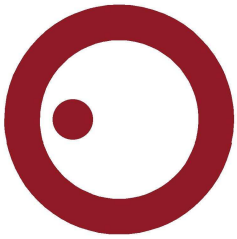


escola superior de tecnologia e gestão
instituto politécnico de leiria

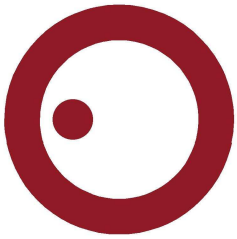
Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

AUTOPSY – ENHANCED DISTRIBUTED FORENSIC ANALYSIS

PEDRO HENRIQUE GASPAR CORDEIRO FERREIRA

Leiria, Fevereiro de 2020

[November 15, 2019 at 17:27 – version 1.0]



escola superior de tecnologia e gestão
instituto politécnico de leiria

Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

AUTOPSY – ENHANCED DISTRIBUTED FORENSIC ANALYSIS

PEDRO HENRIQUE GASPAR CORDEIRO FERREIRA
Número: 2180078

Relatório de estágio realizado sob orientação da Professora Doutora Marisa da Silva Maximiano (marisa.maximiano@ipleiria.pt).

Leiria, Fevereiro de 2020

[November 15, 2019 at 17:27 – version 1.0]

ACKNOWLEDGEMENTS

I would like to express appreciation for the opportunity provided by “VOID SOFTWARE, S.A.“ to work in an enterprise environment. I’d also like to thank my faculty advisor “Marisa da Silva Maximiano“ who provided much needed help writing this report in LaTeX 2_ε.I’m also thankful that Antonio Branco took time off his work to help me with so many minor details in my CSS layouts, as well as everyone one else in the company who provided me with everything I needed to succeed in my internship.

RESUMO

TODO.

A B S T R A C T

TODO.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 DIGITAL FORENSICS

1.1.1 *Contextualization*

Forensic science is the use of scientific methods or expertise to investigate crimes or examine evidence that might be presented in a court of law.

The definition of digital forensics is directly related to the definition of computer forensics which is the collection, preservation, analysis, and presentation (*Digital Forensics for Legal Professionals*) of evidence stemming from digital sources for use in a legal matter using investigative processes, tools, and practices.

Digital forensics is the application of computer technology to criminal cases where evidence includes items that are created by digital systems.

Digital forensics, according to NIST, is the field of forensic science that is concerned with retrieving, storing and analyzing electronic data that can be useful in criminal investigations. This includes information from computers, hard drives, mobile phones and other data storage devices.

Digital forensic investigators face challenges such as extracting data from damaged or destroyed devices, locating individual items of evidence among vast quantities of data, and ensuring that their methods capture data reliably without altering it in any way.

Personal data should ultimately be attributable to an individual; however, making that attribution can be difficult due to the presence or absence of individualized user accounts, security to protect those user accounts, and the actual placement of a person at the same location and time when the data is created.

1.1.2 *Digital Evidence*

Digital evidence is any type of digital data with incriminating characteristics, which can result from any type of action preformed by a user, like transactions, recordings, or virtually any action preformed on a device.

Nowadays it's virtually impossible not to leave a digital track behind, since most of us carry and use devices capable of connecting to the internet.

The explosion of social media sites has created a whole new area of electronic evidence. Most people today are willing to share all kinds of information through social media platforms.

In order for electronic data to become digital evidence, it must be stored and be recoverable by a forensic examiner. One of the great challenges is not whether digital evidence may exist, but where the evidence is stored, getting access to that storage, and finally, recovering and processing that digital evidence for relevance within a civil or criminal action.

The potential storage options for electronic evidence has shifted from being only contained locally to being either located locally or remotely in what is called "The Cloud".

More and more everyday computing processes are moving to the Internet where companies offer software as a service. Software as a service means that the customer no longer has to install software on their computer, allowing access to the software remotely, and not storing any data locally.

A PARTIR DAQUI TODO O CONTEUDO DA INTRODUCAO É COPIA INTEGRAL DE FONTES EXTERNAS

1.1.3 *Processes and Procedures*

Digital forensics is the application of forensic science to electronic evidence in a legal matter.

While there are many different subdisciplines and many types of devices, communication, and storage methods around today, the basic tenets of digital forensics apply to all of them.

These tenets encompass four areas:

1. Acquisition
2. Preservation
3. Analysis
4. Presentation

Each of these areas includes specific forensic processes and procedures.

1.1.3.1 *Acquisition*

Acquisition is the process of actually collecting electronic data. For example, seizing a computer at a crime scene or taking custody of a computer in a civil suit is part of the acquisition process. Making a forensic copy of a computer hard drive is also part of the acquisition process. In the digital forensics field, examiners refer to making these forensic copies of evidence as "acquiring" a hard drive rather than copying a hard drive. This is to avoid the confusion that could be caused by using the term "copy," since making a copy of something does not imply that the copy was made in a forensically sound manner.

Acquisition is the first step in the forensic process and is critical to ensure the integrity of the evidence. As acquisition is the first contact with the evidence, it is the point where evidence is most likely to be damaged or destroyed. Simply turning on a computer can lead to the modification of hundreds of evidentiary items including files, date and time stamps, introduction of new Internet history, and the destruction of files that could be recovered from areas of the hard drive that are in the area of unallocated space (see the section "Deleted File Recovery" in Chapter 29).

1.1.3.2 *Preservation*

As evidence is collected, it must be preserved in a state that is defendable in court. Preservation is the process of creating a chain of custody that begins prior to collection and ends when evidence is released to the owner or destroyed. Any break in the chain of custody can lead to questions about the validity of the evidence. Additionally, preservation includes keeping the evidence safe from intentional destruction by malicious persons or accidental modification by untrained personnel.

A chain of custody log best illustrates an example of preservation. Chain of custody logs should include every instance that a piece of evidence has been touched, including the initial collection of the device storing the evidence, the transport and

storage of the evidence, and any time the evidence is checked out for handling by forensic examiners or other personnel. At no time should there be a break in this chain.

1.1.3.3 *Analysis*

Analysis is the process of locating and collecting evidentiary items from evidence that has been collected in a case. In a case involving spousal infidelity, the evidence that must be located can include e-mails and chat logs between the spouse and the paramour. In a fraud case, financial records would be the target of the analysis, as well as the possible deletion of records involving financial transactions. In a child pornography case, locating contraband pictures and movies would be the target of the examination. Each case is unique in this respect as the circumstances surrounding each case can vary widely, not only in the evidence being sought, but also in the approach used to perform the analysis. The analysis portion is also the area where the individual skills, tools used, and the training of the forensic examiner have the greatest impact on the outcome of the examination. For information on forensic tools used by examiners, see Chapter 5. Considering that electronic evidence appears in so many forms and comes from so many disparate locations and devices, the training and experience of the examiner begins to have an ever-greater impact on the success of the examination.

The analysis phase is also where the greatest disparity begins to become a factor between the skills and approach of a "computer expert" and those of a computer or digital forensics expert. While a computer expert may understand many aspects of computer usage and data, a properly trained forensic expert will be well versed in recovering data as well as in proper examination techniques.

Analysis of digital evidence is more than just determining whether something like a file or e-mail message exists on a hard drive. It also includes finding out how that file or e-mail message got on the hard drive, and if possible, who put the file or message on the hard drive.

1.1.3.4 *Presentation*

Presentation of the examiner's findings is the last step in the process of forensic analysis of electronic evidence. This includes not only the written findings or forensic report, but also the creation of affidavits, depositions of experts, and court testimony. There are no hard and fast rules or standards for reporting the results

of an examination. Each agency or private entity may have its own particular guidelines for reporting. However, forensic examination reports should be written clearly, concisely, and accurately, explaining what was examined, the tools used for the examination, the processes used by the examiner, and the results of that examination. The report should also include the collection methods used, including specific steps taken to protect and preserve the original evidence and how the verification of the evidence was performed.

In general, a digital forensics report should include:

- Background and experience of the examiner
- Tools used in the examination
- Methods used to verify the data
- Processes used to recover and extract the data
- Statement of what the examiner found
- Actual data recovered to support the statement of findings.

1.2 THE SLEUTH KIT

The Sleuth Kit (TSK) is a library and collection of command line tools that allow you to investigate disk images. The core functionality of TSK allows you to analyze volume and file system data. The plug-in framework allows you to incorporate additional modules to analyze file contents and build automated systems. The library can be incorporated into larger digital forensics tools and the command line tools can be directly used to find evidence.

Description

The original part of Sleuth Kit is a C library and collection of command line file and volume system forensic analysis tools. The file system tools allow you to examine file systems of a suspect computer in a non-intrusive fashion. Because the tools do not rely on the operating system to process the file systems, deleted and hidden content is shown. It runs on Windows and Unix platforms.

The volume system (media management) tools allow you to examine the layout of disks and other media. The Sleuth Kit supports DOS partitions, BSD partitions (disk labels), Mac partitions, Sun slices (Volume Table of Contents), and GPT disks.

With these tools, you can identify where partitions are located and extract them so that they can be analyzed with file system analysis tools.

When performing a complete analysis of a system, we all know that command line tools can become tedious. Autopsy is a graphical interface to the tools in The Sleuth Kit, which allows you to more easily conduct an investigation. Autopsy provides case management, image integrity, keyword searching, and other automated operations.

A complete analysis also requires more than just file and volume system analysis. However, a single tool can't provide support for all file types and analysis techniques. The TSK Framework allows tool so easily incorporate file analysis modules that were written by other developers. If you are developing a tool, consider incorporating in the framework or developing your analysis technique as a module into the framework.

Input Data

Analyzes raw (i.e. dd), Expert Witness (i.e. EnCase) and AFF file system and disk images. Supports the NTFS, FAT, ExFAT, UFS 1, UFS 2, EXT2FS, EXT3FS, Ext4, HFS, ISO 9660, and YAFFS2 file systems (even when the host operating system does not or has a different endian ordering). Tools can be run on a live Windows or UNIX system during Incident Response. These tools will show files that have been "hidden" by rootkits and will not modify the A-Time of files that are viewed.

Search Techniques

List allocated and deleted ASCII and Unicode file names. Display the details and contents of all NTFS attributes (including all Alternate Data Streams). Display file system and meta-data structure details. Create time lines of file activity, which can be imported into a spread sheet to create graphs and reports. Lookup file hashes in a hash database, such as the NIST NSRL, Hash Keeper, and custom databases that have been created with the 'md5sum' tool. Organize files based on their type (for example all executables, jpegs, and documents are separated). Pages of thumbnails can be made of graphic images for quick analysis.

The Sleuth Kit is written in C and Perl and uses some code and design from The Coroner's Toolkit (TCT). The Sleuth Kit has been tested on:

- Linux
- Mac OS X
- Windows (Visual Studio and mingw)
- CYGWIN

- Open & FreeBSD
- Solaris

1.3 AUTOPSY

Autopsy is a digital forensics platform and graphical interface to The Sleuth Kit and other digital forensics tools. It is used by law enforcement, military, and corporate examiners to investigate what happened on a computer. You can even use it to recover photos from your camera's memory card.

Easy to Use Autopsy was designed to be intuitive out of the box. Installation is easy and wizards guide you through every step. All results are found in a single tree. See the intuitive page for more details.

Extensible Autopsy was designed to be an end-to-end platform with modules that come with it out of the box and others that are available from third-parties. Some of the modules provide:

- Timeline Analysis - Advanced graphical event viewing interface (video tutorial included).
- Hash Filtering - Flag known bad files and ignore known good.
- Keyword Search - Indexed keyword search to find files that mention relevant terms.
- Web Artifacts - Extract history, bookmarks, and cookies from Firefox, Chrome, and IE.
- Data Carving - Recover deleted files from unallocated space using PhotoRec
- Multimedia - Extract EXIF from pictures and watch videos.
- Indicators of Compromise - Scan a computer using STIX.
- See the Features page for more details. Developers should refer to the module development page for details on building modules.

Fast Everyone wants results yesterday. Autopsy runs background tasks in parallel using multiple cores and provides results to you as soon as they are found. It may take hours to fully search the drive, but you will know in minutes if your keywords were found in the user's home folder. See the fast results page for more details.

Cost Effective Autopsy is free. As budgets are decreasing, cost effective digital forensics solutions are essential. Autopsy offers the same core features as other digital forensics tools and offers other essential features, such as web artifact analysis and registry analysis, that other commercial tools do not provide.

1.4 ALTERNATIVE SOFTWARE

1.4.1 *nuix*

1.4.2 *OSForensics*

1.5 PROPOSED SOLUTION

Why is the project being developed???

Este documento serve de orientação para o relatório da unidade curricular de Projecto Informático do Curso de Engenharia Informática da ESTG – IPLEIRIA. Como tal, é constituído por um conjunto predefinido de estilos a utilizar. Estes estilos devem ser utilizados sem serem alterados ou substituídos. Para começar facilmente a escrever o relatório, basta guardar uma cópia deste documento e substituir os campos e as secções de acordo com o projecto em questão.

Embora possa parecer uma abordagem demasiadamente descritiva para a escrita do relatório, as intenções pretendidas com este documento são:

- Focar os alunos na produção de conteúdos com qualidade, em vez de se preocuparem com formatações de tipos de letra, parágrafos, etc.;
- Ao fornecer um documento de orientação de estilos a Escola beneficia de um aspecto profissional e consistente da globalidade dos seus relatórios de projecto.

Quanto ao conteúdo de uma introdução, ele deve preparar o leitor para o resto do relatório. Deve conter o detalhe suficiente para que alguém das áreas de conhecimento envolvidas possa entender o assunto do trabalho. A maior parte das introduções contêm três partes para fornecer contexto ao trabalho: objectivos, âmbito e background do trabalho do projecto. Estas partes muitas vezes sobrepõem-se, e podem por vezes ser omitidas simplesmente porque não faz sentido incluir alguma delas.

É de extrema importância considerar os objectivos do trabalho e do relatório na introdução. Se os autores não entenderem bem os objectivos do trabalho, dificilmente o leitor os entenderá. As seguintes questões ajudam a pensar nos objectivos do trabalho e na razão da escrita do relatório:

1. O que foi descoberto ou provado?
2. Em que tipos de problemas se trabalhou?
3. Porque é que se trabalhou nestes problemas? Se o problema lhe foi atribuído, deve tentar-se saber as razões pelas quais os orientadores o formularam, e o que era suposto que os alunos aprendessem ao trabalharem neste problema;
4. Qual a razão da escrita deste relatório?
5. O que é que o leitor deve ficar a saber quando acabar de ler este relatório?

O âmbito deve indicar as áreas de conhecimento envolvidas e realçar a metodologia utilizada no trabalho de projecto. Referir o âmbito do projecto na introdução ajuda o leitor a perceber os parâmetros de entrada do trabalho e do relatório, bem

como a identificar as principais restrições consideradas (por exemplo "existem 5 Sistemas Operativos para trabalhar com determinado hardware, mas somente 3 foram considerados neste estudo"). As seguintes questões ajudam a pensar no âmbito do trabalho e do relatório:

1. De que forma foi abordado o problema, e qual a razão para tal abordagem?
2. Existiam outras abordagens óbvias que se poderiam ter adoptado ? Que limitações impediram que se tentassem outras abordagens?
3. Que factores contribuíram para a escolha da forma de como se abordou o problema, e qual o mais relevante nessa escolha?

A informação de background inclui os conhecimentos que o leitor deve possuir por forma a compreender o trabalho de projecto e correspondente relatório. Estes conhecimentos incluem a percepção de trabalhos prévios que motivaram a proposta do projecto corrente, ou referências a trabalhos teóricos e práticos relacionados com os objectivos e âmbito descritos acima. Devem remeter-se para anexos documentos que poderão ajudar na percepção de teorias, metodologias, técnicas ou ferramentas utilizadas no trabalho de projecto. As seguintes questões ajudam a pensar no background necessário para o trabalho e para o relatório:

1. Que factos deve o leitor conhecer para perceber o relatório?
2. Porque é que o projecto foi autorizado ou atribuído?
3. Quem já fez trabalho prévio para resolver o problema colocado pelo projecto?

Por fim, a introdução deve descrever como foi organizado o relatório, referindo brevemente o propósito de cada secção considerada no mesmo.

O resto deste documento dá uma breve perspectiva das partes seguintes que devem constar do relatório, bem como de outros aspectos de formatação.

2

HOST ENTITY CHARACTERIZATION

Escrever aqui tudo o que é trabalho relacionado com o projeto a ser desenvolvido. Neste capítulo as referências bibliográficas são extremamente importantes e podem ser feitas da seguinte forma (ver código fonte do LaTeX):

Para fazer uma citação no fim de uma frase: (Sims, 1992). Multiplas citações (Darwin, 1859; Koza, 1992)

Para fazer uma citação que serve também como sujeito dessa frase (por exemplo no início): Sims (1992)

Obter apenas o nome do autor: Sims

Obter apenas o título do obra: «Interactive evolution of dynamical systems»

Segundo Rudolph (2016) isto assim assado, bla .... *The minted package: Highlighted source code in LaTeX 2ε*

fgdfgdf

- 1212
- dsafsdfds
- dsfdsf

sadsadsa

1. asdsad
2. sdfsfdsf
3. dsfdsfds

3

INTERNSHIP PROGRAMME

3.1 INTERNSHIP START

The internship started on September 2nd 2019, being planned to last for 9 months, ending on the 29th of May 2020.

The first day at the company was interesting, meeting most of the personalities of the workplace and getting set up with my work environment, consisting of a desk, chair, computer and 2 monitors.

As soon as the computer was properly configured with all the software needed the project became the main focus of my time at the company, making progress daily, sometimes just planning the architecture or analyzing the source code, sometimes just writing code, and sometimes a mix of both.

3.2 PROJECT PLANNING

Autopsy by itself is capable of providing a distributed solution for multi user collaboration, but it is very resource intensive and requires many complex configuration steps, and is also only fully supported on the Windows O.S.

The plan for this internship is to achieve the same kind of functionality provided by the original software, but without the dependency on harduous pre-configuration or hardware intensive requirements, resulting in needing only single capable server, and allowing the program to be used by multiple low capacity client devices using any web capable O.S.

To achieve that goal, the project will be an adaptation of the original Autopsy source code, into a server-client model, with the server developed in Java using the Quarkus framework, and the client developed in Javascript using the React framework.

The project is outlined to work in a multi user environment, allowing users to be assigned to teams and teams assigned to cases, and allowing multiple users to interact with a case simultaneously.

Autopsy has a major limitation, which is it only allows one case to be open at a time, ideally in this project we should find a workarround to allow working on multiple cases at once, but we feel like dedicating a server instance (which a single machine can have many virtualized within) per case is a good enough approach, though as future work the ability to spawn different containers as requested to work on different cases at once is an interesting challenge.

Given that the core features will be running in a remote server, it was decided that the addition of data sources to cases will be handled by an FTP client, allowing users to transfer image files to the respective data source directories of each case, and FTP access will be controlled according to each user's assigned cases.

3.3 AUTOPSY SOURCE CODE ANALYSIS

Autopsy is a digital forensics analysis software that is available as Open Source Software on GitHub.

With the goals set for this project, the source code was analyzed to understand which components need to be replicated and adapted in order to obtain the same logic flow.

The "Core" module is where the most important components are located, and after analysis it was concluded that the following directories contain relevant information:

- Actions: user interactions
- Casemodule: case class and other resources needed for the functioning of an autopsy case like data sources and artifacts
- Centralrepository: data persisted and accessed by multiple cases (Correlation Engine)
- Contentviewers: panels used for data representation
- Coordinationservice: configuration information distribution system
- Core: addition of command line options, system configurations and collaboration monitor
- Corecomponents: main user interface components

- Datamodel: all the entities needed to represent ingested data
- Datasourceprocessors: data processing utilities
- Directorytree: file explorer for ingested artifacts
- Ingest: utilities and events for data ingestion
- Keywordsearchservice: utility to search artifacts by keyword
- Modules: all the pre-included modules (data ingestion procedures)
- Progress: progress indicators and similar classes
- Python: resources needed for the functioning of the Jython language
- Rejview: resources used to analyse Windows registry
- Report: report generator
- Timeline: recent addition to Autopsy, allows visualization of artifacts in temporal chart, only available for Windows Operating System

The "KeywordSearch" module is also of critical importance as it provides one of the most meaningful features which is filtering all the artifacts in a case with a keyword search using the Apache Solr search platform, which indexes the text contents of all the artifacts and allows extremely fast searching through a large amount of data.

3.4 DEVELOPMENT

3.4.1 *Management Entities*

As a first step in the development, the different persisted entities were created, which are Users, Teams, and Cases. All the endpoints for actions involving these entities were created, resulting in the ability for the client program to interact with these entities and modify their relationships and other variables. For these functionalities there are two roles associated, the Manager role allows manipulation of the existing entities while the Investigator role only has access to his own information and the teams and cases he was assigned to. For these interactions, it was decided to create a drag and drop interface, which allows users to be dragged into teams and teams dragged into cases. All these entities are listed side by side and each have their own options and filtering input, as can be observed in Figure 1.

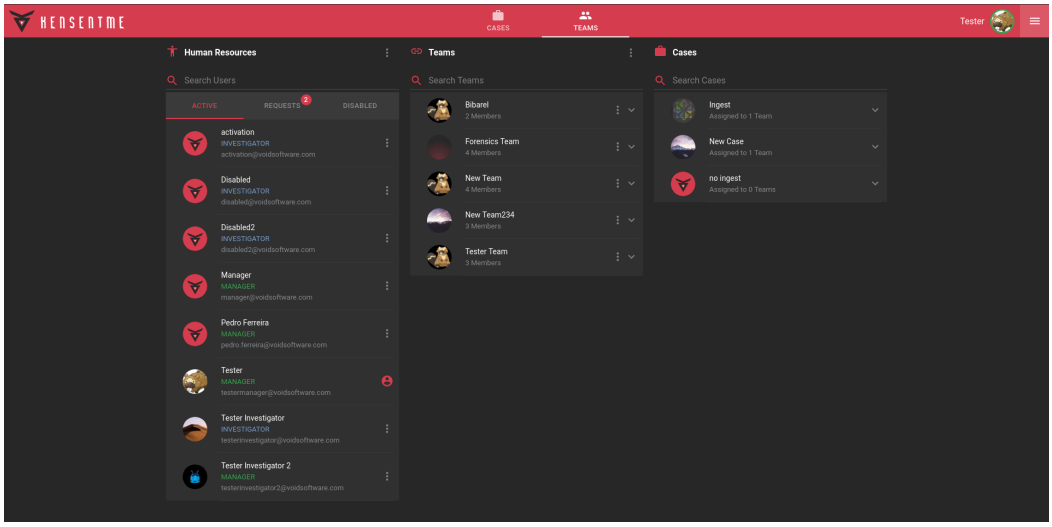


Figure 1: Entity Management Interface

Users can change their own profile picture, while Managers can also change any team or case's display picture. Managers can add new users to the platform, can approve membership requests, can enable/disable user accounts and can create new teams. When a user is added by a Manager or his membership request is approved he must define a password while activating his e-mail account.

3.4.2 *Basic Autopsy Functionalities*

Then the most basic functionality from Autopsy was adapted, the ability to open an autopsy case. For this some elements of the original Casemodule package were adapted, and after that all the other similar actions like closing, creating and deleting cases were also adapted.

Autopsy cases have a case file containing case metadata, which allows the program to connect to the right database when the case is open, this database is also present in a file inside the filesystem, which uses the SQLite database engine, so for the cases to be usable in the server these files must also be present in the server, which resulted in the creation of a directory within the server called "Server Repository", contaning all the different cases created within the application. This same directory will also be used to store data sources that are added to cases, which will be added through FTP protocol.

Later in the development there was the need to create an aditional directory inside the "Server Repository" called "Central Repository" which contains the database used by the Correlation Engine to ingest data that can be queried by any case.

3.4.3 *Ingested Results Presentation*

Ingested results are the items present inside the provided data sources, Autopsy runs multiple modules on each data source and extracts these results using the Sleuth Kit, extracted results can either be a file instance or an artifact (which is something that corresponds only to a piece of information inside a file).

The ingested results are presented in three different containers, one taking the shape of a file explorer, allowing exploration of the structure of all the results, one taking the shape of a table, presenting all the contents of the result selected from the explorer, and one taking the shape of a content viewer, allowing visualization of the data contained inside the result selected from the table, as can be seen in Figure 2.

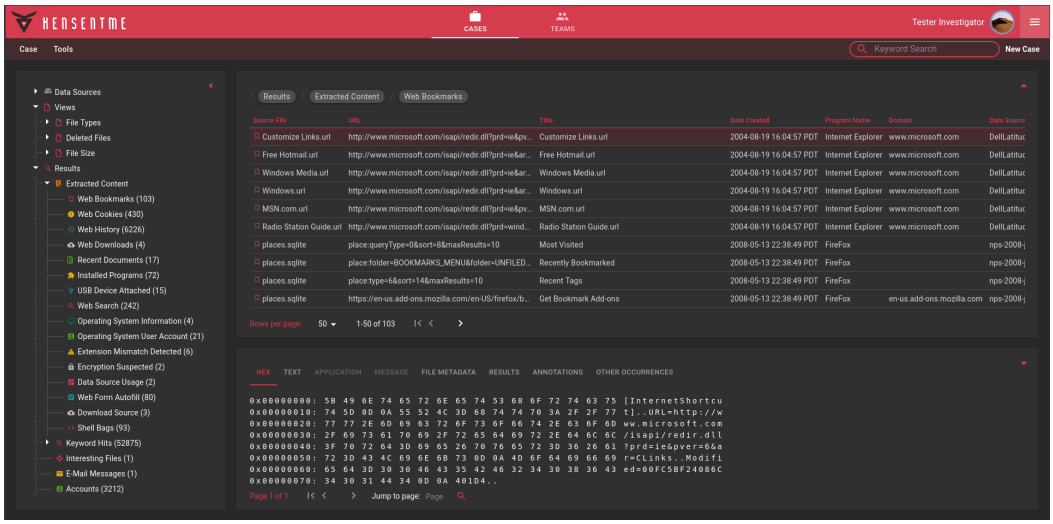


Figure 2: Ingested Results Presentation

The layout for the ingested results presentation, and for all the case related actions, was based on the original Autopsy layout, allowing each container to be resized as needed, allowing the user to focus on the information that is most important to him.

O corpo do relatório compõe, normalmente, a parte mais extensa do relatório, e contém todos os conteúdos necessários para que o leitor perceba o assunto do mesmo. Estes conteúdos incluem detalhes, dados, resultados de teste, factos e conclusões. O que incluir exactamente no corpo do relatório e como será organizado é determinado pelo contexto do trabalho desenvolvido. Geralmente, o corpo do relatório inclui 7 secções distintas:

1. Uma secção para teorias, modelos e hipóteses. Esta secção tem uma maior proeminência em artigos de investigação, onde é sugerida uma hipótese (contribuição) inovadora. Esta secção deve ser omitida para o caso de trabalhos mais práticos, cuja elaboração não origine uma contribuição inovadora, mas sim num produto de aplicação de tecnologias e metodologias;
2. Uma secção onde são discutidas as tecnologias, metodologias, ferramentas e técnicas utilizadas, e a forma como foram adequadas para se fazerem cumprir com os objectivos do trabalho. Algumas questões que esta secção deve procurar responder incluem:
   - Que equipamentos de hardware e ferramentas de software foram utilizados para o desenvolvimento do trabalho?
   - Qual a metodologia de desenvolvimento foi adoptada, e como é que ela se reflecte em termos de protótipos, modelos, diagramas, código, testes e documentação, de acordo com os objectivos do projecto? Sugere-se a utilização de exemplos no corpo do relatório, remetendo para anexos a descrição dos produtos intermédios completos;
   - Como foi planeado o trabalho, em termos de sequenciamento de actividades, recursos necessários, estimativas de tempo, e produtos intermédios., de acordo com a metodologia de desenvolvimento adoptada?
3. Uma secção na qual se apresentam e interpretam os resultados da elaboração do trabalho. A apresentação dos resultados finais do trabalho deve contrapor-se com os objectivos iniciais do projecto, e deve ser acompanhada de uma avaliação comprovada, por exemplo, através de testes elaborados e devidamente documentados. Deve também procurar-se quantificar o grau de satisfação dos requisitos do problema do projecto, através da exposição de funcionalidades não cumpridas ou cumpridas parcialmente (por exemplo, incluir uma lista de bugs de uma aplicação de software desenvolvida), bem como funcionalidades que extrapolam os objectivos iniciais do projecto;

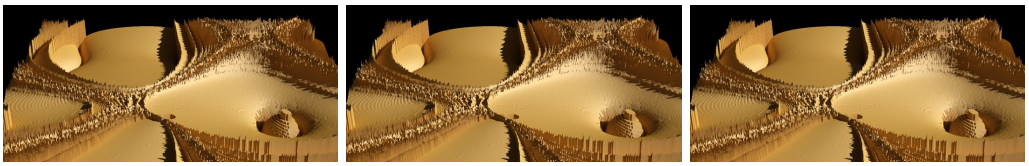


Figure 3: Imagem composta por três figuras em ficheiros separados

4. Uma secção de conclusões, onde são resumidos os principais resultados do trabalho e onde se usufrui de uma outra hipótese de expressar a sua qualidade/relevância através de um resumo conciso e coerente com o trabalho desenvolvido. É também o local onde se devem referir quais as principais forças e fraquezas do trabalho desenvolvido.

5. Uma secção de trabalho futuro, onde se devem propor possíveis desenvolvimentos futuros para colmatar as deficiências e lacunas identificadas atrás, ou simplesmente para evoluir o produto do trabalho desenvolvido;

6. Uma secção para referências bibliográficas, onde cada referência deve incluir, no mínimo, o nome dos seus autores, o título, data de publicação (ou de acesso, para o caso de URLs) e o tipo de documento (livro, artigo, website, etc.);

7. Uma secção para anexos, para a colocação dos produtos finais ou intermédios do projecto, por forma a não interromper a linha de desenvolvimento adoptada para a escrita da introdução e corpo do relatório. Deve ser utilizado um cabeçalho do estilo Heading 1 para identificar cada uma destas secções.

3.5 ESTILOS

O LaTeX 2$_\varepsilon$ trata da formação, apenas temos de usar as tags correctas. Seguem-se alguns exemplos. A Figura 3 é constituída por 3 imagens em ficheiros **.jpg** separados. A Figura 4 é constituída apenas por um ficheiro e ocupa 50% da largura de uma linha de texto.

Na Tabela 1 temos um exemplo de uma tabela onde existem linhas que ocupam mais do que uma linha da tabela.

As técnicas evolutivas baseiam-se em algoritmos bio-inspirados que aplicam a teoria de Darwin (Darwin, 1859). Esta defende a evolução natural das espécies onde os organismos vivos são recompensados, através da sobrevivência e da propagação dos seus próprios genes aos sucessores. Actualmente existem quatro classes principais de algoritmos evolutivos: Algoritmos Genéticos (AG) (Holland, 1975), Estratégias

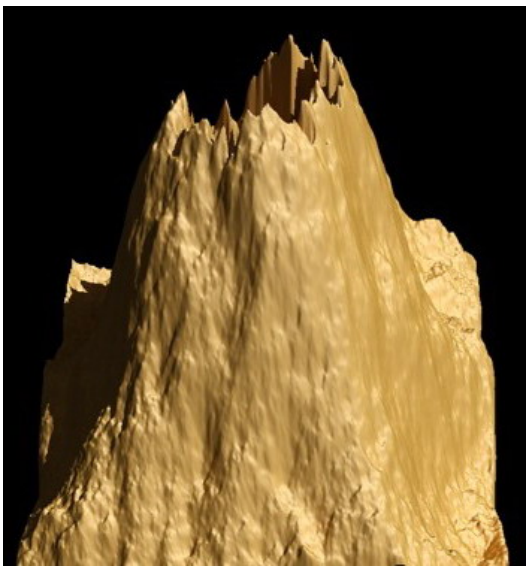


Figure 4: Vulcão

Table 1: Floating point benchmark. $\mathbf{R}_{max}$: the performance in Gflops for the largest problem run on a machine; $\mathbf{N}_{max}$: the size of the largest problem run on a machine; $\mathbf{N}_{1/2}$: the size where half the R_{max} execution rate is achieved; $\mathbf{R}_{peak}$: the theoretical peak performance in Gflops for the machine

Linpack Benchmark (Full precision)	Proc. or Cores	$\mathbf{R}_{max}$ GFlops	$\mathbf{N}_{max}$ Order	$\mathbf{N}_{1/2}$ Order	$\mathbf{R}_{peak}$ GFlops
Thinking Machine CM-5	32	1,900	9216	4096	4
Pentium 4 3.0 GHz	1	4,730	7600	365	6
IBM Cell BE 3.2 GHz	9	98,05	4096	1536	204,8 (32 bits) 14,6 (64 bits)

Evolutivas, Programação Genética (GP) (Koza, 1992) e Programação Evolutiva. Todos os algoritmos evolutivos mantêm uma população de soluções candidatas sobre a qual efectuam uma pesquisa para determinar os indivíduos mais fracos. De acordo com um determinado critério, estes são substituídos por outros gerados através de operadores aplicados aos melhores indivíduos da população, criando assim uma nova geração. Este processo é repetido sobre sucessivas gerações até se encontrar uma boa solução, que pode não ser a óptima.

Existem vários trabalhos e até video jogos que usam algoritmos evolutivos (Sims, 1992; Wikipedia, 2007).

3.6 INCLUIR CÓDIGO FONTE

Nas Listagens 1 e 3 temos um exemplo da inclusão de código fonte diretamente a partir do ficheiro fonte. Para mais informação ler o Manual da packge minted (Rudolph, 2016). Nestes exemplos a formação foi configurada no ficheiro `config.tex` (procurar por `minted`).

Listagem 1: Código fonte C com sintaxe colorida

```c
#include <stdio.h>
int main()
{
    int i, n, t1 = 0, t2 = 1, nextTerm = 0;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci Series: ");

    for (i = 1; i <= n; ++i)
    {
        // Prints the first two terms.
        if(i == 1)
        {
            printf("%d, ", t1);
            continue;
        }
        if(i == 2)
        {
            printf("%d, ", t2);
            continue;
        }
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
        printf("%d, ", nextTerm);
    }
    return 0;
}
```

Listagem 2: Código fonte Bash que ocupa mais que uma página

```
#!/bin/bash
# Simple line count example, using bash
#
# Bash tutorial:
↪ http://linuxconfig.org/Bash_scripting_Tutorial#8-2-read-file-into-bash-array
# My scripting link: http://www.macs.hw.ac.uk/~hwloidl/docs/index.html#scripting
#
# Usage: ./line_count.sh file
# --------------------------------------------------------------------------

# Link filedescriptor 10 with stdin
exec 10<&0
# stdin replaced with a file supplied as a first argument
exec < $1
# remember the name of the input file
in=$1

# init
file="current_line.txt"
let count=0

# this while loop iterates over all lines of the file
while read LINE
do
    # increase line counter
    ((count++))
    # write current line to a tmp file with name $file (not needed for counting)
    echo $LINE > $file
    # this checks the return code of echo (not needed for writing; just for demo)
    if [ $? -ne 0 ]
     then echo "Error in writing to file ${file}; check its permissions!"
    fi
done

echo "Number of lines: $count"
echo "The last line of the file is: `cat ${file}`"

# Note: You can achieve the same by just using the tool wc like this
echo "Expected number of lines: `wc -l $in`"

# restore stdin from filedescriptor 10
# and close filedescriptor 10
exec 0<&10 10<&-

#!/bin/bash
# Simple line count example, using bash
#
# Bash tutorial:
↪ http://linuxconfig.org/Bash_scripting_Tutorial#8-2-read-file-into-bash-array
# My scripting link: http://www.macs.hw.ac.uk/~hwloidl/docs/index.html#scripting
#
# Usage: ./line_count.sh file
```

```
# ----------------------------------------------------------------------------

# Link filedescriptor 10 with stdin
exec 10<&0
# stdin replaced with a file supplied as a first argument
exec < $1
# remember the name of the input file
in=$1

# init
file="current_line.txt"
let count=0

# this while loop iterates over all lines of the file
while read LINE
do
    # increase line counter
    ((count++))
    # write current line to a tmp file with name $file (not needed for counting)
    echo $LINE > $file
    # this checks the return code of echo (not needed for writing; just for demo)
    if [ $? -ne 0 ]
     then echo "Error in writing to file ${file}; check its permissions!"
    fi
done

echo "Number of lines: $count"
echo "The last line of the file is: `cat ${file}`"

# Note: You can achieve the same by just using the tool wc like this
echo "Expected number of lines: `wc -l $in`"

# restore stdin from filedescriptor 10
# and close filedescriptor 10
exec 0<&10 10<&-
```

Também é possível incluir código diretamente no ficheiro LaTeX 2_ε, como no exemplo em baixo. A numeração das linhas é importante para ser possível referir o número da linha numa descrição.

Listagem 3: Código fonte Python com sintaxe colorida

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

return M
```

4

CRITICAL ANALYSIS AND PROPOSED IMPROVEMENTS

O uso do LaTeX 2$_\varepsilon$ permite-nos focar no essencial: o conteúdo, a formatação é tratada de forma automática.

Para mais informações sobre o LaTeX 2$_\varepsilon$ aconselha-se a consulta do livro *The Not So Short Introduction to LaTeX 2$_\varepsilon$* Oetiker et al., 2000.

Para a gestão de referências bibliográficas aconselha-se o JabRef.

5

CONCLUSIONS

O uso do LaTeX 2$_\varepsilon$ permite-nos focar no essencial: o conteúdo, a formatação é tratada de forma automática.

Para mais informações sobre o LaTeX 2$_\varepsilon$ aconselha-se a consulta do livro *The Not So Short Introduction to LaTeX 2$_\varepsilon$* Oetiker et al., 2000.

Para a gestão de referências bibliográficas aconselha-se o JabRef.

BIBLIOGRAPHY

Daniel, Larry and Lars Daniel (2011). *Digital Forensics for Legal Professionals*. Syngress.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. John Murray.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*.

Koza, J. R. (1992). *Genetic Programming. On the programming of computers by means of natural selection*.

NIST (n.d.). *Forensics Science*. Website. https://www.nist.gov/topics/forensic-science.

Oetiker, T. et al. (2000). *The Not So Short Introduction to LaTeX 2ε*. http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.

Rudolph, Konrad (2016). *The minted package: Highlighted source code in LaTeX 2ε*. http://mirrors.fe.up.pt/pub/CTAN/macros/latex/contrib/minted/minted.pdf. CTAN.

Sims, Karl (1992). «Interactive evolution of dynamical systems». In: *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Ed. by F. Varela and P. Bourgine. Paris, FR: MIT Press, pp. 171–178.

Wikipedia (2007). *Spore video game*. Website. http://en.wikipedia.org/wiki/Spore_(video_game).

APPENDICES

A

APPENDIX A

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

A.1 APPENDIX SECTION TEST

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris.

LABITUR BONORUM PRI NO	QUE VISTA	HUMAN
fastidii ea ius suscipit instructior	germano titulo	demonstratea personas
quaestio philosophia	facto	demonstrated

Table 2: Autem usu id.

Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

A.2 ANOTHER APPENDIX SECTION TEST

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

B

APPENDIX B

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

D E C L A R A Ç Ã O

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"AUTOPSY – Enhanced distributed forensic analysis"*, é original e foi realizado por Pedro Henrique Gaspar Cordeiro Ferreira (2180078) sob orientação de Professora Doutora Marisa da Silva Maximiano (marisa.maximiano@ipleiria.pt).

Leiria, Fevereiro de 2020

Pedro Henrique Gaspar Cordeiro Ferreira