UFORIA - A FLEXIBLE VISUALISATION PLATFORM FOR DIGITAL FORENSICS AND E-DISCOVERY

Arnim Eijkhoudt & Sijmen Vos
Amsterdam University of Applied Sciences
Amsterdam, The Netherlands
a.eijkhoudt@hva.nl, sijmenvos@gmail.com

Adrie Stander
University of Cape Town
Cape Town, South Africa
adrie.stander@uct.ac.za

ABSTRACT

With the current growth of data in digital investigations, one solution for forensic investigators is to visualise the data for the detection of suspicious activity. However, this process can be complex and difficult to achieve, as there few tools available that are simple and can handle a wide variety of data types. This paper describes the development of a flexible platform, capable of visualising many different types of related data. The platform's back and front end can efficiently deal with large datasets, and support a wide range of MIME types that can be easily extended. The paper also describes the development of the visualisation front end, which offers flexible, easily understandable visualisations of many different kinds of data and data relationships.

Keywords: cyber-forensics, e-discovery, visualisation, cyber-security, computer forensics, digital forensics, big data, data mining

1. INTRODUCTION

With the growth of data that can be encountered in digital investigations, it has become difficult for investigators to analyse the data in the time available for an investigation. As stated by Teerlink & Erbacher (2006) “A great deal of time is wasted by analysts trying to interpret massive amounts of data that isn’t correlated or meaningful without high levels of patience and tolerance for error”.

Data visualisation might help to solve this problem, as the human brain is much faster at interpreting images than textual descriptions. The brain can also examine graphics in parallel, where it can only process text serially (Teerlink & Erbacher, 2006)

According to Garfinkel (2010), existing tools use the standard WIMP model (Window, Icon, Menu, Pointing device). This model is poorly suited to representing large amounts of forensic data in an efficient and intuitive way. Research must improve forensic tools to integrate visualisation with automated analysis, allowing investigators to interactively guide their investigations (Garfinkel, 2010).

Many computer forensic tools are not ideally suited for identifying correlations among data, or for the finding of and visually presenting groups of facts that were previously unknown or unnoticed. These limitations of digital forensic tools are similar to the forensic analysis of logs in network forensics. For example, logs residing in routers, webservers and web proxies are often manually examined, which is a time-consuming and error-prone process (Fei, 2007). Similar considerations apply to E-mail analysis as well.

Another issue with current tools is that they do not always scale well and will likely have problems dealing with the growth of data in digital investigations (Osborne, Turnbull, & Slay, 2010).

Currently, there are few affordable tools suited to

and available for these use-cases or situations. Additionally, the available tools tend to be complex, requiring extensive training and configuration in order to be used efficiently.

Investigative data visualisation is used to assist viewers with little to no understanding of the subject matter, in order to reconstruct a crime or item and to understand what is being presented, for example an investigator which is not familiar with a particular scenario. On the other hand, analysis visualisations can be used to review data and to assess competing scenario hypotheses for investigators that do have an understanding of the subject matter (Schofield & Fowle, 2013).

A timeline is a valuable form of visualisation, as it greatly assists a digital forensic investigator in proving or disproving a hypothetical model proposed for the investigation. A timeline can also provide support for the mandate the digital forensic investigator received prior to commencing the investigation (Ieong, 2006). Interaction between role players can normally also be shown in network diagrams, so that the combination of a timeline and network diagram can generally answer many *who* and *when* answers.

The aspects of *what* and *where* can often be answered by examining the contents of evidence items, such as E-mails or the positional data of mobile phone calls. It is therefore important to be able to display the details of data with ease as well.

This paper describes the development of a flexible platform, Uforia (Universal Forensic Indexer and Analyser), that can be used to visualise many different types of data and data relations in an easy and fast way.

The platform consists of two sections, a back end and a front end, and is based on readily available open source technologies. The back end is used to pre-process the data in order to speed up the indexing and visualisation process handled by the front end. The resulting product is a simple and extremely flexible tool, which can be used for many types of data with little or no configuration. Very little training is needed to use Uforia, making it accessible and usable for forensic investigators without a background in digital investigations or systems, such as auditors.

2. ADVANTAGES

Uforia offers many advantages, of which the first is very low cost.

A second advantage is that the system scales well due to its use of multiprocessing and distributed technologies such as ElasticSearch, so that extremely large numbers of artefacts can be handled in a very short time. The processing of the Enron set, consisting of roughly 500 000 E-mails without attachments, typically takes less than ten minutes to complete on contemporary consumer-grade hardware. This pre-processing step also ensures that little to no processing needs to be done at the time of visualisation.

Thirdly, the Uforia's development heavily focused on making it as user- and developer-friendly as possible. Many forensic tools need a substantial amount of training and configuration to accomplish meaningful tasks. As this makes the systems difficult and expensive to use and develop for, it was considered paramount during Uforia's continued development to address these issues. Although a full UX study has not been conducted yet, the UI and feature set was developed using mock-ups and feedback from UX- and graphical designers, as well as potential users from several fields of expertise, such as process, compliance and risk auditors, forensic investigators and law enforcement officers, where none of the participants were given prior usage instructions.

Another advantage is the extreme flexibility of the system. It is very easy to add new modules, e.g. for handling new MIME types, as the programming of such a module can normally be accomplished in a very short time using simple Python programming. Additionally, the front end is completely web based, and no special software needs to be installed to use it. This, combined with the following common web design and UX standards, suggests that even novice users can achieve meaningful results with little to no training.

3. BACK END

3.1 START-UP PHASE

Uforia's back end is used to process the files containing the data that will eventually be indexed and used in the visualisation process.

The back end's first step is to create a MySQL table for the files. This table contains all metadata common to any digital file, as well as calculated metadata (such as NIST hashes).

A second database table is then generated, and it contains information about the supported MIME types. This table is built by looking at a configurable directory containing the modules for the MIME types that can be handled by the system.

Every module that can handle a specific MIME type is identified and added to this table. The table eventually contains zero, one or more 1:n key/value pairs for each of the supported MIME types and their respective module handlers. The module handlers are themselves stored as key/value pairs, with their original name as keys to the matching unique table name.

These tables are then created for each module, so that Uforia can store the returned, processed data from each particular module in its unique table.

Modules are self-contained files and extremely easy to develop. They only require the structure of their database table to be stored as a simple Python comment line in the particular module, starting with *# TABLE: ...*, and a predefined *process* function which should return the array of the data to be stored.

3.2 PROCESSING

Once all tables are created, the processing of the files that need to be analysed can start.

The first step is to build a list of the files involved. This is read from the config file. Once this list is completed, every file in the list is processed.

The MIME type of the file is determined and then the relevant processing modules (0, 1 ... n) are called to process the file. The results returned by each module are then stored in the database table that was generated earlier for that particular module.

When Uforia encounters a container format, it can deal with it efficiently by recursively calling itself. For instance, the Outlook PST module will unpack encountered PST files to a temporary directory and then call Uforia recursively for that temporary location. The unpacked individual E-mails are then automatically picked up by the normal E-mail module and processed accordingly.

Uforia can also deal efficiently with flat-file database(-like) formats by having modules return their results as a multi-dimensional array. Uforia's database engine turns these into multiple-row inserts into the appropriate modules' tables. Examples of modules that deal with flat-files in this fashion, are the modules that handle the mobile phone data (CSV-format) and the simple PCAP-file parser.

Due to its highly-threaded operation, the back end can pre-process large volumes of data efficiently in relatively little time. Once the processing steps are completed, the stored data needs to be transferred from the back end storage in JSON-format to the ElasticSearch engine for use by the visualisation front end.

4. FRONT END

The front end uses ElasticSearch, AngularJS and D3.js for the visualisation and administration interface.

The first step during the visualisation process is to select the modules or file types that need to be visualised in the admin interface.

The next step is to select (and possibly group any identical) fields that need to be indexed by the ElasticSearch engine. The administration interface will hint at similar field names in other supported data types to allow for the merging of data types into one searchable set. This makes it possible to correlate the timing of e.g. cell phone calls and E-mails.

During or after the indexing and storing in ElasticSearch, one or more visualisations must then be assigned to the mapping in the admin interface. This also includes specifying the fields that should be laid out on the visualisation's axes.

The data in ElasticSearch can then be searched and visualised, even if the index process has not been completed yet. Because the front-end uses ElasticSearch, searches are fast and highly scalable. Only when full detail views of selected evidence items are necessary, the underlying back-end database needs to be accessed.

5. USER INTERFACE

The interface is designed with the goal of optimizing user-friendliness and ease of understanding. The user interface sports a 'responsive design', with UI elements automatically resizing and repositioning themselves for different screen sizes, such as with laptops, tablets and mobile phones, as can be seen in Figure 1.

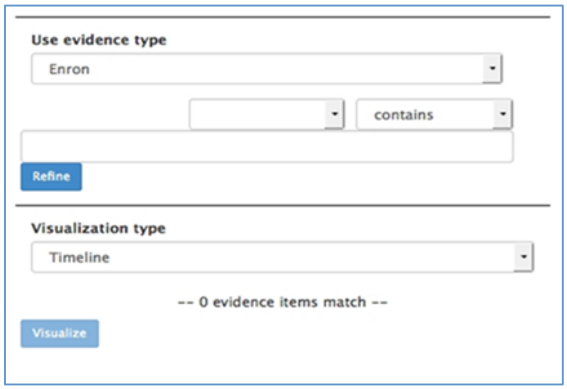

Figure 1: Mobile Interface

1) The user selects an 'evidence type', which is the name used for the collection, as it was generated in the admin interface
2) Uforia then loads the module fields that have been indexed for that evidence type, e.g. 'Content' for E-mails or documents.
3) The user selects whether the field should 'contain[s]' or 'omit[s]' the information in the last field.
4) Finally, the user selects one of the visualisations that have been assigned to the evidence type.
5) Uforia will now render the requested information using the selected visualisation, with some of the visualisations offering additional manipulation (such as a network graph).
   Lastly, all visualisations have one or more 'hot zones' where the user can 'click-through' to bring up a detailed view of the selected evidence item(s).

6. EXAMPLES

In this section, an examples can be seen of how Uforia is can be used to quickly determine the E-mail contacts of suspects. Despite limited available space in this paper, it is nevertheless possible to recreate similar scenarios for other data types.

Figure 2 shows an example set of a network graph derived from a sample set of PST-files, where the E-mail content was searched for the words 'investigate', 'books', 'suspect' or 'trading' and shown as a network graph indicating which individuals communicated about these words, with the size of the node indicating the amount of communication received. This immediately indicates the links between several possible suspects, including one whose PST mailbox was not included in the dataset and processed by Uforia.

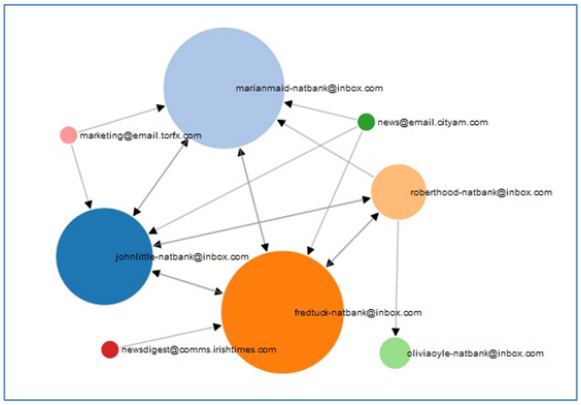

Figure 2: Network Graph

Another example is creating a timeline, as seen in Figure 3, to determine when messages were sent and which were sent around the time of the possible transgression.

It is easy to determine the times of the E-mail messages by hovering over the intersections on the timeline, and to investigate the original E-mails by clicking on the intersections (see Figure 4).

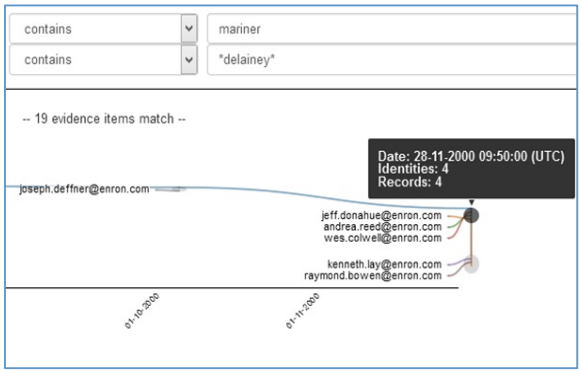

Figure 3: Timeline

The timeline visualisation can handle multiple items like calls from a large number of mobile phones. Figure 4 shows anonymised data from a real case, illustrating how contacts and time can easily be determined. The horizontal axis indicates the flow of time, while the graph nodes and coloured lines indicate the moment of contact between the two phone numbers. By clicking on the intersections, the original data can once again be displayed**.**

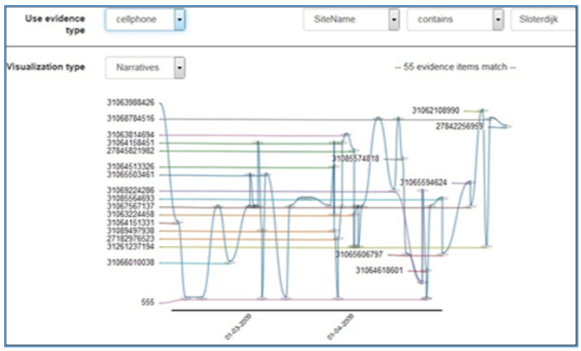

Figure 4: Mobile Phone Timeline

7. CONCLUSION

Uforia shows that it is possible to create a simple, user-friendly product that is nevertheless powerful enough to use in the most demanding investigations.

It is easy to extend if any new MIME types are encountered or new features are needed.

Uforia was tested on a number of real life scenarios, and in all cases it was able to produce real results in a fast and efficient way, requiring hardly any operator training.

In conclusion, Uforia is fast, flexible and low cost solution for investigating large volumes of data.

REFERENCES

Fei, B. K. (2007). *Data Visualisation in Digital Forensics.* Pretoria, South Africa: Maters Dissertation, University of Pretoria.

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *Digital Investigation*, 64-73.

Ieong, R. S. (2006). FORZA - Digital forensics investigation Framework that incorporate legal issues. *Digital Investigation*(3), 29-34.

Osborne, G., Turnbull, B., & Slay, J. (2010). The 'Explore, Investigate and Correlate' (EIC) conceptual framework for digitalforensics Information Visualisation. *International Conference on Availability, Reliability and Security*, (pp. 630 - 634).

Schofield, D., & Fowle, K. (2013). Visualising Forensic Data : Evidence (Part 1). *Journal of Digital Forensics, Security and Law, Vol. 8(1)*, 73-90.

Teerlink, S., & Erbacher, R. F. (2006). Foundations for visual forensic analysis. *7th IEEE Workshop on Information Assurance.* Westpoint, NY: IEEE.