -i WEBCAM_ID to take a snapshot and save it on our computer.

2.10.6 Meterpreter commands

More Meterpreter commands

keyscan_start	starts the keylogger software
keyscan_dump	dumps the content of the keylogger software
keyscan_stop	stop the keylogger software
recod_mic -d SEC	record audio for SEC seconds using the target systems microphone
screenshot	save a screenshot of the target system on your device
webcam_list	lists all webcams on the target system, each with an ID number
webcam_snap -i ID	take a picture with the ID webcam
run webcam	taking pictures in a loop, refreshing the displayed picture
download	download a file from the target device to your device
upload	upload a file from your device to the target device

2.11 Word Document Artifacts

2.11.1 Introduction

The Microsoft-Office suite is probably the most widely used word-processing tool when preparing and writing documents, spreadsheets and presentations. Starting with the 2007 version (Microsoft Office 2007), Microsoft has completely changed the format of its document files, from the binary doc format to basically a zip-file that contains all the xml-files pertaining to the document. In Order to perform this hack you need to get an basic understanding on the architecture and function of the docx-file format.

Construction of a docx-file A docx-file adopts the OOXML format, which is based on XML. The eXtensible Markup Language (XML) is used for the representation of structured data and documents and thus is composed of instructions, defined as tags and markers. Therefore, in XML a document is described, in form and content, by a sequence of elements. Every element is defined by a tag or a pair start-tag/end-tag, which can have one or more attributes.

So basically a docx-file is a container like files with a zip extension and contains:

- XML-files, which describe application data, metadata and even customer data, stored inside the container file
- non- XML-files, may also be included within the container, including such parts as binary files representing images or OLE objects embedded in the document
- relationship parts thats specify the relationships between the parts, this design provides the structures for an Microsoft Office file

In Addition, with the usage of the OOXML format :

- it is possible to show just the text of the document. For example in a Word document only the document.xml will be analysed without opening all the files which contain the remaining informations of the document
- the files are compressed, thus they are short and easy to manage

- it is simpler to scan for viruses or malicious contents thanks to its textual format
- if some of the files in the zip container are damaged, the integrity of the entire document could be preserved, and in some case the main document could be reconstructed.

Further explanations and illustrations about the inner workings of docx-files will be displayed in the later section, which describes the manual extraction of data.

2.11.2 Hidden information in a docx file

Most people are unaware that the documents they create and edit are used by Microsoft Office suite, which contains a large amount of data related to the lifecycle of the documents. There is a big selection of information types in a docx-file, but not every information is easy to discover. Some of the collectable data needs special tools to unveil its content. In general the extractable informations are called metadata:

- name
- initials
- company or organisation name
- name of your computer
- name of the network server or hard disk where you saved the document
- other file properties and summary information
- non-visible portions of embedded OLE objects
- names of previous document authors
- document revisions
- document versions
- template information

- hidden text
- personalised views
- comments

Normally the informations in a docx-file aren?t the decisive factor in a forensic examination. These informations are often used to get a clue of what could have been happened or who could have been the culprit. How these informations can be found will be explained in the following pages.

2.11.3 Extract Metadata from a word document

Manual extraction

In the first step you have to change the extension .docx to a .zip file to open the ?hidden information?.

Figure 2.74: Converting .docx to .zip

There is the content of the ZIP-file.

The rels rels-file contains information about the structure of the document. This relationship file contains XML-specified information about how parts in the XML document interact with each other. Rels files are usually stored in subfolder rels.

The docProps-folder contains both a app.xml and core.xml. The app-file stores the information of the application itself. Which version of Microsoft Office was used or e.g. whether the ?DocSecurity? is activated or not. The DocSecurity can be deactivated, password protected or even Read-only. The other file is the core-file. This is the most interesting part of the document

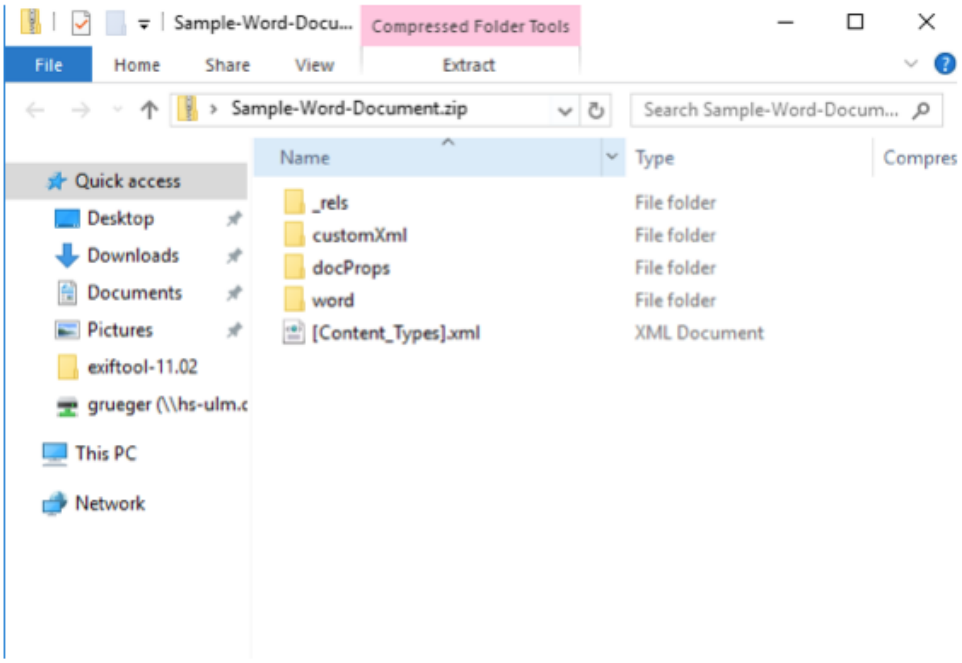


Figure 2.75: Content of Zip file

extraction. In this core-file you can see both the ?original creator?, the ?creation-time? and the ?lastModifiedBy? and ?lastModifiedTime?.

The word-folder is the folder where the main document.xml-file is located, these document contains the content of the document itself. There is also the comments.xml -file, if someone added a comment within the docx-file. You are able to hide text in the document or in the comments. Normally such hid text isn?t too easy to discover. The hidden content can be uncovered with extracting the metadata out of the docx-file.

Exitftool

ExifTool is a platform-independent Perl library and command-line application for reading, writing, and editing metadata in a variety of files [3]. Download: $https://www.sno.phy.queensu.ca/\ phil/exiftool/$

This application does not require any installation, thus you just need to start it. There are two ways to extract the metadata out of the document files.

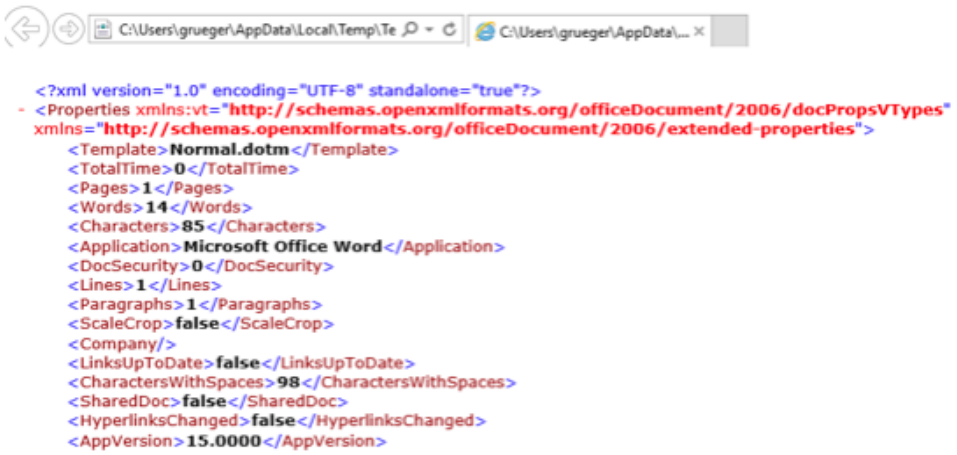


Figure 2.76: Content of app.xml

Figure 2.77: Content of core.xml

The first option is the easier one. Here do you have to drag the file, you want to analyse onto the exiftool(-k).exe. After dropping the file, a command-line window opens and shows the metadata which is saved within the docx-file. (-k) is important, because this expression causes the window not to be closed immediately after opening.

In using exiftool.exe with the command interpreter (e.g. CMD or Power shell), you now are able to access the advanced options of the tool. That is the second option to use exiftool.

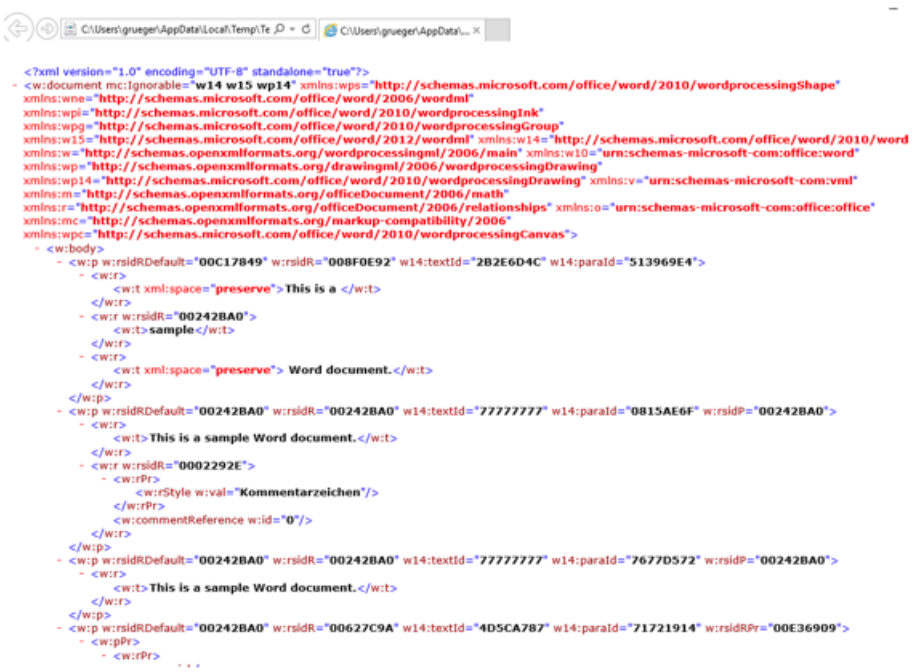
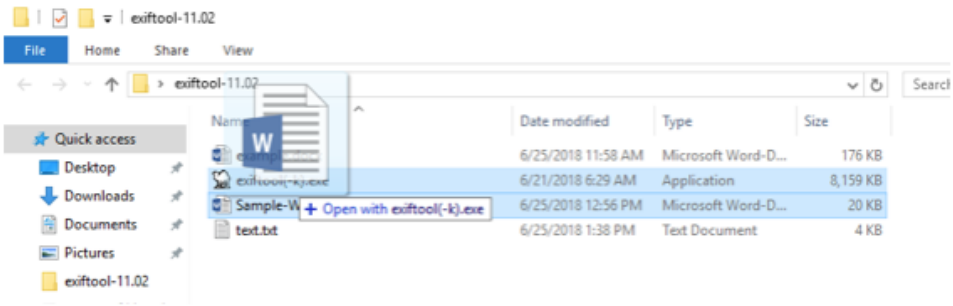


Figure 2.78: Content of document.xml



To get a short overview of the possible command-line options, please follow the link below: *https* : *//owl.phy.queensu.ca/phil/exiftool/examples.html* The following table lists tags that exiftool.exe observes in a OOXML documents. Exiftool.exe extracts all tags from the XML-files of the OOXML document properties directory ("docProps").

To get information to the complete description and usage of exiftool.exe, just use the following command below. Within the complete description,

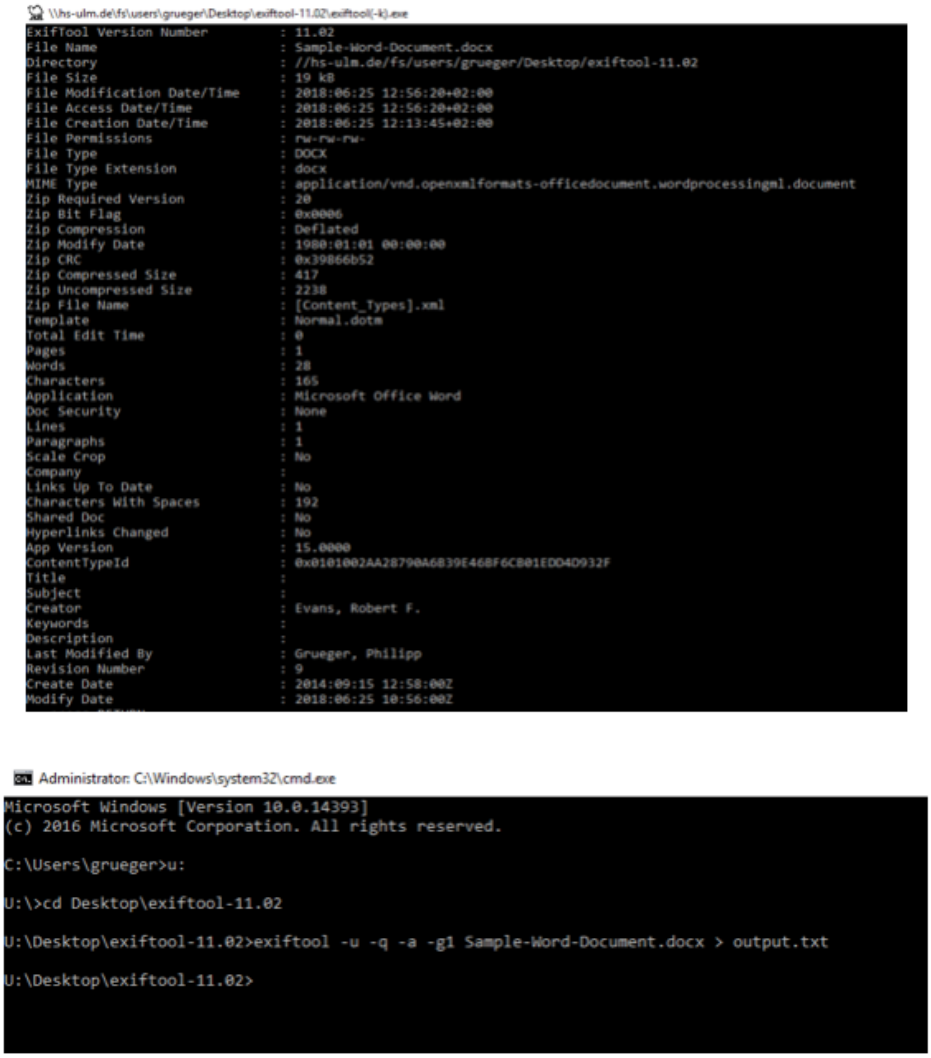


Figure 2.79: output saved in output.txt

there are informations to the options, advanced options and a huge list of commands, which are used to get the max out of this tool:

output.txt - Notepad

File Edit Format View Help

```
---- ExifTool ----
ExifTool Version Number         : 11.02
---- System ----
File Name                       : Sample-Word-Document.docx
Directory                       : .
File Size                       : 19 kB
File Modification Date/Time     : 2018:06:25 12:56:20+02:00
File Access Date/Time           : 2018:06:25 12:56:20+02:00
File Creation Date/Time         : 2018:06:25 12:13:45+02:00
File Permissions                : rw-rw-rw-
---- File ----
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlformats-officedocument.wordprocessingml.document
---- ZIP ----
Zip Required Version            : 20
Zip Bit Flag                    : 0x0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0x39866b52
Zip Compressed Size             : 417
Zip Uncompressed Size           : 2238
Zip File Name                   : [Content_Types].xml
Zip Required Version            : 20
Zip Bit Flag                    : 0x0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0x057e5599
```

Tag Name	Writable	Values / Notes
AppVersion	no	
Application	no	
Category	no	
Characters	no	
CharactersWithSpaces	no	
CheckedBy	no	
Client	no	
Company	no	
CreateDate	no	
DateCompleted	no	
Department	no	
Destination	no	
Disposition	no	
Division	no	
DocSecurity	no	0 = None 1 = Password protected 2 = Read-only recommended 4 = Read-only enforced 8 = Locked for annotations
DocumentNumber	no	
Editor	no	
ForwardTo	no	
Group	no	
HeadingPairs	no	
HiddenSlides	no	
HyperlinkBase	no	
HyperlinksChanged	no	'false' = No 'true' = Yes
Keywords	no	
Language	no	
LastModifiedBy	no	
LastPrinted	no	
Lines	no	
LinksUpToDate	no	'false' = No 'true' = Yes
MMClips	no	
Mailstop	no	
Manager	no	
Matter	no	
ModifyDate	no	
Notes	no	
Office	no	
Owner	no	
Pages	no	
Paragraphs	no	
PresentationFormat	no	
Project	no	
Publisher	no	
Purpose	no	
ReceivedFrom	no	
RecordedBy	no	
RecordedDate	no	
Reference	no	
RevisionNumber	no	
ScaleCrop	no	'false' = No 'true' = Yes
SharedDoc	no	'false' = No 'true' = Yes
Slides	no	
Source	no	
Status	no	
TelephoneNumber	no	
Template	no	
TitlesOfParts	no	
TotalEditTime	no	
Typist	no	
Words	no	

Figure 2.80: list of extractable data

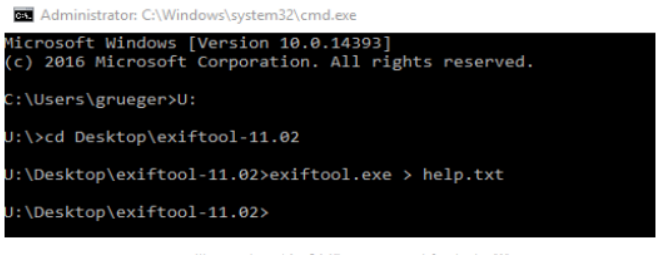


Figure 2.81: CMD command for help

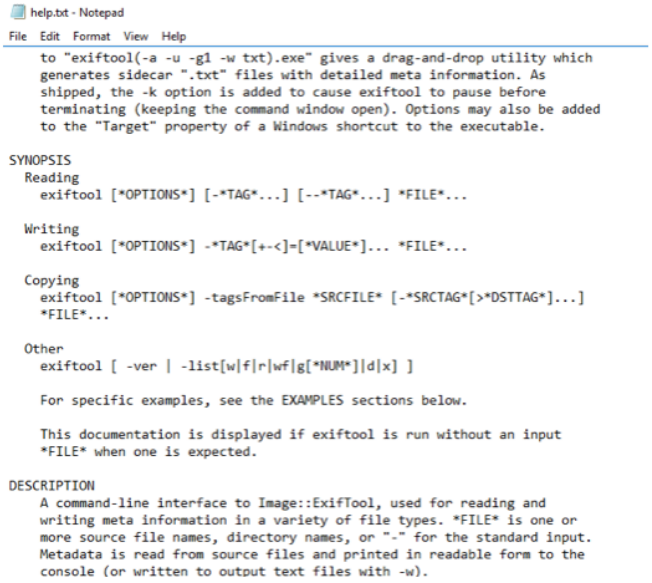


Figure 2.82: the content of the help.txt

2.11.4 Can metadata be deleted?

There are possibilities to ensure that Office applications not to create and store the metadata. If there is no data to save, so the is no metadata to extract. In Microsoft Office itself you are able to configure the settings in this way, that nearly no data is created and stored in the docx-file. This technique seem to solve the stored informations issues but in practice there are still a couple of fields which MS Office will create and store, even with all configured user settings. To be able to delete such informations about the user you have to work with the ?Document Inspector?. Go to the File-tab,

and then click Info. Click on check for issues and then on inspect document. After inspecting the file, you can see which data was actually saved and remove it from the docx-file.

The second technique is to delete created and stored metadata. In this case you need an application or utility tool to scrub these hidden informations. There are tools available such as iScrub by BigHand. Unfortunately most of the tools are not freeware. Since Office 2007 most of the tools does not work for the older version anymore.The tool DocScrubber was used to delete metadata, but DocScrubber only supports doc-files. Therefore it is recommended to use the Microsoft Office included feature Document Inspector, which was mentioned before.

2.12 RAM Imaging

2.12.1 Introduction

RAM imaging is a process in which the contents of the RAM are copied bit-by-bit (in a similar way the hard disk is copied into an image)[1]. It is an important tool in digital forensics, as it gives the forensics investigator an extra evidence registering tool. However, this extra tool also adds an extra complexity dimension: now the investigator has two choices upon gaining access to the target device - either force an abrupt removal of power by disconnecting its power source (so that the target device is left in the more preserved state in contrast with turning it off the expected way), or interact with the system without turning it off.

The RAM is volatile memory (that is, it is not expected that it will preserve its state after a shutdown), so in order to create a usable image of the RAM, the target device must not be turned off. This memory dump of the RAM can be used to later analyse what the target machine?s user was doing at the moment of device takeover.

Analysing this RAM image is better than analysing the target device RAM directly, as performing many actions overwrites potential evidence in memory akin to creating new files on a suspect hard disk drive.

The decision of creating a memory dump before turning off the target device, against shutting it down as soon as control is gained, is not to be taken lightly and depends on the objective and variables of the case. This is because, although creating a RAM image effectively preserves most of the

contents of the RAM, it can still trigger certain actions that overwrite or damage potential evidence in the device?s hard disk. Therefore, a conscious decision must be taken ideally before the device takeover act.

2.12.2 Why copy RAM ?

There are several reasons that a complete RAM capture may prove useful; most revolve around key differences between data stored in RAM and data stored on a hard disk drive.

- Volatile memory, e.g., RAM, is perceived to be more trusted than non-volatile memory (ROM or magnetic memory, for instance). This means, data that is stored usually stored encrypted in the hard drive, when loaded in memory it will most likely not be protected. This includes: passwords, financial transaction information, encryption keys, etc.

- Malware can residing completely in memory. In such a situation, the malware may not ever even touch the hard disk drive. This means that after removing the power from the target device, no record of the malware would exist.

- Memory is latent/dormant. Similar to how the recovery of deleted files became a widespread act early in the field of digital forensics, the recovery of prior (deleted) processes has become a focus of current research in memory forensics. For example, cached files may be stored in memory (and maybe never written to disk), thus making a RAM image more useful.

- Evidence. Whether a malware was executed or not, can be proved using a memory image. This is due to the fact that if the malware runs, it has to leave a footprint in the ram (although of course, it?s not safe to assume that it will be recoverable 100 percent of the time).

- Malware analysis. Executed code must exist somewhere in executable form, and sometimes it is better to analyse it from the RAM image. For instance, a packed executable (which is, it binary obfuscated) are hard to understand; however, in some situations, the unpacked version of the binary could be retrieved directly from memory, making the malware analysis process much easier.

Although under most circumstances the act of copying RAM will be shown to have a negative impact to potential evidence, the impact should be outweighed by potential gain. This gain can be achieved with good procedures and documentation, which in turn will minimise the effect of potential damage to evidence.

2.12.3 Main Objective

The main objective of a RAM imaging process, is to create an ideally identical image of the RAM at the time of target device takeover, so that it can be later analyzed without risk from that passive memory dump.

Therefore, the RAM imaging process can be divided into two separate areas:

- Creating the RAM image
- Analyse the image

In this document we will be performing both steps in different situations.

2.12.4 Tools Used

The following tools/systems were used to perform the examples:

For memory acquisition and simple analysis

- FTK Imager
- DumpIt
- Belkasoft

For cold boot attack

- RMPrepUSB
- Bios_memimage-1.2.tar.gz
- Target device cold boot: Intel i386 (32 bit) architecture, 4GB RAM, Windows 7
- Cold spray (-45 degrees celsius) and compressed air can (duster)

For advanced RAM image analysis

- Volatility
- Foremost
- Pdfid.py

2.12.5 Save RAM image and simple analysis

Process

In preparation to a memory acquisition, we would prepare a USB stick with enough storage capacity to hold up a minimum of the double the size of the RAM and containing the tool that would be executed to obtain an image of the memory. There are many tools available for memory imaging of which we tested the three mentioned above.

1. The first test was using FTK Imager. This software contains a very easy to use interface which requires two clicks after its installation.

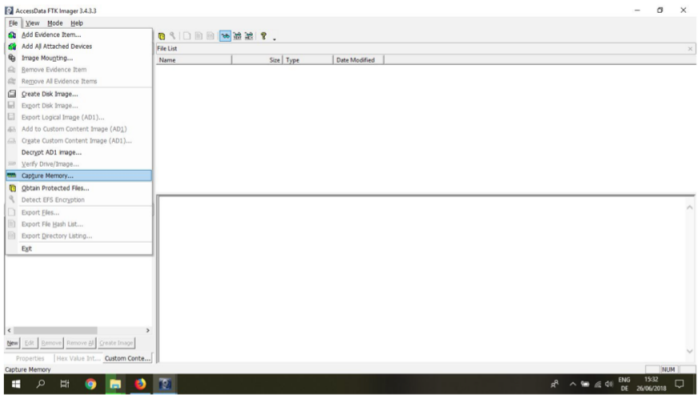


Figure 2.83: Selecting memory capture

2. The second test was using DumpIt. This is a portable software that can be run from a USB stick, and starts with one question, it provides the possibility of choosing between .DMP or .RAW and define the saving path of the image.

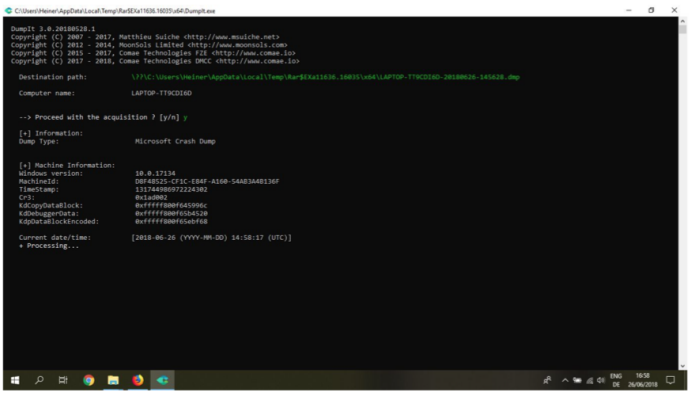


Figure 2.84: Memory captures initiated

3. The third software we tested was Belkasoft. This software can also be run from a USB stick and has a simple interaction to input the storage path and unlike most memory acquisition softwares, it runs in kernel-mode, which allows bypassing anti-debugging protection.

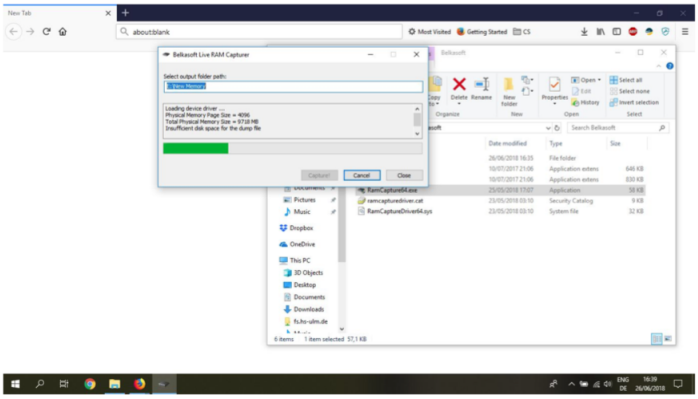


Figure 2.85: Memory capture after selecting the storage path

Additionally, Belkasoft offers the option to analyze the saved RAM image. Including the detection of artifacts, document formats, emails, registry files and so on.

Cold Boot Feasibility: Test Performed

Ordinary RAM typically lose their contents gradually over a period of seconds [5], even if the memory sticks are removed from the motherboard, and data will persist for minutes (or even hours), if they are kept at low temperatures.

Here we present the results of a relevant test done by J. Alex Halderman et al. [5].

Memory chips used:

	Memory Type	Chip Maker	Memory Density	Make/Model	Year
A	SDRAM	Infineon	128Mb	Dell Dimension 4100	1999
B	DDR	Samsung	512Mb	Toshiba Portégé	2001
C	DDR	Micron	256Mb	Dell Inspiron 5100	2003
D	DDR2	Infineon	512Mb	IBM T43p	2006
E	DDR2	Elpida	512Mb	IBM x60	2007
F	DDR2	Samsung	512Mb	Lenovo 3000 N100	2007

Figure 2.86: Test systems used

Memory decay in percentage after n seconds for each machine:

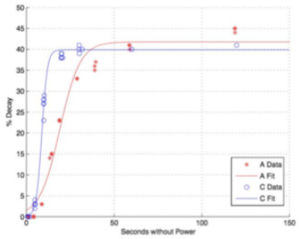


Figure 1: Machines A and C

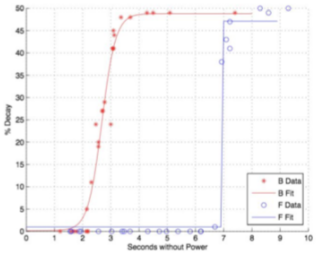


Figure 2: Machines B and F

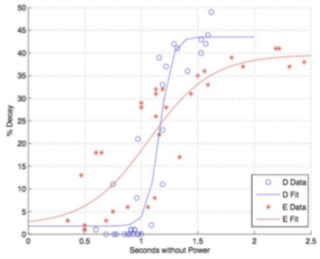


Figure 2.87: Test systems used

After cooling the RAM sticks to a temperature around 50 Celsius, the error percentage is relatively negligible, as shown in the following table:

	Seconds w/o power	Error % at operating temp.	Error % at −50°C
A	60	41	(no errors)
	300	50	0.000095
B	360	50	(no errors)
	600	50	0.000036
C	120	41	0.00105
	360	42	0.00144
D	40	50	0.025
	80	50	0.18

Figure 2.88: Effect of cooling on error rates

These tests confirm that cooling the RAM is indeed a viable method to

preserve volatile memory. However, the tests still show a decay percentage after an extended

2.12.6 Hack B: Cold Boot Attack

Goal

In this particular example, we will analyse a computer with Windows from an employee that reported strange activity in her bank account. The employee also reported receiving an email with a file attached from a fellow coworker, that upon opening, nothing opened, but her account got logged off and couldn?t log in again. The reports were made little time after these events happened.

To reach our goal, we will divide the process into two steps (detailed in this demo section and in the following one): Create an image from the PC using the physical cold boot method (for demonstration purposes, we will use an training image obtained in [6]). Analyse the image obtained in order to understand what happened, and ideally identify the culprit file for later analysis.

Process

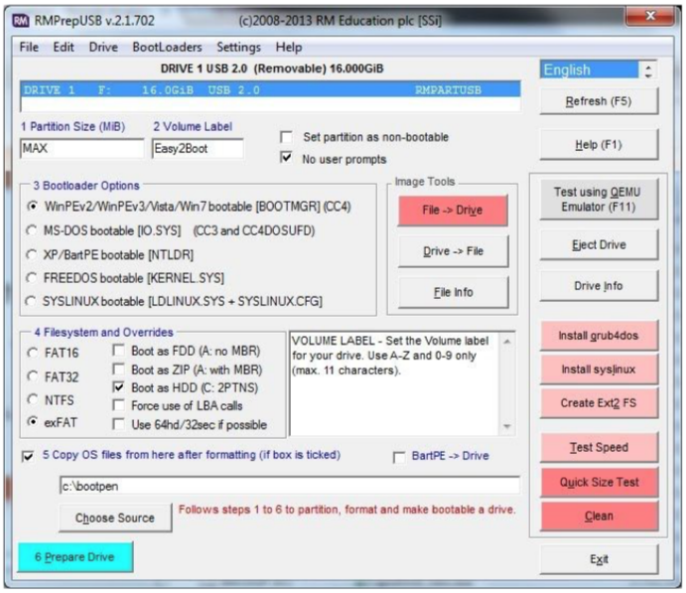
We will first create an USB stick that will be loaded when the target device is restarted, and will create a copy of the preserved cold RAM.

For this step, we can use plenty of different tools (described in section 9. Extra tools); we will use a tool called RMPrepUSB [3], which has the advantage of being able to copy a .bin to a specific sector of the USB, which, coupled with the correct RAM memory scraper file, will use a minimal amount of space in the target device?s RAM (thus better preserving its state).

We will also use the scraper called bios_memimage-1.2.tar.gz (from source [3]). After extracting and compiling it for a 32 bit architecture (the same as the target device), we proceed to save it into the USB stick (NOTE: the previous contents of the USB stick will be destroyed in the process).

(Select the drive in RMPrepUSB and use the File-¿Drive button, select the scraper.bin file and a USB start and File start address of 0 and length of 0).

After that, we should have a FAT16 formatted USB drive with scraper.bin as the boot code.



We can now perform the cold boot method on the target device:

- With the PC still turned on, we open the PC in a way so that the RAM memory is exposed and accessible.
- We start cooling the RAM with the cold spray, trying to do it in a way so that all sections of the ram sticks are evenly cooled. This should be done for around 5 to 10 seconds.
- We unplug the PC.
- We insert the USB stick into the PC.
- We plug the PC again, and turn it on.
- We continue to cool down the RAM sticks.
- After this, the USB stick will begin copying the memory into itself. This process duration depends on both the amount of RAM on the device, and the speed of the file transfer to the USB stick. In this case, as the target device had 4GB of RAM and its USB port is USB2.0, the process lasted approximately 50 minutes.

After this, we should have an image of the RAM in the USB stick. This completes the first step of our goal. We can now proceed with the analysis of the RAM image.

Remarks

If the boot order of the system is unknown, it is necessary to modify this setting in the BIOS of the computer while the RAM is being cooled. However, this adds an extra relatively big step, and could influence the quality of the final image.

If the target device?s specifications were unknown, a better approach would be to use a controlled system, in which the RAM sticks of the target device are cold-removed and re-inserted into the controlled device for imaging.

2.12.7 Extra hack C: Advanced Memory Analysis

Goal

With the RAM image successfully obtained from the target device, we can now begin our analysis. Our goal is to define what is happening in the target device, and to find a potential culprit file. For this, we will use Ubuntu as the main operating system, and the programs Volatility, strings and pdf-parser (all referenced in section 3.0 Tools used).

Process

For the first step, we will list the processes that were running on the victim?s PC in order to know which process was most likely responsible for the initial exploit. Using Volatility?s pslists command, we can do that:

Here, we can see all the processes that the victim was using at the moment of imaging. Highlighted, we can see that the process ?AcroRd32.exe? was running, and was started by PPID 888, which is firefox.exe. We know that the victim opened a file from her email, so it is plausible that the file was a PDF file, opened from Firefox.

Next, we can list the connections and sockets that were open on the victim?s machine, so that we can see if there is any suspicious processes that have sockets open.

```
Patricios-MacBook-Pro-189:DIFO preller$ ./volatility  pslist -f pcmem.vmem
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
0x823c8830 System                    4      0     58      573 ------      0
0x81f04228 smss.exe                548      4      3       21 ------      0 2010-02-26 03:34:02 UTC+0000
0x822eeda0 csrss.exe               612    548     12      423      0      0 2010-02-26 03:34:04 UTC+0000
0x81e5b2e8 winlogon.exe            644    548     21      521      0      0 2010-02-26 03:34:04 UTC+0000
0x82256da0 services.exe            688    644     16      293      0      0 2010-02-26 03:34:05 UTC+0000
0x82129da0 lsass.exe               700    644     22      416      0      0 2010-02-26 03:34:06 UTC+0000
0x81d3f020 vmacthlp.exe            852    688      1       35      0      0 2010-02-26 03:34:06 UTC+0000
0x82266870 svchost.exe             880    688     28      340      0      0 2010-02-26 03:34:07 UTC+0000
0x822e1da0 svchost.exe             948    688     10      276      0      0 2010-02-26 03:34:07 UTC+0000
0x822ea020 svchost.exe            1040    688     83     1515      0      0 2010-02-26 03:34:07 UTC+0000
0x81dea020 svchost.exe            1100    688      6       96      0      0 2010-02-26 03:34:07 UTC+0000
0x81de55f0 svchost.exe            1244    688     19      239      0      0 2010-02-26 03:34:08 UTC+0000
0x81dde568 spoolsv.exe            1460    688     11      129      0      0 2010-02-26 03:34:10 UTC+0000
0x821018b0 vmtoolsd.exe           1628    688      5      220      0      0 2010-02-26 03:34:25 UTC+0000
0x81ddd8d0 VMUpgradeHelper        1836    688      4      108      0      0 2010-02-26 03:34:34 UTC+0000
0x820d6b88 alg.exe                2024    688      7      130      0      0 2010-02-26 03:34:35 UTC+0000
0x81cdd790 explorer.exe           1756   1660     14      345      0      0 2010-02-26 03:34:38 UTC+0000
0x81ca96f0 VMwareTray.exe         1108   1756      1       59      0      0 2010-02-26 03:34:39 UTC+0000
0x820cd5c8 VMwareUser.exe         1116   1756      4      179      0      0 2010-02-26 03:34:39 UTC+0000
0x81cee5f8 wscntfy.exe            1132   1040      1       38      0      0 2010-02-26 03:34:40 UTC+0000
0x82333620 msiexec.exe             244    688      5      181      0      0 2010-02-26 03:46:06 UTC+0000
0x81ce1af8 msiexec.exe             452    244      0 --------      0      0 2010-02-26 03:46:07 UTC+0000
0x81c80c78 wuauclt.exe             440   1040      8      188      0      0 2010-02-27 19:48:49 UTC+0000
0x8221a020 wuauclt.exe             232   1040      4      136      0      0 2010-02-27 19:49:11 UTC+0000
0x82068020 firefox.exe             888   1756      9      172      0      0 2010-02-27 20:11:53 UTC+0000
0x820618c8 AcroRd32.exe           1752    888      8      184      0      0 2010-02-27 20:12:23 UTC+0000
0x82209640 svchost.exe            1384    688      9      101      0      0 2010-02-27 20:12:36 UTC+0000
Patricios-MacBook-Pro-189:DIFO preller$
```

```
Patricios-MacBook-Pro-189:DIFO preller$ ./volatility connections -f pcmem.vmem
Volatility Foundation Volatility Framework 2.6
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x81c6a9f0 192.168.0.176:1176       212.150.164.203:80       888
0x82123008 192.168.0.176:1184       193.104.22.71:80         880
0x81cd4270 192.168.0.176:2869       192.168.0.1:30379        1244
0x81e41108 127.0.0.1:1168           127.0.0.1:1169           888
0x8206ac58 127.0.0.1:1169           127.0.0.1:1168           888
0x82108890 192.168.0.176:1178       212.150.164.203:80       1752
0x82210440 192.168.0.176:1185       193.104.22.71:80         880
0x8207ac58 192.168.0.176:1171       66.249.90.104:80         888
0x81cef808 192.168.0.176:2869       192.168.0.1:30380        4
0x81cc57c0 192.168.0.176:1189       192.168.0.1:9393         1244
0x8205a448 192.168.0.176:1172       66.249.91.104:80         888
Patricios-MacBook-Pro-189:DIFO preller$
```

```
0x81c96b98     1752   1178      6 TCP            0.0.0.0         2010-02-27 20:12:32 UTC+0000
```

Here we can see the connections that were opened at the time of imaging. Highlighted we can see that the process id 1752 (Acrobat Reader) has a suspicious connection to the remote address 212.150.164.203. Also, that same process has an open socket assigned on port 1178.

Doing a quick ip lookup, we see that the IP address is registered under the name NetVision, in Israel.

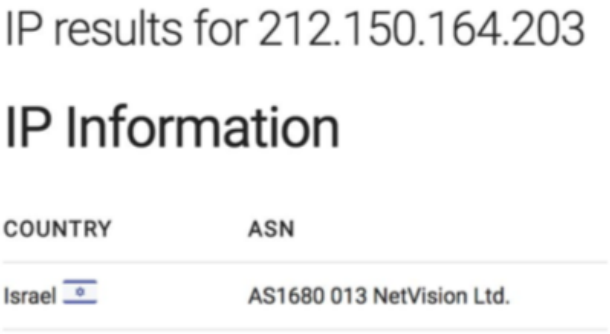
This confirms our initial suspicion of the Acrobat Reader process. We can now try to find the original PDF file.

Dump memory of the initial process (Acrobat Reader).

We can now use foremost [4] to extract the PDF files:

This will create a directory, in which the following PDF files are located:

Six of the eight files weight less than 500B, so they must be broken PDFs;




```
Patricios-MacBook-Pro-189:DIFO preller$ ./volatility memdump -p 1752 -f pcmem.vmem --dump-dir=./dump
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing AcroRd32.exe [  1752] to 1752.dmp
Patricios-MacBook-Pro-189:DIFO preller$
```

```
Patricios-MacBook-Pro-189:pdf preller$ foremost -i 1752.dmp -o pid1752
```

```
Patricios-MacBook-Pro-189:pdf preller$ ls -lah
total 1352
drwxr-xr--   9 preller  staff   288B Jun 26 15:06 .
drwxr-xr--  14 preller  staff   448B Jun 26 15:06 ..
-rw-r--r--   1 preller  staff   419B Jun 26 15:06 00445397.pdf
-rw-r--r--   1 preller  staff   419B Jun 26 15:06 00446730.pdf
-rw-r--r--   1 preller  staff   425B Jun 26 15:06 00579981.pdf
-rw-r--r--   1 preller  staff   425B Jun 26 15:06 00585184.pdf
-rw-r--r--   1 preller  staff   425B Jun 26 15:06 00600544.pdf
-rw-r--r--   1 preller  staff    59K Jun 26 15:06 00600928.pdf
-rw-r--r--   1 preller  staff   593K Jun 26 15:06 00601560.pdf
Patricios-MacBook-Pro-189:pdf preller$
```

however, there are two interesting files: 00600928.pdf, and 00601560.pdf, that weight around 60KB and 600KB, respectively.

We should now analyse both files using pdfid.py [4], to see if there is anything interesting:?

The first file turned out to be encrypted, but nothing really interesting is there.

However, the bigger file 00601560.pdf, is really interesting:

It has JavaScript code in it. This is certainly not usual in a normal PDF file; therefore, this file is now our prime suspect, since it most likely had malicious code in it.??Analysing the file is out of the scope of our study; however, as next steps, it can be further investigated by parsing the PDF into a readable file, and later studying the JavaScript code in order to know

```
Patricios-MacBook-Pro-189:pdf preller$ ./pdfid.py 00600928.pdf
PDFiD 0.0.11b 00600928.pdf
 PDF Header: %PDF-1.4
 obj                   104
 endobj                104
 stream                 34
 endstream              34
 xref                    2
 trailer                 2
 startxref               2
 /Page                   8
 /Encrypt                1
 /ObjStm                 0
 /JS                     0
 /JavaScript             0
 /AA                     0
 /OpenAction             0
 /AcroForm               0
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /Colors > 2^24          0
```

```
Patricios-MacBook-Pro-189:pdf preller$ ./pdfid.py 00601560.pdf
PDFiD 0.0.11b 00601560.pdf
 PDF Header: %PDF-1.3
 obj                     6
 endobj                  6
 stream                  1
 endstream               1
 xref                    2
 trailer                 2
 startxref               1
 /Page                   1
 /Encrypt                0
 /ObjStm                 0
 /JS                     1
 /JavaScript             1
 /AA                     1
 /OpenAction             0
 /AcroForm               0
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /Colors > 2^24          0

Patricios-MacBook-Pro-189:pdf preller$
```

which vulnerability was used, and hopefully, a perpetrators address.

2.12.8 Conclusion

Data proceeding from RAM can provide an additional depth and broadness on information concerning the system state of the device at the moment of the acquisition. As result, despite the minimal negative impact on the integrity of evidence a lot can be gained during the analysis.

2.13 PDF Malware analysis

2.13.1 Introduction

Manipulated word documents are very popular with criminals to infect computers with their malware. The fact that even Pdf files can contain executable code is often forgotten. Many users still believe that Pdf files are basically harmless. This lack of knowledge is increasingly used by attackers today to spread their malware in a perhaps unexpected way. Therefore it is appropriate to consider one of the most commonly used document formats which is the Portable Document Format (Pdf).

Nowadays, Pdf-documents may even contain interactive elements JavaScript, three dimensional objects and video content (Rich Media pdf) which provides ideal conditions for malware to hide. A Pdf file is similar to an archive, it contains different Pdf-objects which are describing the corresponding document and are arranged in a COS object tree Carousel Object Structure.

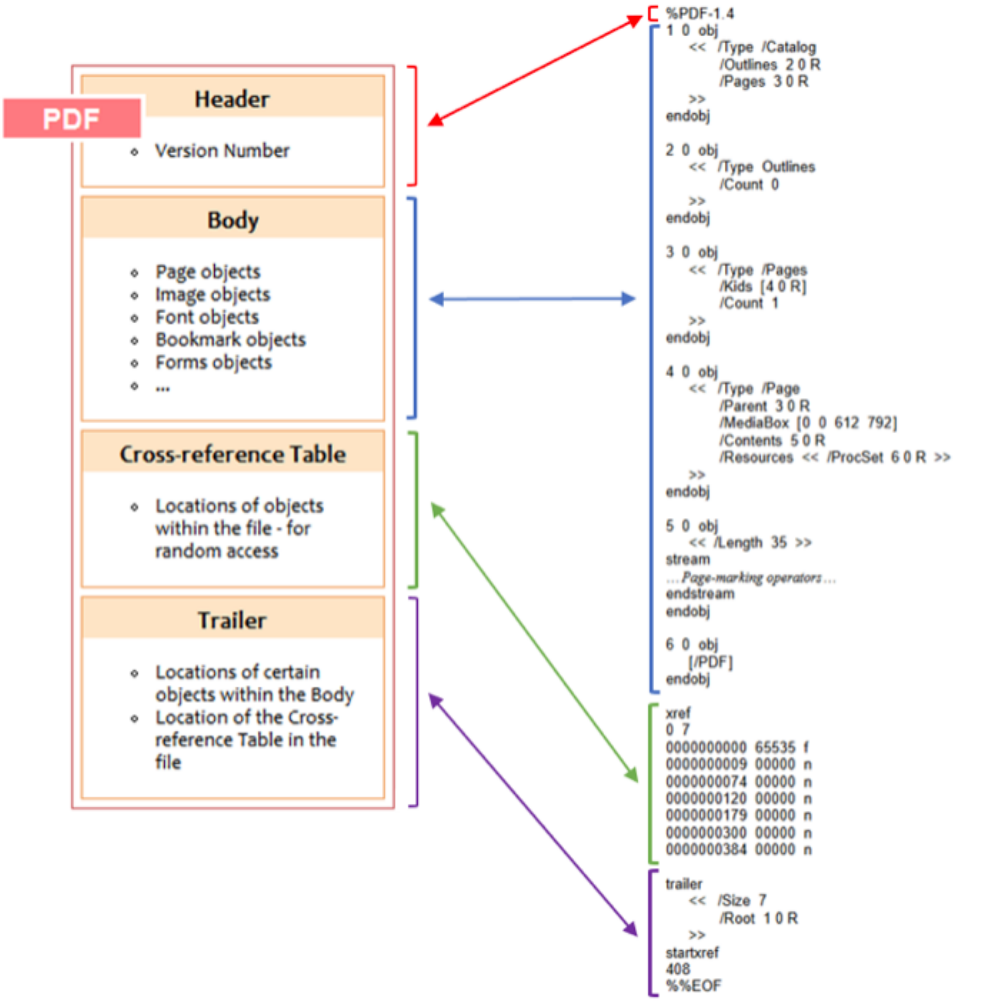
The malicious Pdf file usually contains an exploit. After opening the suspicious Pdf document, the exploited code runs and then other files can be executed it or could also trigger downloading files.

During every forensic investigation it has to be clarified, how the system was infected with the malware. Furthermore, as a digital forensic specialist it is advantageous to know as much different ways of potential infection scenarios as possible, since security incidents need to be reconstructed and retraced. This hack deals with the examination of Pdf documents to evaluate their content and the risk behind them.

2.13.2 PDF file structure

Before we take a closer look into the analysis of PDF documents it is useful to understand the structure behind. Therefore this section provides a brief introduction of the Portable Document Format (PDF). A PDF file consists of 4 elements:

Header The first section of the file is the header and consists normally of two lines. The first line specifies the PDF version number of the file and is mandatory. This allows applications to determine if they are able to process the file or not.



The second line contains some non-printable characters, which are usually used to tell applications, that the file contains binary data and should not be treated as ASCII text.

Body The body of a PDF File contains the indirect objects that compose the content of a document. For the forensic investigation, this section provides an important source of information about the content of a PDF document. The contained objects allows to draw a conclusion if a document is compromised with malicious content. A closer look on these objects can

be found in the section Adobe PDF Objects.

For the beginning we are focusing on the objects that are mandatory for a PDF-Document:

- The Catalog dictionary represents the root of a document. It contains references to other objects that defines the document.
- This part describes the page tree, which defines the ordering of pages in the document.
- The leaves of the Page Tree are called Page Objects. They are specifying the attributes of a single page.
- Consists of a sequence of bytes containing the content of an object.

Cross-reference Table The next section is the cross-reference table, that contains the references to all the objects in the document. The big advantage here is that it allows access to random objects in a file. This means that it?s not necessary to read a whole PDF file to locate a certain object. This table is initiated with the tag xref, followed by two numeric values. The first value 0 indicates the root of the body in the document. The next value indicates the number of objects in the body.

Example xref 0 4 0000000000 65535 f 0000000021 00000 n 0000000086 00000 n 0000000195 00000 n Afterwards we can find a 20 byte long entry for each of this objects. This information can be easily broken down into the following components:

10-digit byte offset to the object from the beginning oft he document

5-digit generation number

Entry type: n = in use, f = free

Each entry is followed by a linefeed, to start a new line.

Trailer The last section is the trailer, that contains a link to the cross-reference table of the document and starts with the line trailer. The trailer must contain at least two entries:

Root the root entry contains an indirect reference (object number) of the Catalog dictionary. This allows a reader to quickly find the cross-reference table and other objects.

Size the size entry specifies the total amount of entries in the file?s cross-reference table. It?s important to mention here, that PDF readers should read PDF files from it?s end.

Example trailer << /Size 7 /Root 1 0 R >> startxref 408 EOF

Before the end of the file there are two line with a string ?startxref? and a number. This entries define an offset (in our case 408 Byte) from the beginning of the file to the cross-reference table of the document, that starts with ?xref?. Finally, the line ?

2.13.3 Adobe PDF objects

A PDF documents is a data structure that contains different objects that allows a wide range of functionalities to enable the user the creation of complex and dynamic documents. In principle a good idea, but these allows cybercriminals to integrate JavaScript code into pdf documents or specify malicious actions that run automatically. The Table below represents a general overview about risky PDF Format tags.

PDF Objects	Action
/OpenAction /AA	Specify the script or action to run automatically
/JavaScript /JS	Specify JavaScript to run
/GoTo	Changes the view to a specified destination within the PDF or in another PDF File
/Launch	Launch a program or document
/URI	Access a resource by it's URL
/SubmitForm /GoToR	Send data to URL
/RichMedia	Can be used for embedded Flash content
/ObjStm	Can hide objects inside an Object Stream
Hexcodes like /J#61vaScript	Used to mask information

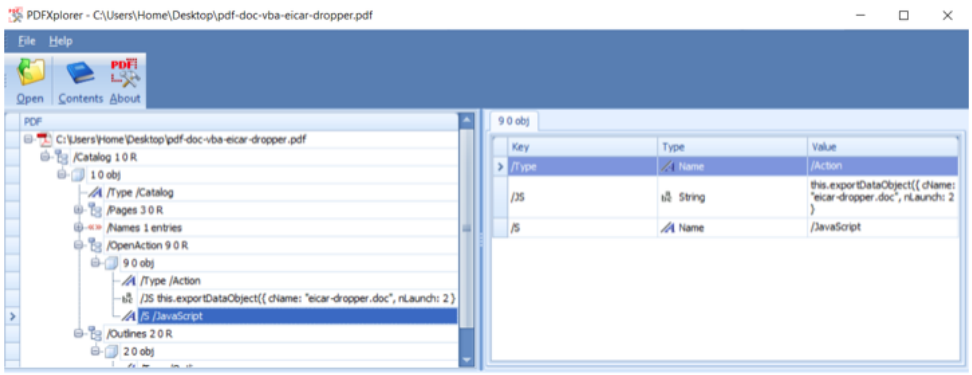
Regrettably since the PDF specification version 1.2, it?s allowed to replace characters with ist hexadecimal ASCII Code. As can be seen below, the same code seems unreadable and can be very for a manual analysis. Fortunately,

there are some programs that automatically check PDF documents for there objects, which are able to find these hidden objects.

```
/#54#79#70#65 /#41#63#74#69#6f#6e
/#53 /#4a#61#76#61#53#63#72#69#70#74
/#4a#53(#28#74#68#69#73#2e#65#78#70#6f#72#74#44#61#74#61#4f#62#
6a#65#63#74#28#7b#20#63#4e#61#6d#65#3a#20#22#65#69#63#61#72#2d
#64#72#6f#70#70#65#72#2e#64#6f#63#22#2c#20#6e#4c#61#75#6e#63#68
#3a#20#32#20#7d#29#3b#29
```

2.13.4 PDFXplorer

Sometimes it can be useful to view the internal structure of the PDF files in order to understand the objects of the PDF file and their relationships. For this purpose we can use the Tool PdfXplorer to represent the structure in a Tree.



2.13.5 Malware Analysis

Tools

- Python Script peepdf: https://github.com/jesparza/peepdf (pre-installed on kali)

- Python Script make-pdf: https://blog.didierstevens.com/programs/pdf-tools/
- Python Script pdf-parser: https://blog.didierstevens.com/programs/pdf-tools/
- Malicious testfile: https://blog.didierstevens.com/2015/08/28/test-file-pdf-with-embedded-doc-dropping-eicar/

A malicious PDF document usually contains an exploit. When the file is opened, the exploited code runs and then other files are dropped and executed. Moreover, files may be downloaded from the internet or other malware could be copied from the document onto the computer.

For demonstrating on how to perform a PDF malware analysis, we use a malicious but harmless PDF file at first, which was prepared by Didier Stevens. (https://blog.didierstevens.com/2015/08/28/test-file-pdf-with-embedded-doc-dropping-eicar) The EICAR Institute (European Institute for Computer Antivirus Research) developed many different malware files for testing purposes of e.g. anti-virus-systems.

The PDF file we use in the first point (malicious_testfile.pdf), contains JavaScript which extracts and opens a DOC file. This DOC file contains a VBA script which will be executed when the DOC file has been opened and which writes an EICAR test file (log) into the temporary TEMP directory.

After downloading und unzipping the test file, we get a notification from our anti-virus system that a potential threat has been recognised. Some anti-malware programs would remove the file directly into a protected quarantine area, where we can decide if we want to keep the document or if it should be deleted. For this case we can dismiss the warning and keep the file.

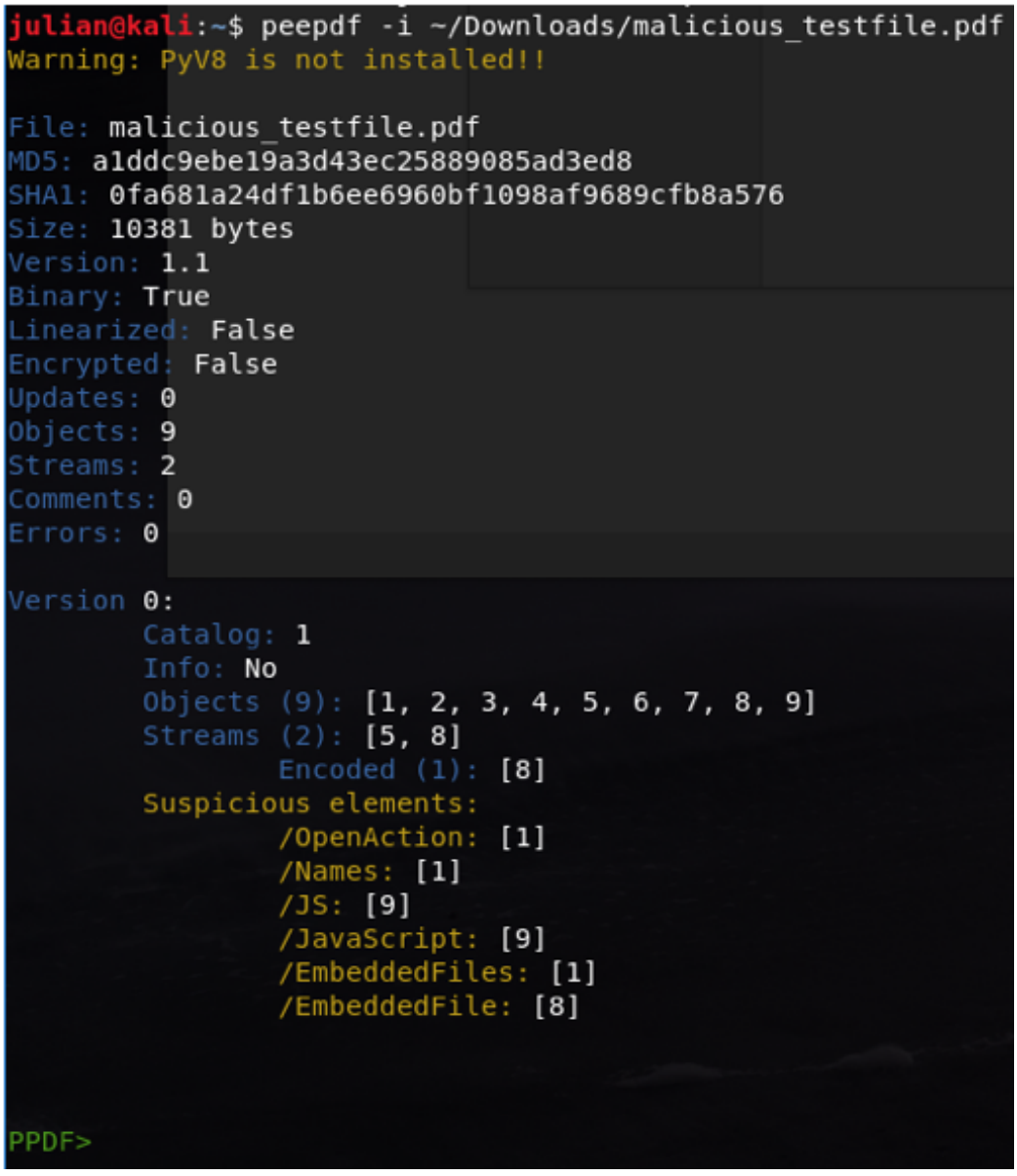
For using the python scripts as analysing tools, there must a python interpreter be installed (often pre-installed). We can check for our installed python version via following command:

```
python -V
```

If we verified a python interpreter to be available and after downloading the peepdf script from the link above in the Tools section, we can start the script together with the PDF file that should be analysed as a parameter, via:

```
peepdf -i ~/Desktop/malicious_testfile.pdf
```

With the -i option we induce the ?peepdf interactive console to launch, where we can later use various commands for analysing the file.



We receive an overview of the inspected file and as we can already see, suspicious objects have been found, in particular object [9] which is a JavaScript Object. Below at the green prompt, we can now enter different commands into the peepdf interactive console to give us more information about the file. Subsequently some useful commands:

Command	Result
Tree	shows PDF file structure as a tree
metadata	search file for metadata
Info	shows initial overview again
object 9	inspect object [9]
info 9	further description of object [9]
js_analyse 9	analyze JavaScript object [9]
Help	list of all commands

For a quick overview of a suspicious Pdf file, we can use peepdf with following command and find suspicious objects inside the suspicious$_e$$lementstag$.

```
peepdf -x ~/Desktop/malicious_testfile.pdf
```

With the -x option we specify XML format for displaying information about the PDF file.

For further analysing the JavaScript code of object [9] with peepdf, the js_analyse-command requires to have PyV8 installed, which acts like a bridge between Python and JavaScript objects and which is not installed on Kali by default. For reproducing reasons and to introduce other tools, we use the python-script ?pdf-parser? for further analysing, in particular JavaScript objects. Before we start analysing with the pdf-parser, we create our own malicious testfile with a javascript code. Therefore we use the make-pdf tool like following:

Alternatively we could also use, specially for other embedded script-files from different languages, the following command:

For testing purposes we created a very simple JavaScript file which simply shows an alert box stating that the victim just got hacked:

YOU JUST GOT HACKED !!

Now we can verify if our manipulated pdf document works properly by opening it. Since Kali Linux is blocking the execution of the embedded javascript code, we sent the pdf document to a windows machine and disabled the malware scanner there.

```
<suspicious_elements>
  <triggers>
    <trigger name="/OpenAction">
      <container_object id="1"/>
    </trigger>
    <trigger name="/Names">
      <container_object id="1"/>
    </trigger>
  </triggers>
  <actions>
    <action name="/JS">
      <container_object id="9"/>
    </action>
    <action name="/JavaScript">
      <container_object id="9"/>
    </action>
  </actions>
  <elements>
    <element name="/EmbeddedFiles">
      <container_object id="1"/>
    </element>
    <element name="/EmbeddedFile">
      <container_object id="8"/>
    </element>
  </elements>
</suspicious_elements>
```

```
python make-pdf-javascript.py [options] pdf-file
```

Option	Result
-h or –help	help page
-j or --javascript=	embed javascript code
-f or --javascriptfile=	embed javascript file

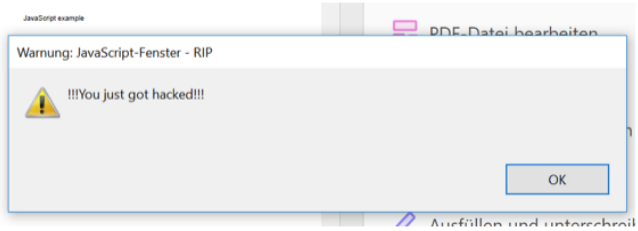
```
julian@kali:~$ python make-pdf-javascript.py -f ./testcode.js ./test.pdf
```

Since we now successfully created our malicious pdf document we can proceed with the malware analyses using the pdf-parser. The tool parses through the file without rendering it and thus no code from /OpenAction or /AA objects could be automatically executed. Find below in the subsequent table a collection of the most useful options for the pdf-parser-tool.

```
python make-pdf-embedded.py [options] file-to-embed pdf-file
```

Option	Result
-h or –help	help page
-a or –autoopen	open the embedded file automatically when the pdf is opened
-b or –button	add a button to launch the embedded file
-m or --message=	Text to display in the pdf document

```
julian@kali:~$ cat testcode.js
app.alert({cMsg: '!!!You just got hacked!!!', cTitle: 'RIP', nIcon: 1});
```



Option	Result
-h or –help	help page
-s or --search=	search for strings inside objects (not case-sensitive)
-o or --object=	select an object by its id (e.g. -o 9)
-w or –raw	raw output for data
-a or –stats	display stats for pdf document
-t or --type=	select an object by its type (e.g. --type=/OpenAction)

It is recommendable to use the –stats option first of all, to classify the pdf document and to get a rough overview of the contained objects. Unexpected or unusual objects can be identified here at the first time. Frequently, pdf files have almost identical stats, although they are completely different in their content and origin.

If we execute the command with our self-manipulated test file we get the following output:

As we can see there is one object with a /Action tag which indicates that there could be some code execution done within the pdf document. With the -s option we can search for string inside objects (not case-sensitive) that are of our interest. Useful search terms are openaction, javascript, aa, richmedia among many.

```
julian@kali:~$ pdf-parser --stats test.pdf
Comment: 2
XREF: 1
Trailer: 1
StartXref: 1
Indirect object: 7
  1: 5
 /Action 1: 7
 /Catalog 1: 1
 /Font 1: 6
 /Outlines 1: 2
 /Page 1: 4
 /Pages 1: 3
```

```
julian@kali:~$ pdf-parser -s openaction test.pdf
obj 1 0
 Type: /Catalog
 Referencing: 2 0 R, 3 0 R, 7 0 R

  <<
    /Type /Catalog
    /Outlines 2 0 R
    /Pages 3 0 R
    /OpenAction 7 0 R
  >>
```

As we can see there is object 1 (which is the /Catalog and by that the root of the document) where we should be aware of. It contains an /OpenAction tag referencing to object 7. We remember, /OpenAction and /AA indicate an automatic action that is performed when the pdf file is rendered (when pdf file is opened). Especially the combination of /OpenAction (/AA) and JavaScript objects (/JS, /JavaScript) makes a pdf file suspicious since it is a very common attack vector. To search for JavaScript-objects in particular, we use the command like following:

```
julian@kali:~$ pdf-parser -s javascript test.pdf
obj 7 0
 Type: /Action
 Referencing:

  <<
    /Type /Action
    /S /JavaScript
    /JS "(app.alert({cMsg: '!!!You just got hacked!!!', cTitle: 'RIP', nIcon: 1}
);\n\n)"
  >>
```

But since we already know that the root object is referencing to object 7 we can directly inspect this object which gets us the same results:

```
julian@kali:~$ pdf-parser -o 7 test.pdf
obj 7 0
 Type: /Action
 Referencing:

  <<
    /Type /Action
    /S /JavaScript
    /JS "(app.alert({cMsg: '!!!You just got hacked!!!', cTitle: 'RIP', nIcon: 1}
);\n\n)"
  >>
```

According to the above results, object 7 is an Action object containing javascript code which is exactly the script we embedded previously. With the following command we can find out which other objects are referencing to this JavaScript-object. This is especially interesting since it causes the code to run automatically if it is a /AA or /OpenAction object referencing to the /JavaScript-object.

```
julian@kali:~$ pdf-parser --reference  7 test.pdf
obj 1 0
 Type: /Catalog
 Referencing: 2 0 R, 3 0 R, 7 0 R

  <<
    /Type /Catalog
    /Outlines 2 0 R
    /Pages 3 0 R
    /OpenAction 7 0 R
  >>
```

In this case it is only the first root object (obj 1). The tool peepdf that was used at first, outputs the same results:

```
PPDF> tree

/Catalog (1)
        /Pages (3)
                /Page (4)
                        /Pages (3)
                        stream (5)
                        /Font (6)
        /Action /JavaScript (7)
        /Outlines (2)
```

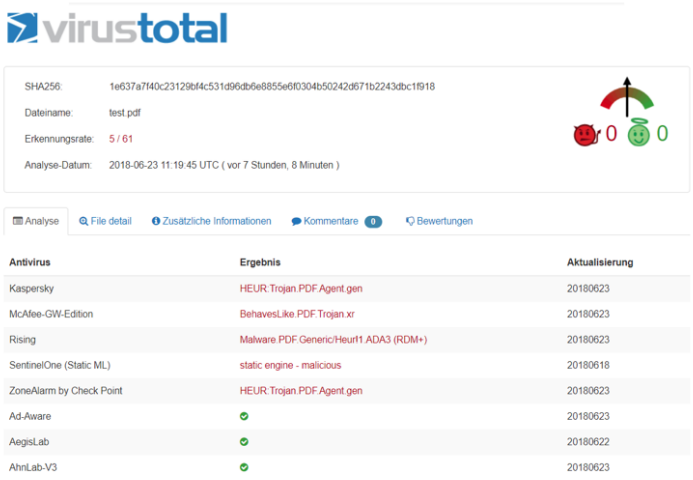
2.13.6 Further Malware Analysis

As a forensic expert, it is very important to always have your field set, tools and other forensic equipments with you. As an expert we come across so much crap stuff which is not useful for us and to filter this crap out takes much of our time. For this we have many softwares which make our work very easy and filtered. As not every expert can afford to have a whole laboratrary with equipments and thus these softwares works wonder for them.

For Example: We have a PDF with some virus in it, normally to find out the type of virus we have to go through number of steps and this will take our time. Software like ?Virus Total? will do this work in minutes or seconds and will give us a list of possible viruses.

Virus Total This software is available on the internet and it is absolutely free (https://www.virustotal.com), as an expert you can use software to analyse any pdf for viruses. You can upload the particluar pdf you want to analyse online and within seconds it will display all the kinds of viruses it has detected. It has 40 different Anti-Viruses which analyse the particular pdf for viruses. The advantage is you do not need to install any packages or other additional softwares as in case of other PDF analysing softwares. You can also send your pdf on an email (scan@virustotal.com) provided by the

company which will analyse your pdf and send you the results.



As we can see in the above figure, in total 5 out of 61 antivirus programs listed on VirusTotal recognised our own prepared pdf document as malware.

There are many available softwares which provides us with a detailed report of the error and viruses that can be present in the pdf file. Some are:

- Anubis (http://forensikhacks.de/anubis)
- CWSandbox (http://forensikhacks.de/mwa)
- MobileSandbox (http://forensikhacks.de/msb)
- Lastline (https://www.lastline.com/)
- Vmray Analyzer (https://www.vmray.com/products/malware-sandbox-products/)

All these softwares or so called sandboxes provides an online overall security for your pdf and other files included with threat intelligence, advanced malware detection and breach protection.

Anubis: It provides with an service for online malware analysing. You can upload any windows program or URL and Anubis will provide you with a report of the file in different formats like HTML, XML, PDF, Text etc.

In the report there will details about the internal processes between the Windows registries and the file as well as there would be a detailed information about the network activities. Also, it provides details about types of detected viruses and malware. These details helps to find more about the other possibilities of attacks or the thinking of the attacker. It give us also binary data and with this we can see what a particular type of malware is going to do in your computer.

Tip:: As there are many softwares available, it always very useful to try most of them as every time you will find something interesting on analysing the same pdf.

Nowadays, there are many viruses being developed for smartphones as well and so that malware analysing softwares as well. In particular MobileSandbox (http://forensikhacks.de/msb) provides us with every details associated with mobile worms and viruses and as a sandbox provides us a detailed description of the malware.

It is very useful to use all these online softwares as it makes our work less and gives us a detailed information about the particular malware. These softwares have mostly all of the anti-virus softwares which will detect different types of malwares and thus save time and money.

2.14 References

RAM Imaging :: 11. References [1] https://www.tandfonline.com/doi/full/10.1080/1556
RMPrepUSB https://www.rmprepusb.com/
$Bios_memimage-1.2.tar.gzhttps://github.com/DonnchaC/coldbootattacks$
Volatility https://www.volatilityfoundation.org/
Foremost https://tools.kali.org/forensics/foremost
Pdfid.py https://github.com/DidierStevens/DidierStevensSuite/blob/master/pdfid.py
[4] https://citp.princeton.edu/research/memory/
[5] https://github.com/DonnchaC/coldboot-attacks/blob/master/coldboot.pdf
[6] https://github.com/volatilityfoundation/volatility/wiki/Memory-Samples
[7] $https://www.forensicswiki.org/wiki/Tools:Memory_Imaging$
[8]https://www.hackers-arise.com/single-post/2016/09/27/Digital-Forensics-Part-2-Live-Memory-Acquisition-and-Analysis

[9] https://users.ece.cmu.edu/ tvidas/papers/JDFP06.pdf

[10] https://www.forensicmag.com/article/2011/06/memory-forensics-where-start

Word Documents : Sources:

[1] url: http://isyou.info/jowua/papers/jowua-v2n4-4.pdf title: Hiding Information into OOXML Documents: New Steganographic Perspectives author: Castiglione, Aniello and D'Alessio, Bonaventura and De Santis, Alfredo and Palmieri, Francesco created: 2011 last accessed : 25th June 2018

[2] author: Philipp Akharath, Philipp Grger title: own documentation, imagery and illustrations created: June 25th 2018

[3] url: https://www.sno.phy.queensu.ca/ phil/exiftool/ title: ExifTool by Phil Harvey author: Phil Harvey last accessed: 25th June 2018

- https://wiki.sleuthkit.org/index.php?title=Mmls

- Lorenz Kuhlee Victor Vlzow : Computer Forensik Hacks; Hacks 56 - 61

- http://manpages.ubuntu.com/manpages/trusty/man8/fdisk.8.html

- http://manpages.ubuntu.com/manpages/xenial/en/man8/netstat.8.html

- https://developer.apple.com/legacy/library/documentation/Darwin/Reference

- https://linux.die.net/man/1/dpkg

- http://ftp.rpm.org/max-rpm/rpm.8.html

- http://man7.org/linux/man-pages/man1/apropos.1.html

- https://linux.die.net/man/8/ifconfig

- https://www.ietf.org/rfc/rfc2131.txt

- Lorenz Kuhlee, Victor Vlzow. Computer Forensik Hacks. O'Reilly, 2012.

- Oechslin, Philippe (2003-08-17). Making a Faster Cryptanalytical Time-Memory Trade-Off

- "Rainbow Tables for ophcrack". Ophcrack.sourceforge.net. Retrieved 2018-04-14.

- www.iso.org

- Photo credits: https://makeameme.org/meme/missiles-missiles-everywhere https://memegenerator.net/instance/18160338/

[1] Alternate Data Streams in NTFS. https://blogs.technet.microsoft.com/askcore/2013/03/24

[2] Everything I know about NTFS. http://www.kes.talktalk.net/ntfs/

[3] The Deep Scan option. https://www.ccleaner.com/docs/recuva/ using-recuva/wizard-mode/the-deep-sca