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

[February 7, 2020 at 16:49 – version 1.0]

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

[February 7, 2020 at 16:49 – 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.

R E S U M O

TODO.

A B S T R A C T

TODO.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASCII	American Standard Code for Information Interchange.
BSD	Berkeley Software Distribution.
CSS	Cascading Style Sheets.
DOS	Disk Operating System.
FTK	Forensics Toolkit.
FTP	File Transfer Protocol.
GPT	GUID Partition Table.
JWT	JSON Web Token.
NFTS	New Technology File System.
NIST	National Institute of Standards and Technology.
OS	Operating System.
OTP	One Time Password.
RSA	Rivest-Shamir-Adleman.
STIX	Structured Threat Information eXpression.
TSK	The Sleuth Kit.
U2F	Universal Second Factor.

UUID Universally Unique Identifier.

1 INTRODUCTION

In the scope of the master's degree in cybersecurity and digital forensics, students had the choice between writing a thesis, developing a project or doing an internship. This document is the report for a curricular internship conducted at VOID SOFTWARE, S.A., where the main goal of the internship was to develop a collaborative digital forensics platform, while also learning to develop software in an enterprise environment.

This chapter encompasses the motivation for this internship, a characterization of the host entity, the scope of the proposed project, and the structure of this document.

1.1 MOTIVATION

As a computer science student, my major interest has always been software engineering, and even after completing the first year of a master's degree in a more advanced subject, my interests remained unchanged. Based on that interest, the opportunity to work in an enterprise environment was captivating, as I could gain experience in the field while also developing an interesting project to complement my education.

1.2 HOST ENTITY

To attain an internship, an interview was conducted at "VOID SOFTWARE, S.A.", and a 9 month long curricular internship was planned. The initial project proposals were the development of a client-server model for the existing Autopsy platform, or the adaptation of the existing Autopsy platform for MacOS environments, the chosen proposal was the former.

1.2.1 *Company Information*

VOID is a privately held software development company established in Leiria, Portugal, in 2006, focused on building high-end products embodied in web, mobile and desktop applications, supported by creative software engineering tailored to each challenge's specific needs. It currently employs 30 high-end professionals in several fields of expertise.

1.2.2 *Work Environment*

VOID prides itself in providing very good conditions to it's workers, making them feel like they are at home while working, and also to feel motivated to come to work every day. These conditions include the work environment itself, which is an open space where everyone can interact with each other, the "play areas" where people can relax while playing a game of pool or video games and the rooftop terrace where workers can relax on sunny days. The company also makes sure nothing is missing to provide the best work environment possible by always having food and drink to all it's workers at any time.

1.2.3 *Areas of Expertise*

VOID mostly functions as a company that develops software tailored to the specifications provided by the client, although it can also provide services in different areas like cybersecurity and digital forensics.

The company is capable of comfortably providing services in the following areas:

- Blockchain
- Machine learning and data science
- Augmented reality and virtual reality
- Mobile applications
- Web applications
- Desktop applications
- Cybersecurity and digital forensics

1.2.4 *Main Projects*

Throughout it's 14 years of being active in the software development industry, VOID has conducted some very interesting projects, as can be seen in Table 2.

Name	Description
Yes Account	A suite of applications for automated digitization of accounting documents
Web Portal	A large scale project for the European commission
Digital Archive	A digital preservation application
Dream Football	A social network along with web and mobile applications
Fuel Write	A comprehensive platform for fleet management, data collection and route optimization
Caspers	A mobile augmented reality customer experience and engagement
PBCore Toolkit	A desktop application to support the creation, editing, and export of moving image-related inventory metadata as PBCore XML records
Avenue Securities	A trading platform

Table 2: VOID's Main Projects

1.3 PROJECT SCOPE

The planned project can be categorized in the field of digital forensics, even though the main focus is on typical software development, there are plenty of advanced concepts related to forensics science and cybersecurity that must be assimilated for the project to succeed.

1.4 DOCUMENT STRUCTURE

This document contains five chapters, the first one provides an introduction about the internship that was carried out, the second contains important concepts to help better understand the contents of this document, the third focuses on the development of the platform that is the main focus of the internship, the fourth presents experiences from being involved in different projects inside the company,

and finally the conclusion about the performed internship is presented in the fifth chapter.

2

BACKGROUND

The scope of this internship concerns digital forensics, as it focuses on adapting an existing forensics platform into a collaborative client-server model.

In this chapter, a contextualization of digital forensics is given, an analysis of both *The Sleuth Kit* (TSK) [1] and *Autopsy* [2] is made, it's given a brief description of the existing forensic platform alternatives and the proposed solution for the scope of the internship is described.

2.1 DIGITAL FORENSICS

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* [3]) 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 [4], 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.

2.1.1 *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" [5].

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.

2.1.2 *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 available, the basic principles of digital forensics apply to all of them.

These principles encompass four areas:

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

Each of these areas includes specific forensic processes and procedures.

Acquisition

Acquisition is the process of collecting electronic data. Seizing a computer at a crime scene or taking custody of a smartphone in a civil suit are examples of device acquisition, but the data must be extracted from these devices using specific procedures that equate to making a copy of the storage devices, while following strict rules to ensure the integrity of all the extracted data.

Since acquisition is the first interaction between the investigators and the evidence, it is the step where it's most likely to occur modifications of the contents of the seized devices, because turning on the device or extracting the data without following the right procedures can alter it's contents irreversibly.

Preservation

For evidence to be defendable in court it must be preserved properly. Preservation in the forensics context is the process of creating a chain of custody that begins before collecting the evidence and ends when the evidence is released. Any interference in the chain of custody can lead to issues regarding the validity of the evidence. Additionally, preservation includes maintaining the evidence in a safe environment, preventing intentional destruction with malicious purpose or accidental modification by unqualified people.

A chain of custody log allows proving that the integrity of the evidence has been maintained from seizure through presentation in court. It should contain entries for every time that a piece of evidence has been touched, including collection, storage transport, and any time the evidence is checked out for handling by any personnel.

Analysis

Analysis is the process of locating and categorizing items from evidence that has been collected in a case. Each case is unique as the circumstances surrounding

each case can vary immensely, not only in the evidence being analyzed, but also in the approach used to perform the analysis. The analysis is 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. Considering that electronic evidence appears in so many forms and comes from diverse locations and devices, the training and experience of the examiner has a much greater impact on the results of the examination.

Analysis of digital evidence is more than just determining whether a file exists on a hard drive, it involves finding out how that file got on the hard drive, and if possible, who put the file on the hard drive.

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 sworn statements, depositions of experts, and court testimony. There are no exact rules or standards for reporting the results of an examination. Each 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 procedures used by the examiner, and the results of the 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.

2.2 THE SLEUTH KIT

TSK is a library and collection of command line tools that allow the investigation of disk images. The core functionality of TSK allows volume and file system data analysis. The plug-in framework allows incorporation of 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.

2.2.1 *Overview*

The original part of TSK is a C library and collection of command line file and volume system forensic analysis tools. The file system tools allow examining file systems of a 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 can be shown.

The volume system tools allow examination of the layout of disks and other media. TSK supports DOS partitions, BSD partitions, Mac partitions, Sun slices, and GPT disks. With these tools, partition locations can be identified and extracted so that they can be analyzed with file system analysis tools.

When performing a complete analysis of a system, command line tools can become tedious. Autopsy is a graphical interface to the tools in TSK, which allows easier conduction of 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 tools to easily incorporate file analysis modules that were written by other developers.

2.2.2 *Search Techniques*

TSK allows listing allocated and deleted ASCII and Unicode file names, can display the details and contents of all NTFS attributes, can display file system and metadata structure details, can create time lines of file activity, which can be imported into a spread sheet to create graphs and reports. TSK allows the lookup of file hashes in hash databases, it organizes files based on their type, and pages of thumbnails can be made from graphic images to facilitate quick analysis.

2.3 AUTOPSY

Autopsy is a digital forensics platform and a graphical interface to TSK along with other digital forensics tools. It is used by law enforcement, military, and corporate examiners to investigate what happened on a computer. It can even be used by anyone to recover photos from a camera's memory card.

Autopsy was designed to be intuitive out of the box. Installation is easy and wizards guide the user through every step. All results are shown in a single tree.

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. An overview of Autopsy's modules can be found in Table 4.

Name	Description
Timeline Analysis	Advanced graphical event viewing interface
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, Internet Explorer and Microsoft Edge
Data Carving	Recover deleted files from unallocated space
Multimedia	Extract EXIF metadata from pictures and videos and display these files
Indicators of Compromise	Scan a computer using STIX

Table 4: Autopsy's Modules

Autopsy runs background tasks in parallel using multiple cores and provides results as soon as they are found. It may take hours to fully search a drive, but the user will know in minutes if certain keywords were found in a specific folder.

Autopsy is free and open source, allowing for cost-effective digital forensics analysis and community contributions.

2.4 RELATED SOFTWARE

There are plenty of alternatives to Autopsy in the form of digital forensic analysis platforms, but the ones with most recognition work on a software as a service model, without providing a free trial, so it wasn't possible to test them and compare them.

2.4.1 *Nuix*

Nuix Lab [6] allows investigators to work on large investigations. It can be used for local or small regional forensic labs handling high data volume, variety, and complex digital evidence and looking to build or upgrade a dedicated digital forensics facility.

The core technologies of the Nuix Lab, Nuix Workstation and Nuix Investigate, give digital forensic technicians and case investigators different viewpoints into the same case data. Investigators can collaborate on the same data at the same time by using browser based tools.

The Implementation of *Elasticsearch* [7] as a data store for the Nuix Lab boosts evidence processing, investigation, and intelligence capabilities.

Nuix Lab also claims to contain powerful artificial intelligence, machine learning, and analytics which should make it stand out from the competition.

2.4.2 *EnCase Forensic*

EnCase Forensic [8] enables searching, identifying, and prioritizing potential evidence, in computers and mobile devices, to determine whether further investigation is warranted.

It can collect from a wide variety of operating and file systems, including over 25 types of mobile devices, and parses the most popular mobile apps across iOS, Android, and Blackberry devices. Has strong decryption capabilities, allowing identification and unlocking password-protected files, and contains a custom indexing engine.

EnCase Forensic provides a wide range of capabilities that enable performing deep forensic analysis as well as fast triage analysis from the same solution, and provides a flexible reporting framework that empowers tailoring case reports to meet specific needs.

2.4.3 *Forensics Toolkit*

Forensics Toolkit [9] is an award-winning, court-cited digital investigations solution.

It locates evidence, collects and analyzes any digital device or system producing, transmitting or storing data by using a single application for multiple devices.

All digital evidence is stored in one case database. It reduces the time, cost and complexity of creating multiple datasets, and there is continuous data transfer between AccessData's forensic and e-discovery solutions, allowing for collaboration between all parties working on the case.

FTK allows users to create images, process a wide range of data types from forensic images to email archives and mobile devices, analyze the registry, crack passwords, and build reports.

FTK allows collaboration, contains indexed searches, constructs timelines and other graphical assets, categorizes all the extracted atrifacts, and also contains malware identification modules.

2.4.4 *Comparison*

A comparison of the features offered by each software is presented in Table 6.

Feature	**Autopsy**	**Nuix**	**Encase**	**FTK**
Collaboration	✓	✓	✓	✓
Web Based Interface	✗	✓	✗	✗
Indexed Searches	✓	✓	✓	✓
Decryption	✗	✗	✓	✓
Artificial Intelligence	✗	✓	✓	✗
Malware Analysis	✗	✗	✗	✓
Report Generation	✓	✓	✓	✓
Free and Open Source	✓	✗	✗	✗
Modular Plugins	✓	✗	✗	✗

Table 6: Software Comparison

2.5 PROPOSED SOLUTION

As can be seen from the existing alternative software, collaboration is a major feature included in all these programs, and while Autopsy also allows the software to be configured in a manner to allow collaboration it involves a complicated setup and has certain limitations. The setup required for collaborative cases in Autopsy is as follows:

- Shared hard drive accessible to every machine using the same drive letter
- PostgreSQL Database server
- Solr indexing server
- ActiveMQ messaging server

The limitations of Autopsy's multi-user case feature are that it requires every user to be using a Windows O.S. computer and it also requires specific configuration on each machine involved in the case that is set up.

The proposed solution aims to cover three main aspects:

1. Accessibility
2. Collaboration
3. Organization

2.5.1 *Accessibility*

Given a client-server architecture, any client with access to the network where the server is located can access the contents provided by the server. The aim is to condense the processing heavy features of Autopsy in a single server and provide any number of clients access to this information, requiring less resources from each client, allowing collaboration and removing any type of setup required for each of the client machines, while also providing a more modern and user friendly design.

2.5.2 *Collaboration*

In order to provide collaboration, all the information is maintained in a single server, or a collection of servers providing different functions (like exposing endpoints, storing data, and indexing searches), and every client can preform all the allowed actions whether they consist in consulting, generating, or removing information. Collaboration comes naturally with a client-server model, as the same server that provides the endpoints can also communicate with each client using websockets, and maintain information in a coordinated state along every connected client.

2.5.3 *Organization*

Digital forensics investigations are usually done by specialized organizations, who need to organize their human resources in an efficient and secure manner. Assigning investigators to teams, assigning teams to cases, allowing access to the platform and certain information is a critical part of the activities preformed by a company that specializes in digital forensics, so having these functionalities properly integrated into a digital forensics platform should be an important feature.

3

KENSENTME PLATFORM DELEVOPMENT

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

The main goal of the internship is the development of a collaborative platform based on Autopsy, which was soon named "Kensentme" by the company's designer.

Each step taken towards the development of this platform is documented in this chapter, from project planning to full roll out on a production environment.

3.1 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 client-server model, with the server developed in Java using the Quarkus [10] framework, and the client developed in Javascript using the React [11] 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 files to their respective data source directories, and FTP access will be controlled according to each user's credentials on the platform.

3.2 AUTOPSY SOURCE CODE ANALYSIS

Autopsy is a digital forensics analysis software that is available as Open Source Software [12] on *GitHub* [13].

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, as can be seen in Table 7.

Name	**Description**
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 generation utilities and modules
Timeline	recent addition to Autopsy, allows visualization of artifacts in temporal chart, only available for Windows O.S.

Table 7: Autopsy Modules Overview

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.

Another module that needs to be adapted is the “RecentActivity“ module, which contains the tools needed to extract information from browsers, registry and other important resources, providing a great amount of critical evidence from data sources.

Each autopsy case contains it’s own SQLite database, which contains the tables present in Table 8.

tsk_db_info	tsk_db_info_extended	tsk_objects
tsk_image_info	tsk_image_names	tsk_vs_info
tsk_vs_parts	tsk_fs_info	data_source_info
tsk_files	file_encoding_types	tsk_files_path
tsk_files_derived	tsk_files_derived_method	tag_names
review_statuses	blackboard_artifacts	blackboard_attributes
ingest_modules	blackboard_attribute_types	ingest_module_types
reports	blackboard_artifact_types	ingest_jobs
ingest_job_modules	ingest_job_status_types	account_types
accounts	account_relationships	tsk_event_types
tsk_examiners	blackboard_artifact_tags	content_tags

tsk_file_layout	tsk_event_descriptions	tsk_events
tsk_pool_info		

Table 8: Case Database Tables

The tables containing the information accessed most frequently are the ones related to blackboard artifacts, along with tables related to files and tags.

Autopsy's source code allows access to most of the important entities present in this database, through pre-defined queries which return entities like data sources, abstract files, or artifacts. But in order to add features like pagination and tag deletion, custom queries must be created, so that the queries may return only a specific amount of entity ids, or so that delete commands may be executed.

3.3 DEVELOPMENT STAGES

3.3.1 *Basic Autopsy Functionalities*

As a first step into replicating the functionalities of the original program, 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 "repository", containing all the different cases created within the application.

Later in the development there was the need to create an additional directory alongside the "repository" called "central-repository" which contains the database used by the Correlation Engine, which is a feature that finds files present in multiple cases, to ingest data that can be queried by any case.

The development of autopsy features was done alongside the original software, so that every new feature implemented into the platform could be compared and verified with the original software, and it was reassuring to see that cases created

through one of the versions of the software were capable of being opened through the other version, meaning that the source code provided by autopsy, with some tinkering, can definetely be used in a client-server model.

3.3.2 *Authentication Process*

The authentication process is handled through a REST API, when the authentication is completed the user receives a JSON Web Token (JWT).

JWT is an open standard that defines a compact and self-contained way for securely transmitting information between parties. This information can be verified and trusted because it is digitally signed. JWT contain three parts, as can be seen in Table 10.

Name	**Contents**
Header	Information about the signing algorithm used
Payload	Information about the user, such as username, e-mail and roles
Signature	A signature used to verify the contents weren't changed

Table 10: JWT Composition

The authentication can be performed with either username or e-mail and a password, and extra authentication factors can be added such as One Time Passwords (OTP) and Universal Second Factor (U2F).

If no extra factors were added to an account, the authentication attempt with the user's credentials will return the JWT when successful and the user can freely access protected content.

When extra factors are present in an account, the response from the same request will contain an array with the extra factors added, and a UUID, which is a pseudo random string, which must be sent in the next request to validate the user's identity. This UUID is saved in the database in the form of salted hash using the blowfish algorithm, and is invalidated after a successful login attempt. The user then has the choice between any of the extra factors and must complete the required steps to validate that factor, as can be seen in Figure 1.

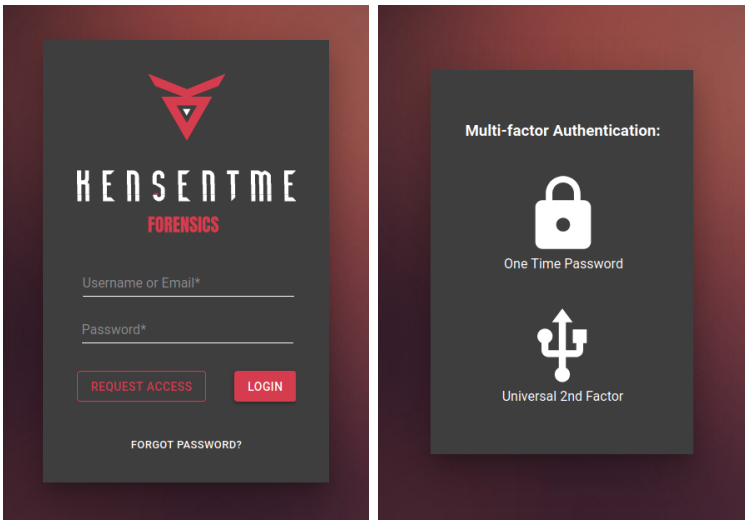

Figure 1: Login Screen and Extra Factors

One Time Password

The process of seting up OTP for an account begins with generating a secret string, which is used by both the server and the authenticator app to generate OTP, when the secret is generated, it is saved as a temporary secret after being encrypted by the RSA algorithm.

The temporary secret must be added to the user's device, either by inserting the string itself or by scanning a QR code, and must be validated by the user by submitting the current OTP, after validation the temporary secret is considered permanent.

All OTP validations have a 30 second threshold, which means the current OTP is valid for an extra 30 seconds after being replaced by the next OTP, allowing the user to have a better experience with these passwords.

The processes of authenticating and removing this authentication factor both depend on validating the current OTP, which is provided by the user's app.

In the event of loss of the device that the user uses to generate OTP, all the authentication factors can be reset using the "Reset Password" functionality, which relies on e-mail validation to prove the user's identity.

Universal Second Factor

TODO

3.3.3 *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 2.

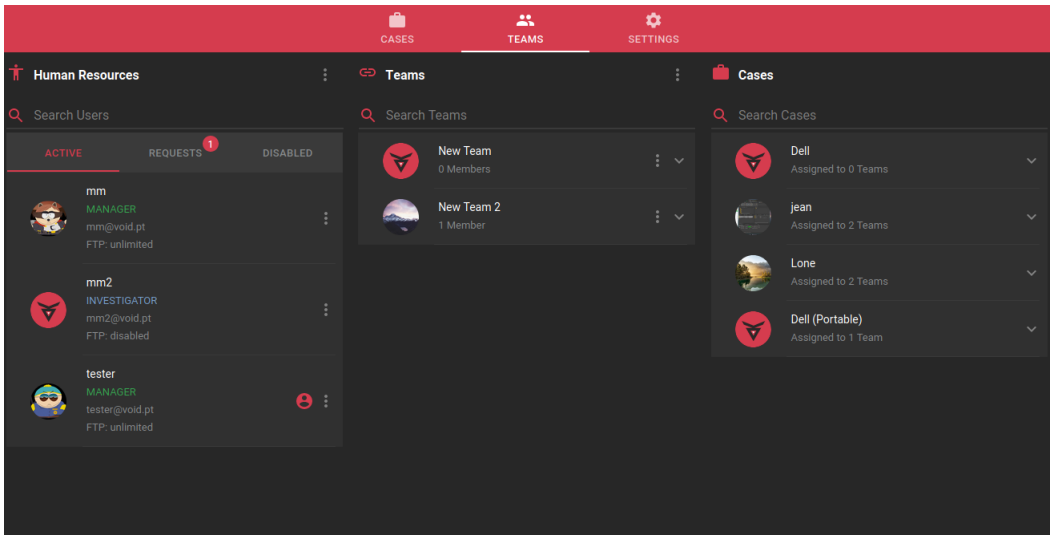

Figure 2: Entity Management Interface

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 when activating his account through a received e-mail message.

It was at this step in development that was made clear by the testers, that validations must be preformed on both server and client side, and that error handling must be streamlined in order to provide a seemless experience to the user, while also allowing future additions to the platform to be implemented in a similar way without much tinkering. Because even though the platform still wasn't fully

developed, it already suffered from possible exploits that could break the experience for it's users, like the upload of a 5GB file as an image file, which would stop the page from loading, or possible directory traversal attacks, resulting from the creation of case directories.

The lessons learned from the testers on this stage of development carried on through the entire development process and ensured that every new feature added was accounting for possible weaknesses which may result from the new implementations, resulting in a more secure development process.

3.3.4 *Ingested Results Presentation*

Ingested results are the items present inside the provided data sources, Autopsy can run multiple modules on each data source to extract results, and the extracted results can either be a file instance or an artifact instance. The difference between a file and an artifact is that an artifact represents only a smaller piece of a file's information, which means a file can contain multiple artifacts, for example a registry file can contain multiple O.S. user account artifacts.

Even though the data ingestion features weren't implemented yet on the platform, the data that was ingested into a case using the original program could also be used on this platform, which allowed this feature to be developed earlier.

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 or thumbnail viewer, presenting all the contents of the results 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 3.

The content viewer can display different kinds of information depending on the type of item selected, these can be some of the following:

- Text browser
- Media viewer
- Database browser
- Registry browser
- Key-value browser
- Table data viewer

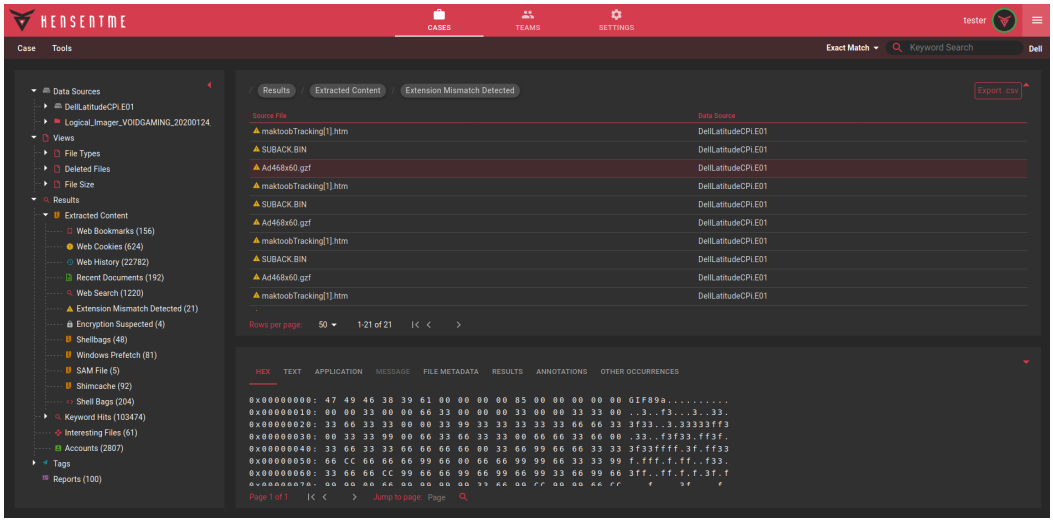

Figure 3: Ingested Results Presentation

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

This development stage was one of the most intensive, as it required understanding Autopsy's data structure very well. The source code used here was rewritten almost completely, because of the way the user interface and the data structure in the original program and interconnected, resulting in the need to adapt and write custom queries to obtain the desired results, creating more than 50 endpoints so that all the needed information could be supplied. As well as creating and adjusting all the UI elements to achieve an user experience similar to what Autopsy's users are used to.

3.3.5 *Data Sources*

Using the same credentials used to log-in to the platform, the user can also upload data source files into his folder located inside the server, using an FTP client like filezilla. Then the user can browse these directories using the web interface and select a data source to add to the case, as can be seen in Figure 4.

The procedure for adding a data source to a case was adapted from Autopsy's source code, and depends on the type of data source added, which can be one of the following:

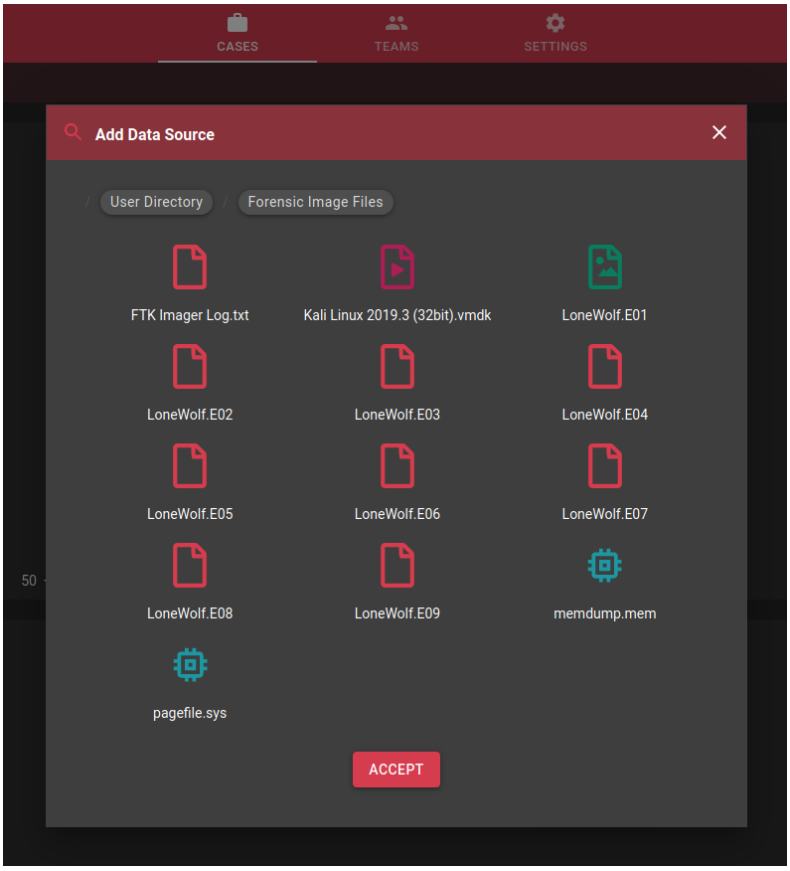

Figure 4: Data Source Selection

- Disk image
- Virtual machine
- Logical file collection
- Autopsy logical imager results

Local disk data sources were also an option provided by Autopsy but since the local disks the software has access to belong to a server, this feature is undesired.

When a data source is added to a case, it is processed and instances of abstract files and other entities are created, which allow the user to navigate the data source's contents as soon as it is processed. Just navigating the data source contents isn't very useful, as the contents are in a raw state, showing mostly the file structure of the source, and not even categorizing files by mime types, so running data ingestion modules is a critical step into allowing a deeper analysis of a data source, which is covered in the next section.

3.3.6 *Data Ingestion Modules*

Firstly the default modules included with Autopsy were adapted, these modules are either file ingest modules, which analyse each file contained in a data source, one by one, or data source ingest modules, which run on specific components of a data source.

To run a collection of modules, the user selects which modules to run, and may also configure some parameters related to the modules, when the request is received by the server, the modules are started in a background thread.

When the modules are running the server communicates to each client through websockets, offering feedback on the progress of the modules, as can be seen in Figure 5.

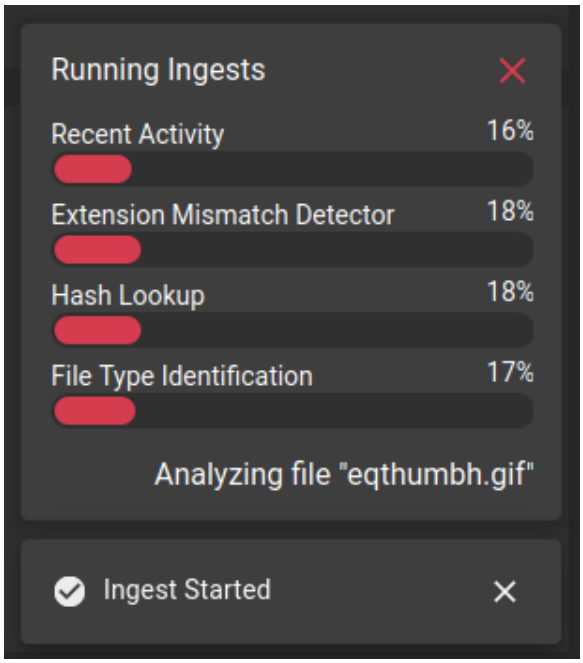

Figure 5: Ingest Modules Communication

Originally it was intended to have the clients update their explorer as new information is ingested, but it would slow down the ingest progress considerably, so what was decided was that every client would refresh it's explorer every 30 seconds while ingests are running.

There were other measures taken into consideration to improve the performance of the ingest procedure, while also maintaining the user informed of it's progress, the server communicates a maximum of one file name per second, and it also only communicates a module's progress when it advances at least one percentual point.

3.3.7 *Report Modules*

TODO

3.3.8 *Add-on Modules*

TODO

3.4 CRITICAL ANALYSIS AND PROPOSED IMPROVEMENTS

TODO

4

OTHER PROJECTS DEVELOPED

4.1 IGIVEU

4.1.1 *Critical Analysis and Proposed Improvements*

4.2 ANOTHER PROJECT?

4.2.1 *Critical Analysis and Proposed Improvements*

5

CONCLUSIONS

TODO

BIBLIOGRAPHY

[1] T. S. Kit, *The Sleuth Kit*, Website, http://www.sleuthkit.org.

[2] T. S. Kit, *Autopsy*, Website, https://www.autopsy.com/.

[3] L. Daniel and L. Daniel, *Digital Forensics for Legal Professionals*. Syngress, 2011.

[4] NIST, *Forensics Science*, Website, https://www.nist.gov/topics/forensic-science.

[5] Microsoft, *Cloud Computing*, Website, https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/.

[6] Nuix, *Nuix Lab*, Website, https://www.nuix.com/solutions/investigations.

[7] elastic, *Elasticsearch*, Website, https://www.elastic.co/products/elasticsearch.

[8] G. Software, *EnCase Forensic*, Website, https://www.guidancesoftware.com/encase-forensic.

[9] A. Data, *Forensics Toolkit*, Website, https://accessdata.com/products-services/forensic-toolkit-ftk.

[10] Quarkus, *Quarkus*, Website, https://quarkus.io/.

[11] React, *React*, Website, https://reactjs.org/.

[12] O. Source, *Open Source*, Website, https://opensource.com/resources/what-open-source.

[13] Microsoft, *GitHub*, Website, https://github.com/.

APPENDICES

A

APPENDIX A

B

APPENDIX B

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