The Potential for Cross-Drive Analysis Using Automated Digital Forensic Timelines

Jonathan Patterson[1], Christopher Hargreaves[2]

Centre for Forensic Computing
Department of Engineering and Applied Science
Cranfield University
Shrivenham
SN6 8LA
United Kingdom
[1]j.patterson@cranfield.ac.uk
[2]c.j.hargreaves@cranfield.ac.uk

Abstract

Cross-Drive Analysis (CDA) is a technique designed to allow an investigator to "simultaneously consider information from across a corpus of many data sources" [1]. Existing approaches include multi-drive correlation using text searching, e.g. email addresses, message IDs, credit card numbers or social security numbers. Such techniques have the potential to identify drives of interest from a large set, provide additional information about events that occurred on a single disk, and potentially determine social network membership.

Another analysis technique that has significantly advanced in recent years is the use of timelines. Tools currently exist that can extract dates and times from the file system metadata (i.e. MACE times) and also examine the content of certain file types and extract metadata from within. This approach provides a great deal of data that can assist with an investigation, but also compounds the problem of having too much data to examine.

A recent paper adds an additional timeline analysis capability, by automatically producing a high-level summary of the activity on a computer system, by combining sets of low-level events into high-level events, for example reducing a setupapi event and several events from the Windows Registry to a single event of 'a USB stick was connected'.

This paper provides an investigation into the extent to which events in such a high-level timeline have the properties suitable to assist with Cross-Drive Analysis. The paper provides several examples that use timelines generated from multiple disk images, including USB stick connections, Skype calls, and access to files on a memory card.

1.0 Introduction

Cross-Drive Analysis (CDA) is a technique designed to allow an investigator to "simultaneously consider information from across a corpus of many data sources"[1]. This paper reports on an initial investigation into how automatically

generated high-level timelines such at those produced in [2] could be used for Cross-Drive Analysis.

The paper makes the following contributions: it demonstrates that Cross-Drive analysis can be performed on automatically generated high-level timelines; it shows how visualisations, filtering and grouping of such timelines can be used as part of investigations; it also demonstrates that when low-level timelines are being produced it is extremely useful to preserve information such as evidence source.

The paper is structured as follows: Section 2 provides background to the research and also discusses related work. Section 3 provides the methodology for the research and justifies the synthesised scenario based approach used. Section 4 provides the results of three case studies: USB connections, Skype use, and access to files on a memory card. Section 5 evaluates the research and discusses limitations of the research in its current form, and Section 6 provides the conclusions and discusses future work.

2.0 Background

This section provides a background discussion of Cross-Drive analysis and timeline analysis techniques.

2.1 Background

Complex digital forensic investigations can involve multiple computer systems, potentially from multiple users located in multiple locations. This introduces challenges such as the volume of data, and the number and variety of devices to be examined. There are multiple approaches to addressing these challenges, including technological e.g. parallel processing [3], and procedural e.g. 'Triage'[4].

A specific technique that not only attempts to address the problem of multiple drives, but also exploits this property to recover more relevant evidence is Cross-Drive Analysis.

2.2 Related Work: Cross-Drive Analysis

Cross-Drive Analysis is a technique designed to "allow an investigator to simultaneously consider information from across a corpus of many data sources" [1]. The technique is presented as offering several benefits: automatic identification of 'hot drives'; improving the analysis of single drives; identification of social network membership; and unsupervised social network discovery. These are discussed below.

Automatic identification of 'hot drives'

This involves processing multiple drives in order to identify those from a large collection that are of particular interest, and therefore which should be given the highest priority for examination. The example provided in [1] relates to the US legislation FACT-ACT: which details the requirement to remove consumer information from IT equipment prior to its disposal. The example goes on to use Cross-Drive Analysis in conjunction with Forensic Feature Extraction (FFE) on a set of disks to recover particular forms of consumer information (e.g. social

security numbers) and to identify those drives that have the most features of this type.

Improving the analysis of single drives
In addition to identifying a subset of drives of interest, through the analysis of multiple drives it is possible that the analysis of a single drive can be improved. The example provided in [1] is to determine information that can be safely ignored on drives examined in future, e.g. a 'stop list'. While not explicitly discussed in [1], it is also possible to use data obtained on a drive or set of drives to establish information about another device, for example, to use an iPhone backup located on a seized drive to recover data that was stored on an iPhone that may not be in the possession of the investigator [5].

Identification of social network membership and unsupervised social network discovery
Cross-Drive Analysis can also be used to determine the nature of social networks e.g. determining if is a drive connected in some manner to another set of drives.

The artefacts that are correlated across drives in [1] are termed 'Features' and are pseudo-unique identifiers which have "sufficient entropy such that within a given corpus it is highly unlikely that the identifier will be repeated by chance". Example features include email address, message ID and subject, date and time stamps, cookies, US social security numbers and credit card numbers.

The feature extraction approach in [1] attempts to address the problem of "improper emphasis on document recovery" i.e. that relevant data on a disk may not necessarily be a file, but may still be useful. However, while feature extraction does not focus on documents, it does still to a certain extent focus on content e.g. credit card numbers or email addresses. Another approach may be instead to focus on event reconstruction.

2.3 Related Work: Timeline Analysis

It is difficult to discuss event reconstruction without discussing timelines. Early timeline analysis involved the recovery of file system dates and times (typically Modified, Accessed, Created, and Entry Modified (MACE) times) in order to determine user activity [6], [7]. More advanced techniques now supplement these basic file times with dates and times contained within files, for example, file times extracted from Link Files, last updated times for Windows Registry keys, URL access times from index.dat, or log entries from setupapi files [8-10]. Use of this technique recovers a vast number of events from computer systems. For example a system that has been in in use for just a few months can produce millions of these 'system' or 'low-level' events.

Reference [2] builds on top of this low-level event extraction and describes the development of a Python based prototype that can automatically reconstruct higher-level, more human understandable events. For example, a low-level event for a specific entry in the SAM registry hive, the creation of a specific

user folder in /Users and the creation of a NTUSER.DAT in that folder can be used to infer the creation of a user account on Windows 7. This automated high-level event reconstruction is achieved through a plug-in based system of 'analysers' that scan the low-level timeline looking for patterns of one or more low-level events that are used to infer the high-level event. In the developed system provenance of the inferred high-level event is preserved in the exported XML timeline which maintains references to the low-level events that were used to identify the high-level event. This includes preserving the low-level events that were searched for and found (supporting artefacts) and also any low-level that were searched for but were not found (contradictory artefacts). Furthermore, all the low-level events are preserved in a SQLite database and each low-level event contains provenance values that describe how this event was extracted e.g. an entry in a log file, an offset in a binary file or a SQL query and row number. This approach provides full traceability from a high-level, human understandable event back down the raw data that caused the event's supporting artefacts to be originally extracted.

This automated analysis approach produces a much smaller set of events that is determined by the number of 'analysers' that are run. Reference [2] highlights that this move from "hundreds of thousands of low-level events to a few hundred, human understandable events may open up new possibilities for visualisation of data from digital forensic investigations and enable the development of tools with much greater analysis capabilities". The example provided is the exporting of the timeline into Timeflow format, which is a timeline visualisation software from Flowing Media [11].

The need to move beyond forensic tools that focus on finding files as pieces of evidence is highlighted in [12], which states that today's tools are difficult to use to "reconstruct a unified timeline of past events or the actions of a perpetrator". The automated approach to high-level timeline reconstruction in [2] is designed to achieve exactly that and could allow more advanced analysis and visualisation techniques to be used.

While Cross-Drive Analysis in [1] focused on correlating content across multiple drives it is possible that the same could be done with the high-level events generated in [2]. This is the focus of the research reported in this paper.

3.0 Methodology

3.1 Aim

The aim of this research is to investigate how the automatically generated high-level timelines produced in [2] could be used for Cross-Drive Analysis.

3.2 Research Method Overview

The research method used in this paper is to use a series of examples to demonstrate how automatically generated high-level timelines could be used for Cross-Drive Analysis. The three examples used demonstrate the three potential benefits to digital investigations described in Section 2.2 (identifying drives of interest, improving analysis of a single device, and determining social network membership).

Each example involves a description of a fictitious scenario, the synthesis of a small dataset that contains artefacts related to the scenarios, automated generation of timelines for the resulting disk images, and the subsequent analysis using a visualisation tool. Each of these stages are discussed in more detail in the following sub-sections.

3.3 Scenarios

As discussed earlier three examples are used to demonstrate the use of high-level timelines for Cross-Drive Analysis.

Scenario 1: Identification of 'hot drives'
In this scenario a personal USB drive has been found on the floor of an office containing four computers. The site is a secure facility and staff within the office are permitted to use only regulation USB sticks to transfer data. The non-approved USB drive that was found contained restricted documents and therefore the person responsible for the USB stick is in violation of a number of regulations.

Scenario 2: Enhancing the analysis of a single disk/device
In this second scenario a computer and camera phone have been seized from a suspect who has allegedly been making indecent images with their the phone and storing them on their computer. Examination of the mobile phone memory card has not revealed any relevant images. The phone cannot be examined as it has been irreparably damaged.

Scenario 3: Determining social network membership
The third scenario involves an investigation into the cause of a recent shopping centre riot. Four suspects have been identified that may have been communicating with each other via Skype. Each of the suspect's computers are to be examined to determine who was involved with the organisation of the riot.

3.4 Data Generation

For each of the scenarios Virtual Machines (VM) were created using VMware Workstation. Each VM had a small 20Gb virtual hard disk created and Windows 7 Enterprise Edition was installed. Actions consistent with the scenarios were carried out on the VMs and documented. Finally the virtual hard disks were preserved as flat dd style images for later analysis.

3.5 Timeline Generation

The disk images from each of the scenarios were analysed using the timeline tool described in Section 2.3. The low and high-level timelines were preserved including the XML timeline and the Timeflow representation.

3.6 Analysis and Visualisation

In each case the 'analysis' was actually performed using Timeflow [11]. This tool from Flowing Media provides grouping and filtering capabilities. 'Grouping' determines how events are split into horizontal groups in the timeline or how events are colour coordinated. 'Filtering' allows keywords or event types to be hidden or shown in the timeline.

4.0 Results

This section details the results of the analysis of the three scenarios described in the previous section.

4.1 Scenario 1: Identifying Drives of Interest

Manual Analysis

In this scenario since a prohibited USB stick was found in an office containing four computers, each of the systems need to be analysed to identify if the USB stick was connected. Forensically sound examination of these systems should involve imaging and an analysis of the disk image. To identify if the specific USB stick was connected, the setupapi.app.log should be examined in conjunction with the Windows Registry.

Automated Timeline Generation

A case file was created for this scenario including the four disk images from the four computers. After the low-level timeline generation, just over 2.6 million events were produced. Such data can be searched using tools such as Grep for keywords, but even this relatively small number of events is difficult to visualise in a useful way. The high-level event analysis process was run using 32 analysers, which produced a high-level timeline consisting of 298 high-level events. The Timeflow version of this high-level timeline was loaded and grouped by 'Evidence Source' as shown in Figure 1. The timeline was then analysed simply by filtering by the term 'USB' as shown in Figure 2. This filtered the results so that 5 events remained, 4 of which are artefacts from the construction of the VM. However, the remaining event clearly shows the connection of a Freecom Databar USB device connected to Computer 2 (Figure 3). While it cannot be shown in the Timeflow visualisation, the provenance of this high-level event is also preserved, including the low-level events that were used to infer this high-level event (shown in Figure 4) and the location of the original data that was used to generate the low-level events, in this case the line number of the setupapi.app.log and the offset in the Registry hive of the raw data used to extract the Registry keys.

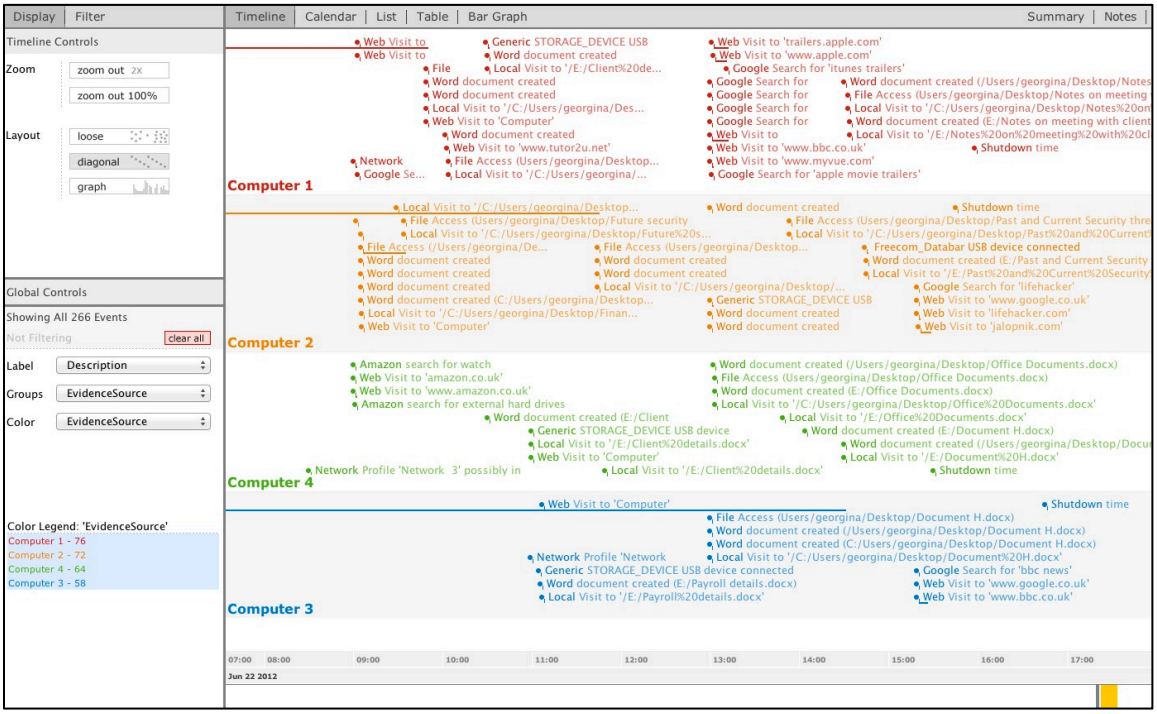

Figure 1: Timeline showing high-level events relating to each computer system that is suspected of connecting a prohibited USB device. The events are grouped by evidence source showing the events that occurred on each computer.

Figure 2: A simple filter applied to the timelines for '*usb*'

Figure 3: The timeline that results from the 'usb' filter. This demonstrates that a Freecom Databar was connected to Computer 2.

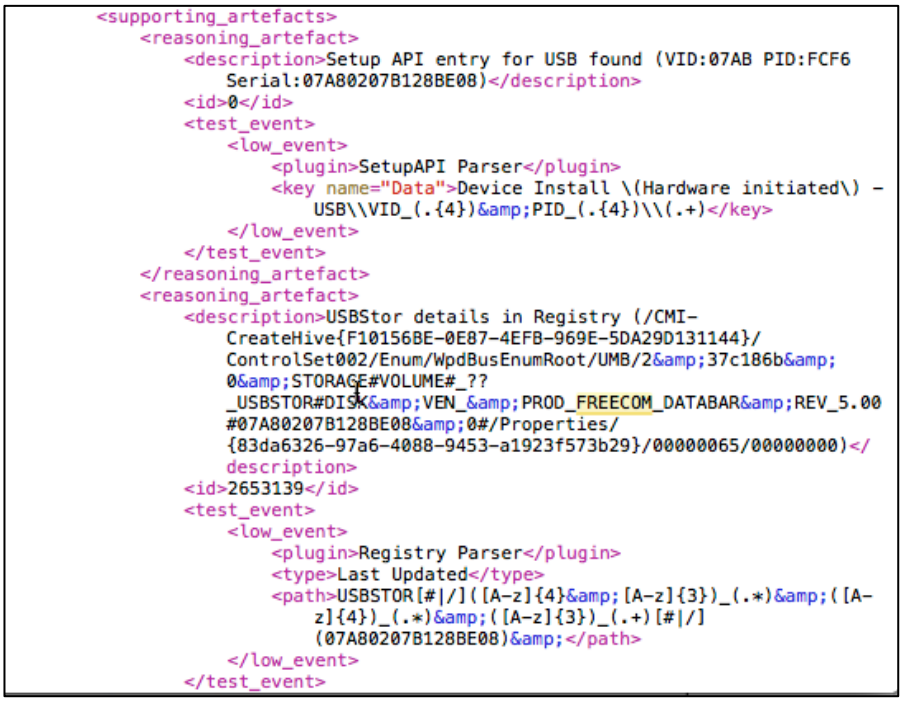

Figure 4: Details for the identified high-level event, including the low-level events that were used to infer it. This can also be used to inspect the SQLite database to view the low-level events and the raw data that caused the low-level event to be created.

4.2 Scenario 2: Enhancing the analysis of a single disk/device

Manual Analysis

In this scenario since a computer and a memory card have been seized for examination, both require analysis. No files of interest have been found on the memory card, but it is still necessary to determine if the images were stored on the memory card, as it is this card that was found in the camera phone that is believed to have produced the images. Manual analysis in this case involves determining the connection of the memory card to the computer and establishing the drive letter that was used at the time. Following on from this it may be possible to use artefacts such as Link Files or the Windows Search database to determine information about files that previously existed on the memory card.

Automated Timeline Generation

Use of the timeline generation tool produced just over half a million low-level events and after running the high-level timeline analysis, 112 high-level events were produced. The high-level events were loaded into Timeflow (Figure 5) and subsequently sorted by device (Figure 6), showing events on Computer 1, and two drives with their volume serial numbers displayed. (C:\ (18e5-a8ac), E:\ (c595-

ce08)).

By further filtering the results (shown in Figure 7), it can be seen that there was an image created on the memory card in 2010 by the name of '*03042010003.jpg*'. The timeline shows a connection of a Nokia S60 device followed by an access of a file of the same name and path as that on the memory card.

The process of mapping the Nokia USB device to drive E:\ has not yet been fully automated and still requires a number of manual steps involving the setupapi.app.log and the computer's system Registry.

In any case, this example demonstrates that additional information about a device (in this case the dates of files created on a memory card) can be obtained by examining a different device (a PC) that the memory card has interacted with.

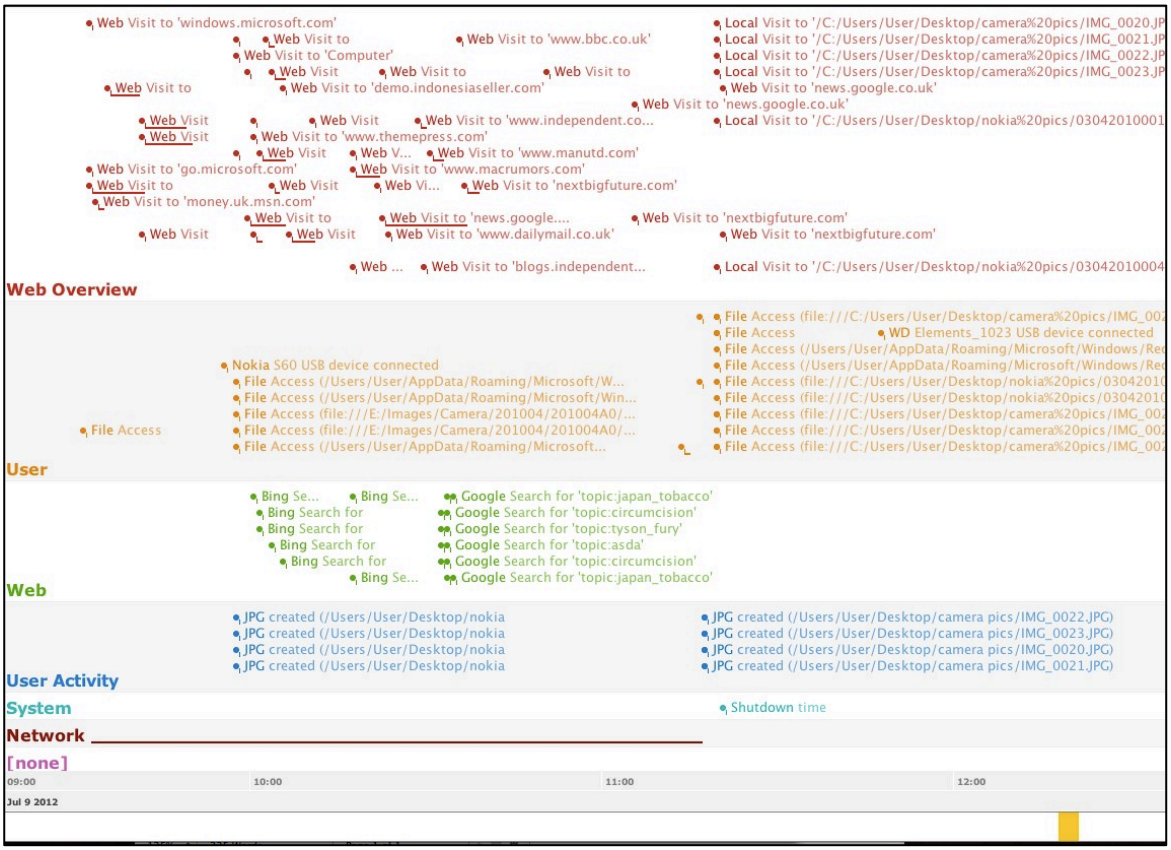

Figure 5: Timeline of high-level events from the memory card example

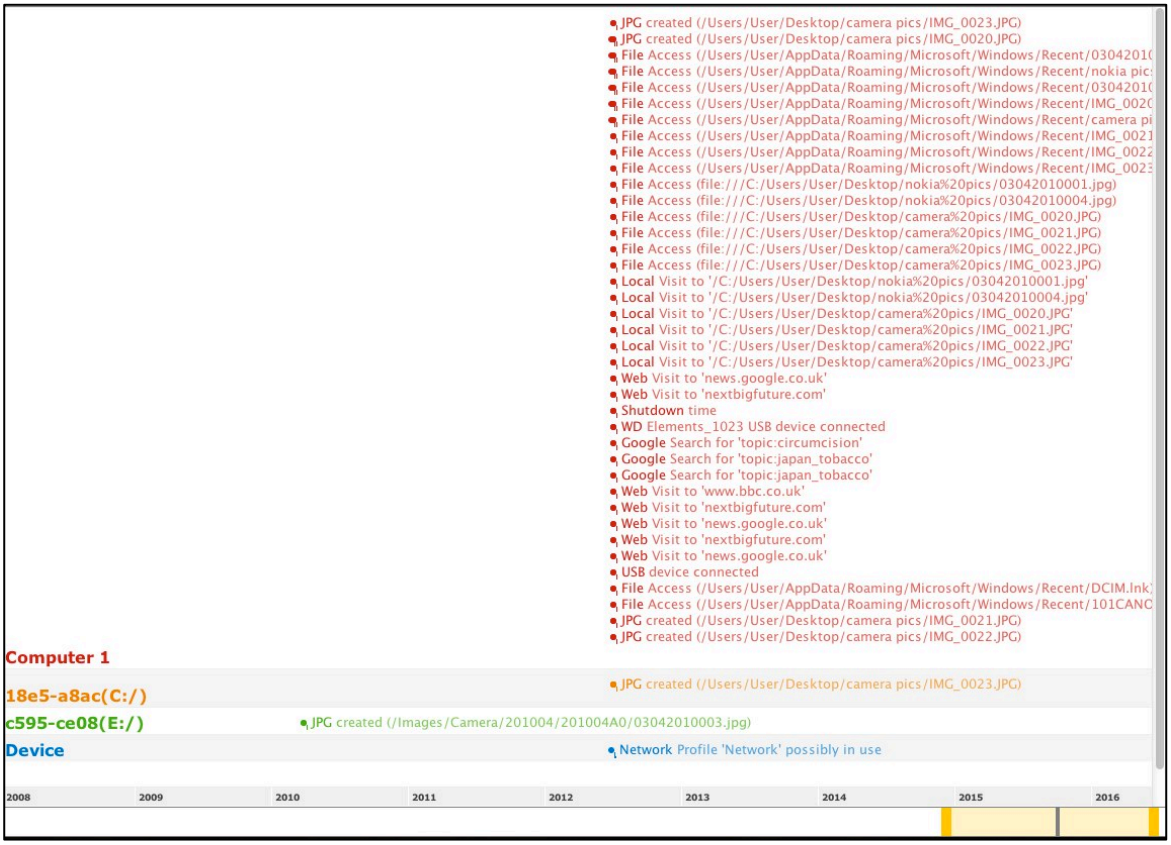

Figure 6: Timeline of high-level events from the memory card example

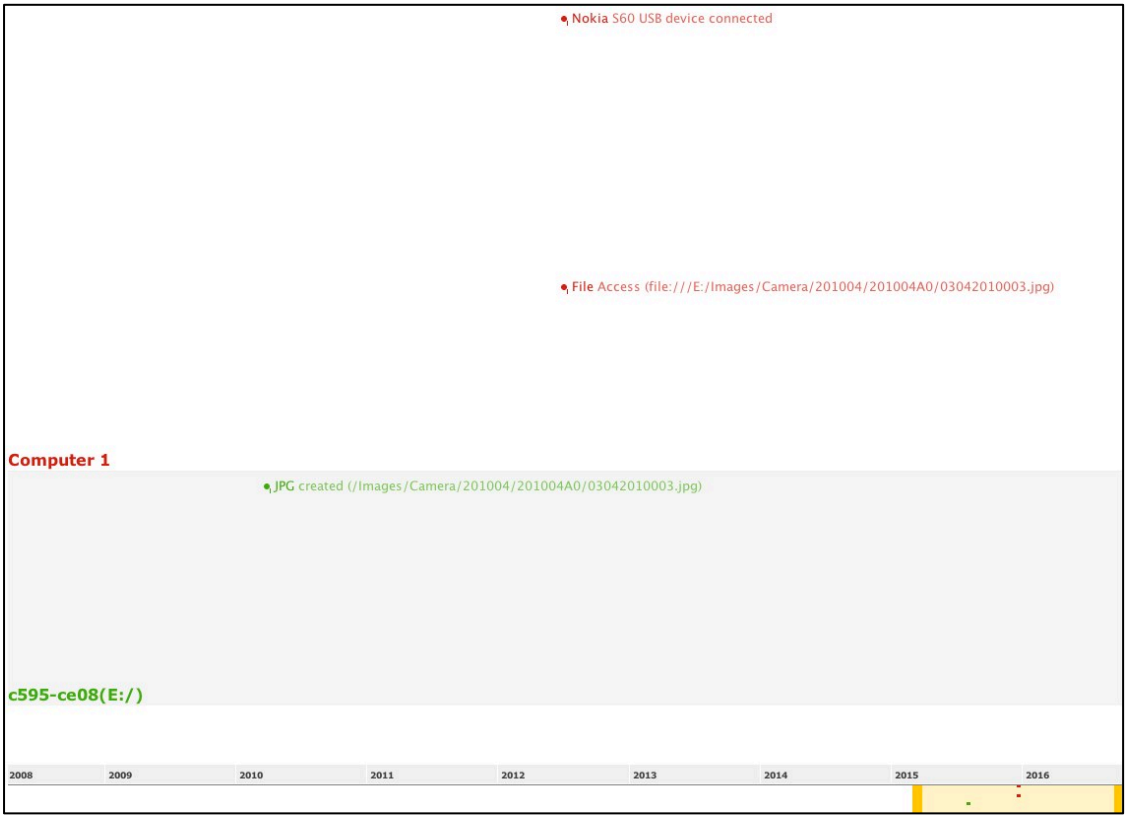

Figure 7: Filtered results showing connection of Nokia USB device and details relating to 03042010004.jpg. The creation date and time of the image on the E:\ drive have been extrapolated from the link file created when the file was accessed on the computer.

4.3 Scenario 3: - Determining Social Network Membership – Skype Example

Manual Analysis

In this example of an investigation into a shopping centre riot, the computers of four suspects are being examined. Manual analysis in this case would involve an examination of Skype databases and manual correlation to establish connections and potentially the construction of a social network diagram.

Automated Timeline Generation

After the examination of all four disks, just over 2.9 million low-level events were generated and 672 high-level events. The high-level events were then loaded into Timeflow and sorted by 'Evidence Source' (Figure 8). The events were then filtered using the term '*skype contact request*' (shown in Figure 9), reducing the

number of events to 8. From these results it can be seen in Figure 10 that users McDee, Ferris and Dylan were communicating with each other prior to the date of the riot (5th July), with user Smith only joining the Skype social network after the event.

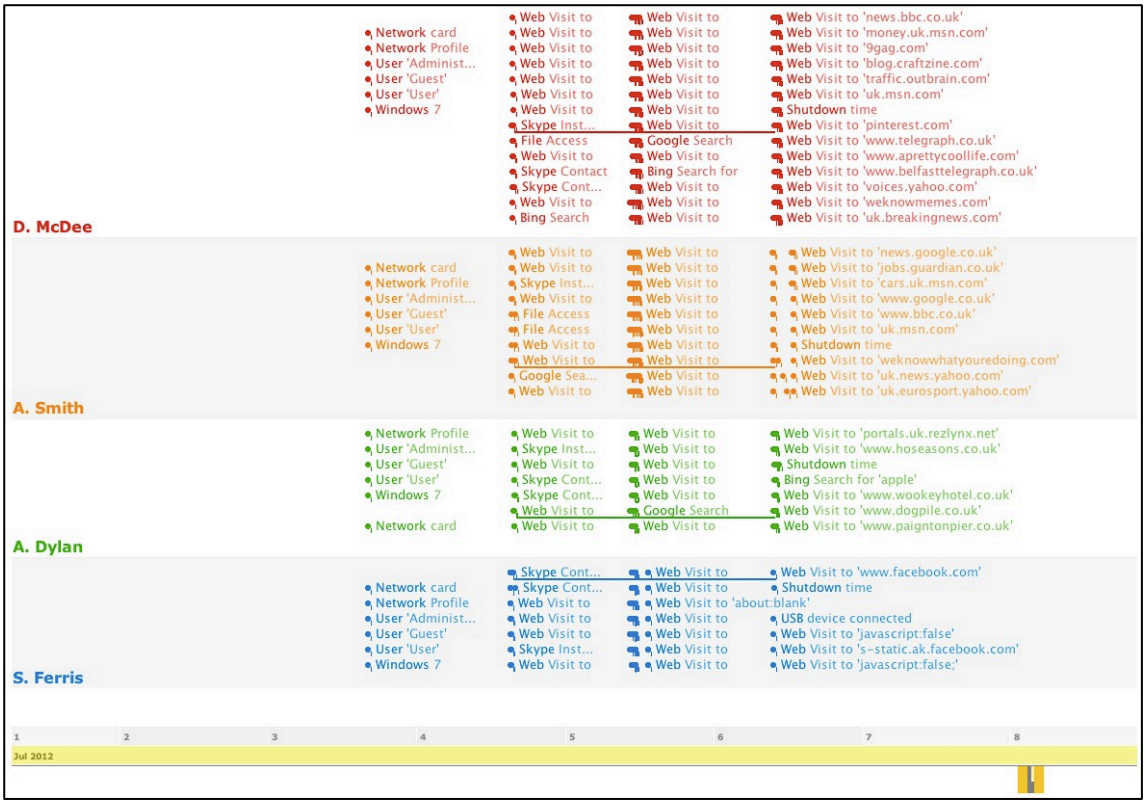

Figure 8: Timeline of high-level events sorted by '*evidence source*'

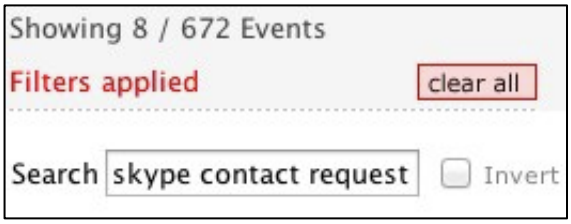

Figure 9: Filter for '*skype contact request*'

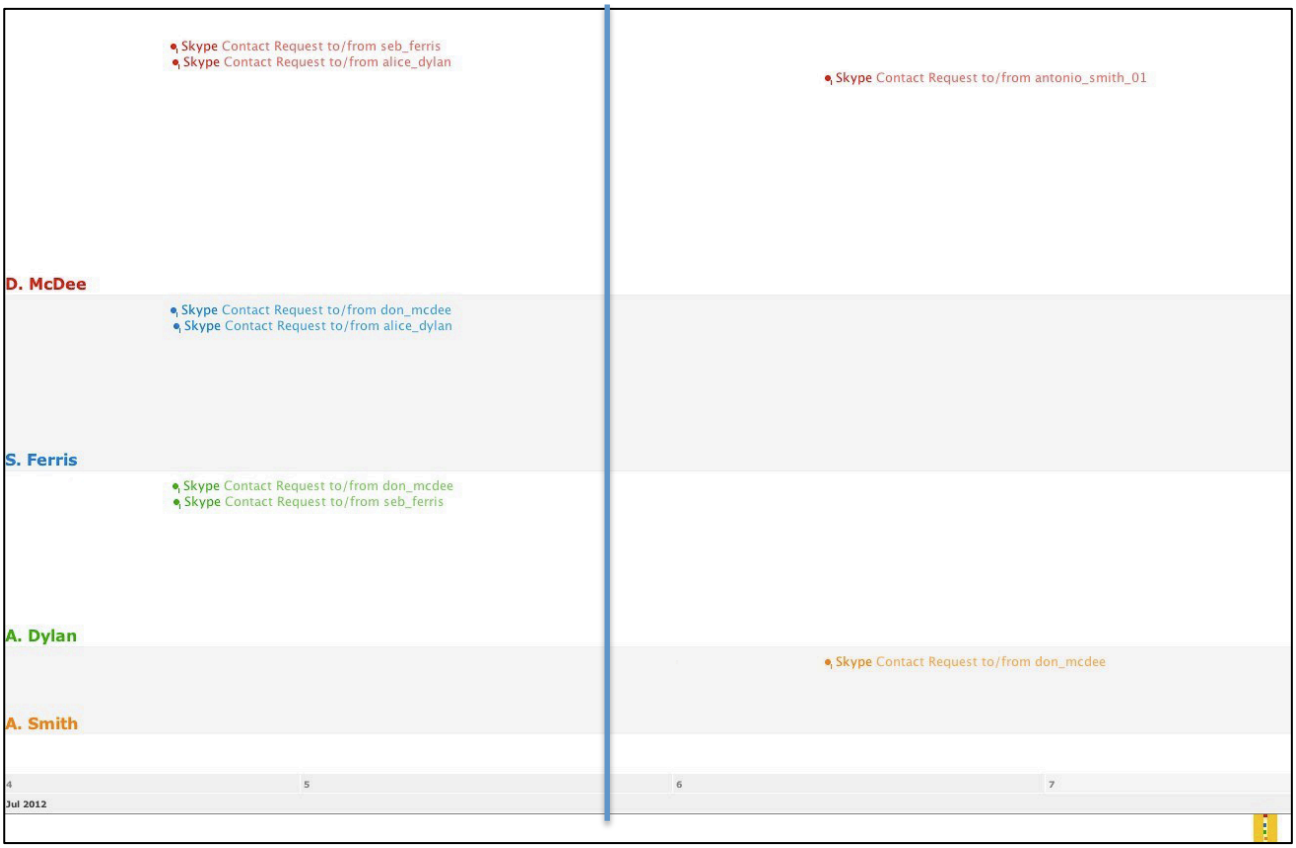

Figure 10: Timeline of Skype contact requests before and after the riot. The vertical line has been added to the diagram to highlight the date of the riot.

The advantage of using a timeline visualisation tool to display a social network is that the evolution of the network can be more easily followed. With traditional social network diagrams such as that shown in Figure 11, the social network is viewed as a static entity and therefore does not communicate the order in which the network was created, or the structure of the network at any given time. This makes it difficult to answer specific questions such as those discussed in this scenario i.e. to understand which users were members of the network prior to an event taking place.

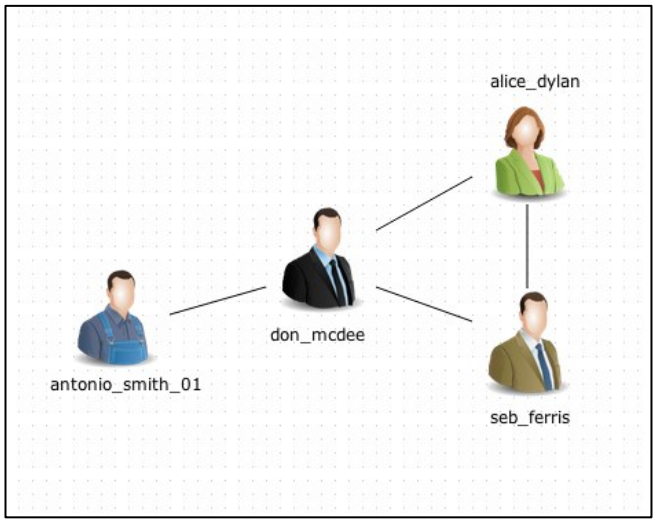

Figure 11: An example static social network diagram of the people in the shopping centre riot scenario. It would be possible to depict the evolution of the social network using an animation, which cannot be shown here, or a sequence of images, which does not scale well for larger networks.

5.0 Evaluation

The results shown in the previous section clearly demonstrate the potential for high-level timelines being used for Cross-Drive Analysis. The first example shows that high-level timeline generation can make the analysis process extremely simple in terms of identifying drives of interest. The second example demonstrates that by examining multiple systems, it is possible to infer additional information about a system that may not have evidence of actions on it. This also applies to devices that may not even be in the possession of the investigator. The final example demonstrates the ability to determine social network membership and also introduces the time dimension so that the evolution of the social network over time can be seen.

However, there are limitations. The data sets are particularly simple in terms of the size of the disk and the complexity of the disk images (the low number of high-level events recovered and the amount of 'background noise'). This is particularly apparent with the first example where a very simple example has been chosen. It is necessary to develop a much more detailed scenario with a complex criteria for drives of interest to provide a through test of the benefits of this approach.

The second example shows that examination of multiple devices can be used to improve the analysis of a single device; in the example provided this was to identify information that could not be recovered from a specific device. However, the application in [1] was not demonstrated i.e. the use of multiple drive analysis to exclude data from future analyses. In the case of high-level timelines, this would involve the creation of a new 'whitelist of events'. These events would be those that are as a result of a Windows installation, for example the creation of the

pictures in the ‘Sample Pictures’ folder, which may be of use as supporting artefacts for a ‘Windows Installation’ event, but as high-level events on their own are of limited interest.

While the third example does demonstrate social network membership it has not taken full advantage of the high-level abstraction offered by the automated timeline analysis. A more extensive scenario in this case would involve more participants that are communicating using a variety of technologies. It should still be possible to produce an analyser that identifies communication between participants using any of these communication methods and how communication patterns and methods change over time. This would exploit the benefits offered by the timeline analysis to a much greater extent.

Despite these the limitations the research has demonstrated use of high-level timelines for Cross-Drive Analysis and highlighted some interesting further work to be carried out to more extensively investigate their use. Furthermore the paper has clearly demonstrated an analysis tool that is not simply a view of files (in table, tree or gallery form) and presented data in a novel way that can assist in an investigation though providing an abstraction layer that would allow an investigator to focus on finding events of interest rather than the extraction of data and reconstruction of what the system has been used for.

While the tool used in this case to generate high-level timelines makes every effort to function in a forensic manner, particularly preserving the provenance of high-level events so they are traceable down to the raw data, it is important to highlight that it is not intended as a replacement for a full forensic analysis. The work in this paper is intended to be used to highlight drives of interest that may need to be prioritised over others or to infer links that may be missed when a manual process is required.

6.0 Conclusions and Future Work

The aim of this paper was to investigate the potential for high-level timelines for Cross-Drive Analysis and that has certainly been addressed, although there are many limitations to the research that will need to be addressed in a future paper, as discussed in the previous section.

In addition to the future work described in the previous section that enhanced the scenarios used to make them more suited to high-level timeline based Cross-Drive Analysis, there are also new scenarios that provide examples that are extremely difficult to determine from a manual analysis of multiple drives, for example, a trip being planned and booked by multiple suspects, on different forms of transport, using different computers and devices. Also multiple people or devices being used to purchase items that individually are not of interest, but may be in combination.

Finally, as digital devices are used more extensively in everyday life the amount of evidence of peoples' actions that these devices contain increases. In addition, the number of devices in use by individuals is also increasing. The need for techniques that are able to examine multiple devices and produce a coherent report of activity across them is likely to increase and therefore is an area that requires much future work.

References

[1] S. L. Garfinkel, "Forensic feature extraction and cross-drive analysis," *Digital Investigation*, vol. 3, no. 1, pp. 71–81, 2006.

[2] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Digital Investigation*, vol. 9, no. 1, pp. 69–79, Jun. 2012.

[3] V. Roussev, "Breaking the performance wall: The case for distributed digital forensics," presented at the Proceedings of the 2004 Digital Forensics Research Workshop, 2004.

[4] M. K. Rogers, J. Goldman, R. Mislan, and T. Wedge, "Computer Forensics Field Triage Process Model," presented at the Conference on Digital Forensics, Security and Law, 2006.

[5] M. Bader and I. Baggili, "iPhone 3GS Forensics: Logical analysis using Apple iTunes Backup Utility," *Small Scale Digital Device Journal*, vol. 4, no. 1, pp. 1–15, Oct. 2010.

[6] B. Carrier, "File Activity Timelines," *bandwidthco.com*, 14-Jun.-2003. [Online]. Available: http://bandwidthco.com/whitepapers/compforensics/fsanalysis/File%20Activity%20Timelines.pdf. [Accessed: 14-Feb.-2012].

[7] S. Bunting, *EnCase Computer Forensics, Includes DVD*. Sybex, 2007.

[8] J. Olsson and M. Boldt, "Computer forensic timeline visualization tool," *Digital Investigation*, vol. 6, no. 1, pp. S78–S87, 2009.

[9] K. Guðjónsson, "Mastering the Super Timeline with log2timeline," pp. 1–84, Feb. 2010.

[10] R. Carbone and C. Bean, "Generating Computer Forensic Super Timelines Under Linux," pp. 1–136, Oct. 2011.

[11] Flowing Media, "TimeFlow," *github.com*. [Online]. Available: https://github.com/FlowingMedia/TimeFlow/wiki/. [Accessed: 31-Jul.-2012].

[12] S. Garfinkel, "Digital forensics research: The next 10 years," *Digital Investigation*, vol. 7, no. 1, pp. 64–73, Sep. 2010.