Implementation and evaluation of an open-source digital forensic timeline visualisation tool for data exploration

Casey Donaldson
School of Design and Informatics
Abertay University
DUNDEE, DD1 1HG, UK

ABSTRACT

Context
Due to the rise of digital evidence, the analysis of such evidence has been slowed down. Forensic investigators have been accelerating the process by using applications to structure the data. One method is timeline visualisation, renowned for showing incidents in which they first occur, helping investigators identify patterns and vital evidence.-

Aims
This project aims to research, implement and evaluate an appropriate open-source application which quickly displays temporal data in a scalable application, the display should be readable and useable for low-technical users.

Method
The project will undertake initial research to examine the most effective ways to implement an application for displaying temporal data. The implementation phase will begin after following rapid prototyping to implement 3 main stages, these are the data structure, the extraction method, and the graphical user interface. Once the application is created, the evaluation stage will begin, basing the success on performance, useability, and scalability.

Results
The outcome of this project is expected to be a working application which allows users to easily display temporal data in a clear format, allowing analysis of digital evidence and assisting in digital forensics investigations.

Conclusion
The outcome of the project will be a high-performance, scalable, and easy-to-use application which will take multiple data sources and generate a timeline—assisting in digital forensic investigations.

Keywords
Digital Forensics, Visualization, EXIF Data, PCAP, SDLC, IDE, Database, Temporal Data, Data Exploration.

1. INTRODUCTION
Digital Forensics is a forensic science where the evidence is digital and must be investigated and presented in a court of law for a criminal conviction. The process follows 5 steps, identification, preservation, examination, analysis and presentation of the digital evidence (Bhandari, 2022). Over time digital forensics has become an increasingly vital part of a criminal investigation, with more digital evidence being generated from mobiles, PCs, Laptops, IoT devices and many more. With the volume of data being generated and the velocity at which the data is generated, this leaves investigators struggling to meet demands (Adedayo, 2016).

Once evidence has been identified, collected and preserved the next step in a forensics investigation is to analyse and present the data. This is hindered due to the overwhelming amounts of data. Current methods for data visualization are bar charts, file system trees, geolocation maps, network traffic flow diagrams, heat maps, timelines and many more. Using timelines to visualize data and when the incident happens, has become one of the most popular methods for visualizing and exploring data. Current technologies have revealed that most of the data exploration applications include a timeline. Magnet AXIOM, FTK, EnCase, Plaso, Sleuth Kit and Redline all contain a timeline method for displaying data. However issues from these tools can arise, these could be a lack of readability, scalability, useability and performance. This is worsened when the data is scaled up and hinders investigators as it can cluster timelines, slow down the generation of said timeline and slow down the overall investigation.

This project aims to design and create an application which will be able to take multiple sources of data and plot a timeline. The application will be designed to make data exploration easy and efficient. Users will be able to upload a database source which can be generated from Autopsy. Additionally, the program will offer to take other data sources like PCAP files, Registry hive, and EXIF data files. Another bonus of this will be the adaptability of taking several different files and applying them to one timeline. The evaluation of the program is expected to be based on performance, useability, adaptability, and scalability. Below is the summary of the objectives set out for the project.

- Research current methods for timeline visualization, applying best-known techniques.
- Implementation of a scalable, high-performance, useable and adaptable program to visualise timeline data for easy exploration.
- Evaluation of the program, including tests on a wide range of data sources.

2. BACKGROUND
90% of data and information is considered unstructured, coming from documents, emails, messages and other similar protocols (Henseler and Hyde, 2019). Using an application to manipulate the data into a structured format sorted from the first occurrence can benefit investigators as this can

make recognising patterns and identifying evidence a lot easier. This approach is referred to as timeline visualisation. This is a type of data exploration. This term is used to describe the process of analysing forensic evidence in forensics investigations, identifying hidden data patterns and anomalies which are critical for investigative and legal outcomes. Digital visualization for timelines has been a challenge for several vendors and investigators. How the data is displayed can vary based on the application. Autopsy and FTK use a bar chart with a timeline below to show density and time. Whereas Plaso (Log2Timeline) uses a CSV-type approach. The list can be sorted and filtered but the abundance of incidents and lack of visualization can cause investigators to miss obvious patterns.

2.1 Other Applications

Other applications have a variety of methods for visualizing timeline data. Cambridge Intelligence created KronoGraph which is a toolkit for timeline visuals. The toolkit explores the use of timelines, calendar heatmaps and scalable interactive views similar to the design shown in Figure 1 (Disney, 2020).

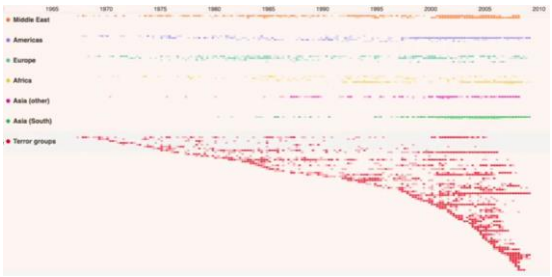

Figure 1 KronoGraph interactive timeline, taken from (Disney, 2020).

A paper by Jens Olsson and Martin Boldt discusses an older application from 2009 called TimeLab made in C#. The application is broken down into two main rows, one to view all the different timelines for different partitions and the second row is where the details for each incident are displayed, shown in Figure 2, (Olsson and Boldt, 2009).

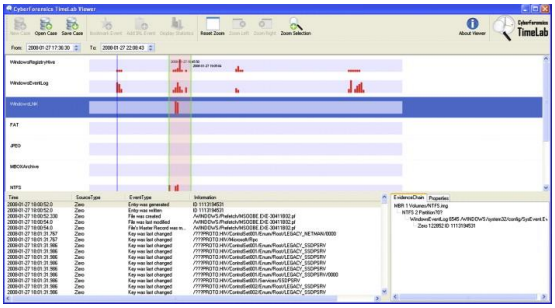

Figure 2 TimeLab timeline, taken from (Olsson and Boldt, 2009).

These applications show a large variety of methods for displaying the data, these methods will be considered for the project. Furthermore, Autopsy, Plaso, and Magnet AXIOM will be further examined to see if there are any benefits from the methods utilised.

3. METHOD

The project will follow 3 main stages, further research, implementation, testing and evaluation. The implementation of the artefact will be the primary objective, the implementation will start after further research into additional methods of timeline visualization. As this project is primarily focused on software development the project will follow the secure software development lifecycle (SDLC). This will benefit the development as it will help facilitate the implementation of a high-quality application and will also ensure efficient planning, timing and risk management (Mohino et al., 2019).

3.1 Research

The initial research undertaken will be used to identify possible methods for displaying data in a timeline fashion, each of these methods will be evaluated based on useability, performance and scalability. As the application will be used by low-technical users it must be easy for users to upload data sources browse the timeline and analyse incidents. The overall performance cannot be poor as this will affect the speed of the investigation. Also, different methods might require different computational power, and this will need to be researched to ensure high performance. This is similar to scalability. The application should be able to take small to large data sources and display the timeline without any cluster or performance issues, furthermore, if any add-ons or additional tasks should be required the application should be adaptable, allowing more versatile options to be developed in the future. This could be file carving from binary files and adding them to the timeline. Additionally, during the research phase, the requirements for the application will be decided. Moreover, during this phase the type of data required to be extracted will be investigated and a structure which will hold the data will be designed.

The research will also investigate possible programming languages which could be used to develop the program, currently, Python with the use of Tkinter has been the most promising in the early stages. However, C++, Java, JavaScript and C# will be further analysed to find the best language for the application, considering performance, libraries for graphical user interfaces (GUIs), ease of programming and the developer's knowledge. Moreover, external libraries will need to be investigated to see if any could be used to assist in building the GUI, Extracting data from data sources and if any library can be used to ease the development. Technologies which are currently highlighted for their uses are Matplotlib and Historical-timelines, these are Python libraries which could be used to help create a clear and useable timeline. If the application was programmed in C++, then GTkmm could be used to help develop a GUI. Furthermore, Log2Timeline and D3.js have been recommended and during the research stage, these will be examined to see if these could be viable routes to pursue. Finally, research into what should be done with files that are found without temporal data.

3.2 Implementation

During the implementation stage, rapid prototyping will be employed in conjunction with the SDLC as the selected development approach. The development of the application will be broken into three main stages, the development of the data structure and class where the data will be stored, the development of extracting the data sources and the development of the GUI. The development of the program will be created using Visual Studio Code as the IDE, this allows for effective debugging and syntax checking.

Stage one will be the design and development of the data structure, this will ensure all data extracted can be stored which will benefit performance allowing searching through the data to be easier. Stage two will be the development of the function(s), these will be responsible for collecting the data from data sources and inserting them into the data structure. This stage is paramount to ensure that all relevant data is captured and stored correctly which will allow stage three to be implemented, without the correct implementation of stage two, data might not be collected correctly and cause inconsistencies throughout the timeline affecting the integrity of the digital forensics' investigation. The types of sources which can be inputted will be dependent on the timescale aiming for the extraction of at least an Autopsy database (DB) and aiming for a variety of additional data sources to be implemented these will be PCAP, EXIF and other file formats which could contain digital evidence.

The final stage is the implementation of the GUI. This stage is the main focus of the project. Several different methods have been investigated and initial research will conclude which method will be used to display the timeline data. Currently, bar charts, line graphs, heat maps and interactive maps have been the most popular methods and an application which can combine these methods might be the most beneficial, the development will look at creating an interactive GUI which also uses a heat map approach to reveal density through the timescale. Once stage one to three is completed then the integration between the GUI and extraction will need to be completed. As this is a full stack development this approach should be simple as Steps two and three are both connected using the same data structure.

3.3 Testing and Evaluation

The final application will be tested using a mix of data sources which are suited for the application extraction. The data will be checked to validate that the application can correctly select and format the data. Additionally, the data for testing will come in a wide range of sizes from small to large to check the performance and readability of the GUI. The same data sources will be used on Autopsy and other timeline visualisation tools to compare the difference noting the pros and cons of the newly designed application providing insight on what further work could be conducted.

The useability of the application will be based on the system useability scale (SUS). Depending on resources and if able to, a focus group will be selected and tasked to analyse a timeline. The SUS will ensure that the program meets useability standards (Lewis, 2018). Once the application is approved the program will be released on GitHub for anyone to use and improve. The last area to evaluate would be the legal consideration, since the tool is newly developed and would be considered open source, courts may attempt to discredit the results.

A paper written by Brian Carrier discusses how open-source tools more clearly and fully meet guidelines, with proper documentation and testing, to reveal the accuracy of the program. This helps to avoid criticism regarding the application's reliability (Carrier, 2002).

4. Summary

In summary, the proposed project aims to investigate current technologies in timeline visualisation for data exploitation. Identifying a strong approach to displaying files with temporal data. Taking into consideration performance, useability and scalability. Research will also include finding an appropriate programming language, external libraries and what to do with files without temporal data. Implementation will take place after following the SDLC in conjunction with rapid prototyping the task will be broken down into 3 conceivable stages which will make the development easier. These stages are the development of the GUI, data structure and extraction of a database file. Once the implementation of the application is completed effective testing in evaluation will take place.

During the testing and evaluation stage, performance, speed, accuracy, useability, and adaptability will be tested and evaluated using a variety of data sources, finishing with evaluating legal considerations. The outcome of this project is to have an application which allows for easier and more efficient temporal data analysis for a digital forensics investigation.

5. REFERENCES

Adedayo, O.M. (2016). *Big data and digital forensics*. [online] IEEE Xplore. Doi:https://doi.org/10.1109/ICCCF.2016.7740422.

Bhandari, S. (2022). *Research and implementation of timeline analysis method for digital forensics evidence /*. [online] Ktu.edu. Available at: https://epubl.ktu.edu/object/elaba:133122013/ [Accessed 10 Oct. 2024].

Carrier, B. (2002). *R e s e a r c h R e p o r t*. [online] Available at: https://www.engineering.iastate.edu/~guan/course/backup-0982/CprE-592-YG-Fall-2002/paper/atstake_opensource_forensics.pdf [Accessed 11 /Oct. 2024].

Disney, A. (2020). *The basics of timeline data visualization*. [online] Cambridge Intelligence. Available at: https://cambridge-intelligence.com/the-basics-of-timeline-data-visualization/ [Accessed 10 Oct. 2024].

Henseler, H. and Hyde, J. (2019). *Technology AssistedD Analysis of Timeline and Connections in Digital Forensic Investigations*. [online] LegalAIIA@ICAIL. Available at: https://www.semanticscholar.org/paper/Technology-Assisted-Analysis-of-Timeline-and-in-Henseler-Hyde/0d83594585c4aeb80e42bc53962d145055caecd2 [Accessed 10 Oct. 2024].

Lewis, J.R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, 34(7), pp.577–590. Doi https://doi.org/10.1080/10447318.2018.1455307.

Mohino, de V., Higuera, B., Higuera, B. and Montalvo, S. (2019). The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies. *Electronics*, [online] 8(11), p.1218. Doi:https://doi.org/10.3390/electronics8111218.

Olsson, J. and Boldt, M. (2009). Computer forensic timeline visualization tool. *Digital Investigation*, 6, pp.S78–S87. Doi:https://doi.org/10.1016/j.diin.2009.06.008.