A USER-ORIENTED NETWORK FORENSIC ANALYSER

D. Joy[1], F. Li[1], N.L. Clarke[1,2] and S.M. Furnell[1,2]
[1]Centre for Security, Communications and Network Research (CSCAN)
Plymouth University, Plymouth, United Kingdom
[2]Security Research Institute, Edith Cowan University, Western Australia
info@cscan.org

Abstract

Network forensics is becoming an increasingly important tool in the investigation of cyber and computer-assisted crimes. Unfortunately, whilst much effort has been undertaken in developing computer forensic file system analysers (e.g. Encase and FTK), such focus has not been given to Network Forensic Analyser Tools (NFATs). The single biggest barrier to effective NFATs is the handling of large volumes of low-level traffic and being able to exact and interpret forensic artefacts and their context – for example, being able extract and render application-level objects (such as emails, web pages and documents) from the low-level TCP/IP traffic but also understand how these applications/artefacts are being used. Whilst some studies and tools are beginning to achieve object extraction, results to date are limited to basic objects. No research has focused upon analysing network traffic to understand the nature of its use – not simply looking at the fact a person requested a webpage, but how long they spend on the application and what interactions did they have with whilst using the service (e.g. posting an image, or engaging in an instant message chat). This additional layer of information can provide an investigator with a far more rich and complete understanding of a suspect's activities. To this end, this paper presents an investigation into the ability to derive high-level application usage characteristics from low-level network traffic meta-data. The paper presents a three application scenarios – web surfing, communications and social networking and demonstrates it is possible to derive the user interactions (e.g. page loading, chatting and file sharing) within these systems. The paper continues to present a framework that builds upon this capability to provide a robust, flexible and user-friendly NFAT that provides access to a greater range of forensic information in a far easier format.

Keywords

Digital Forensics, NFAT, Network Forensics, Analysis, Correlation, Visualisation

INTRODUCTION

In recent years, the number, scale and variety of cyber enabled crimes increases on a yearly basis (https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/246749/horr75-summary.pdf). Indeed, criminals can use their computers not only for traditional crimes (e.g. fraud and money laundry) but also for newly formed attacks (e.g. phishing). By utilising well-established computer forensic tools (e.g. FTK) and techniques, investigators can thoroughly examine suspect's computing devices for evidence of wrong doings (Acessdata, 2013; DFRWS, 2001; Garfinkel, 2010; ISO 27037, 2012). However, technically competent criminals can take the advantage of the existence of anti-forensic tools and techniques (such as overwriting data and metadata, exploiting the vulnerabilities in computer forensic tools and detecting the presence of computer forensic tools) to wipe their computers, leaving little trace for forensic investigators to find (Garfinkel, 2007; Pajek and Pimenidis, 2009).

With the aim of providing additional assistance against computer enabled crimes, network forensics tries to solve the problem by examining network traffics which user has less control over. Nonetheless, being able to trace user activities in an effective and timely manner can be extremely challenging due to several factors, such as the sheer volume of network traffic and the ability of existing forensic tools (Pilli *et al.*, 2010). Indeed, annual global IP traffic would surpass the zettabyte threshold in 2016 according to a Cisco's prediction (Cisco, 2014). Regarding the tools, limited level of analysis and artefact extraction can be performed by network forensic examiners in the past despite a selection of tools could have been utilised; more recent advances in Network Forensic Analysis Tools (NFATs) are beginning to incorporate common object identification and extraction. For example, the tools such as Xplico Open Source Network Forensic Analysis Tool (Xplico, 2007) and PyFlag (Cohen, 2008) can reconstruct a TCP session, extract the payload and perform a data carve against it to identify if any files are contained within it. Nonetheless, these tools can be utilised to view data only at TCP/IP levels of data abstraction but not at the user activity levels. As a result, a tool that can process larger amount of traffic in a timely manner and also offer additional analysis upon user activities is required for the modern network forensics domain.

This paper proposes a novel network forensic analysis tool capable of generating high level information by analysing raw network traffic from an investigation perspective. This high-level information goes beyond simply identifying which servers have been contacted and data carving the payload but provides the investigator with an understanding of what the user was doing whilst accessing those services. For example, the approach will not merely identify what website is being visited but for what purpose and what user interactions take place.

The paper is structured as follows. Section 2 presents a literature review of the current tools and techniques used in network forensics. Section 3 discusses an experiment into the analysis of network-level traffic and the identification of application-level user interactions. A framework for the User-Oriented Network Forensic Analysis Tool (UO-NFAT) along with an illustration of the tool's dashboard in presented in Section 4. Section 5 contains the conclusions and the future directions of the research.

~~LITERATURE REVIEW~~EXISTING WORK ON NETWORK FORENSICS

Network forensics is defined as the "capture, record and analysis of the network events in order to discover the source of security attacks or other problem incidents" (Pilli *et al.*, 2010). As every single time an attack takes place in a network, from that point onwards, the network traffic becomes an essential piece of the evidence. By utilising network forensic tools, investigators can identify potential evidence from existing network traffic logs when incidents are reported. However, network forensic investigations always have to deal with two major challenges: the quantity problem and the complexity problem (Merkle, 2008). The volume of data in any network these days is enormous (reference). Also, filtering the traffic in an intelligent manner that seeks to reduce the complications of the entire analysis process is a challenging task (Mukkamala, 2003). Furthermore, an analysis of some of the most commonly used network forensic analysis tools was conducted based on the type of appliance, their functionalities, whether they are commercial or open source products. These tools are NetIntercept (NIKSUN, 2010), NetDetector (NIKSUN, 2013), NetWitness (RSA, 2012; SC Magazine, 2012), SilentRunner (AccessData, 2010), InfiniStream (Netscout, 2012), Solera DeepSee 5150 (Solera Networks, 2009), OmniPeek (WildPackets, 2012), NetResident (TamoSoft, 2009), PyFlag (Cohen, 2008), NetworkMiner (NETRESEC, 2013), Xplico Open Source Network Forensic Analysis Tool (Xplico, 2007) and Iris Network Traffic Analyser (Lyonesse Software, 2008). It's been observed that many of the commercial network forensic analysis tools are very expensive and they do not serve the requirements that are needed to be fulfilled in order to carry out the investigation in a cost and time effective and an efficient manner. A good network forensic analysis tool should exhibit the capability to:

- Identify the protocols and reconstruct the protocol stream in a reliable manner;
- Reduce the large volume of data by using methods for extracting items of interest such as usernames, passwords, top talkers etc.;
- Recover digital objects such as e-mail messages, documents, images etc. from the network traffic before analysing for evidences;
- Logically search for keywords and perform regular expression searches;
- Create the audit log (to document all the examiner's activities);
- Operate in read-only mode during examination (tool should not be capable of accessing any content from web servers while reconstructing and displaying any webpage; the reconstruction must only be made from the captured network traffic);
- Perform case management (a feature exhibited by computer forensic tools);
- Understand what and how a service is being used for;
- Correlate evidence between services to aid in creating a meaningful chronology of events;
- Use advanced data analytics to predict the likelihood of future events.

Experimenting with some of the open source NFATs showed that there are limitations in the analysis of the raw network traffic, correlation and visualisation of the generated high level information and that there is still scope for further research in this area accompanied by application of techniques such as artificial intelligence and advanced analytics.

Merkle (2008) proposed the concept of implementing intelligence into network forensic analysis leading to the formation of an automated tool, making use of computational intelligence. The research describes the idea of integrating the tools used in different stages of a forensic investigation in to a single system utilizing computational intelligence in order to reduce human interaction. The model used the existing search techniques used for network forensic analysis and combined them with the computational intelligence techniques in order to increase the efficiency of the entire system. Computational intelligence was utilised in the data collection stage. The data set was created by implementing evolutionary computational algorithms on attacker systems generating variations of commonly known attacks against a honeynet. Processes such as analysing the alert data manually which consumed more time and are error-prone were the ones chosen for automation. In a later stage, these alert data, depending on their characteristics and their relationship to the hypothesis generated in an early stage, were organized into clusters by making use of evolutionary computation.

Another work used the application of fuzzy sets into forensic investigation system in order to derive a set of if-then rules reducing the effort of forensic analysis with the aim to overcome two of the existing constraints in forensic data analysis – accuracy and the ability to understand the output easily by investigators (Stoffel *et al.*, 2010). The methodology used in their work was to divide the raw forensic data into groups, called clusters. Then membership functions were extracted from those data fuzzy inference systems were created. The data used for testing were of conventional crime incidents of robberies and residential burglaries. The proposed methodology was tested against an artificially created datasets with some hidden internal structures and the experiments resulted in identifying the hidden structures and by not discovering any structures that do not exist in the original dataset with a satisfactory level of accuracy

Another paper proposed the use of AI techniques for offline intrusion analysis by aiming to rank the input features in order to reduce the inputs received in the collection and detection phases. Proposed work attempted using artificial neural networks (ANN) and support vector machines (SVM) techniques on data taken from MIT's Lincoln Labs which was developed for offline intrusion detection system evaluations by DARPA in 1999 and resulted in concluding that SVM techniques can perform better than ANNs in scalability, training time and prediction accuracy (Mukkamala, 2003). It points out that making use of artificial intelligent techniques can increase the performance level of network forensic investigation by reducing the volume of collected data by eliminating the unimportant events.

While network forensics is an essential part of the investigation, there is another component of the organisational network called Security Information Event Management (SIEM), which acts as a core tool in detecting real-time threats and performing log management for forensic investigation, which has objectives similar to that of network forensics. SIEM is a vital part that interconnects with other components of the organisational infrastructure to perform real-time threat detection, acting as a proactive system and trying to achieve similar analysis goals as network forensics. Recent research in this field proposes the adaptation of security analytics and threat intelligence from external sources in order to make SIEM systems more dynamic and preventive. In order to increase the efficiency in the analysis and correlation phases of the investigation, network forensics also need to adapt advanced analytics and intelligence based techniques. Considering the contributions offered by the AI techniques in various other fields of computing and the proposals made about AI, it gives a scope for the application of AI techniques into the field of network forensics.

NETWORK-LEVEL ANALYSIS OF USER APPLICATION DATA

An essential aim for a forensic investigator is to determine how a crime was undertaken – in order to identify the actors involved, how the crime unfolded and to develop a chronology of the incident. As argued by the literature review, the ability to do this based upon network traffic is extremely challenging and very time consuming. At present, only information about what services were accessed, timestamps and volumes of traffic can be determined – in addition to some object-based data carving. It is suggested that providing the investigator with a wider range of information about the nature of the service access (as illustrated in Table 1), could provide valuable insights into the nature and execution of the crime. This section presents the results of an experiment to determine the viability of extracting user-interactions from network meta-data. The experiment was based upon three key services:

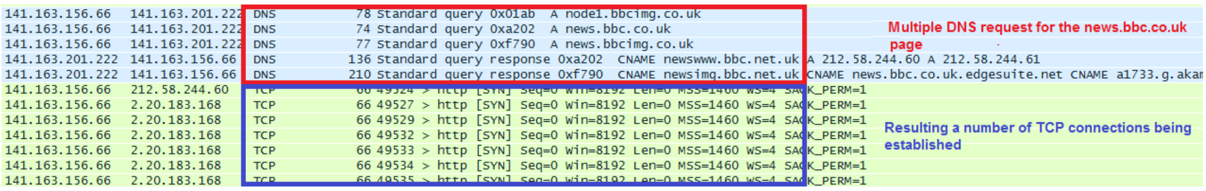
1. Web Surfing – BBC News website
2. Voice/IM Chats – Skype
3. Social Networking – Facebook

Table 1: Examples of selected Internet-enabled services and user actions

Services	**User Actions**
Cloud Storage	Edit, share, upload, download, remove
Emails	Compose, attachment, read, delete
Information gathering	Information browsing, watch video clips, listen to online clips
Instant messenger	Chat, voice call, video conference, file transfer
Online banking	View bank statement, money transfer
Social networking	Posting, viewing wall, uploading photos/videos, chatting, comment

Web Surfing

To analyse what user interactions are possible with general web surfing, it was decided to focus upon the BBC website as its representative of an information portal (and a particularly popular one). It is also an example of a non-encrypted site (in comparison to the other two services which are being analysed). As the service is unencrypted it is possible to analyse the traffic at TCP flow level rather than packet level. Being a dynamic (rather than simple website) each request made to the site generated a number of TCP connections but it is possible to identify specific user interactions. Figure 1 illustrates a user loading a page.



```
141.163.156.66   141.163.201.222 DNS        78 Standard query 0x01ab  A node1.bbcimg.co.uk
141.163.156.66   141.163.201.222 DNS        74 Standard query 0xa202  A news.bbc.co.uk
141.163.156.66   141.163.201.222 DNS        77 Standard query 0xf790  A news.bbcimg.co.uk
141.163.201.222  141.163.156.66  DNS       136 Standard query response 0xa202  CNAME newswww.bbc.net.uk A 212.58.244.60 A 212.58.244.61
141.163.201.222  141.163.156.66  DNS       210 Standard query response 0xf790  CNAME newsimg.bbc.net.uk CNAME news.bbc.co.uk.edgesuite.net CNAME a1733.g.akam
141.163.156.66   212.58.244.60   TCP        66 49524 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49527 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49529 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49532 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49533 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49534 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
141.163.156.66   2.20.183.168    TCP        66 49535 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
```

Multiple DNS request for the news.bbc.co.uk page

Resulting a number of TCP connections being established

Figure 1: Surfing BBC News

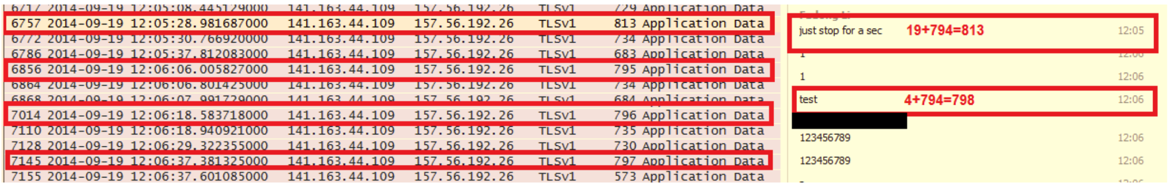
It is possible to derive a number of user interaction signals from web surfing BBC news that provides more information than simply that a suspect/victim accessed the site (as illustrated in Table 2). It is possible to understand how many pages they accessed – whether this was merely a home page or a new story – how long they spent on each page and on the site overall. It is also possible to identify whether they watched any video content.

Table 2: User Interactions derived from BBC News

Actions	**Protocol**	**Destination Port**	**Total length (bytes)**	**Number of packets**	**Directions**
View page	TCP	Random port	Various	Various	server >Client
View video	TCP	Random port	MTU (Almost)	Many	server >Client

Voice/IM Chats

Skype was chosen for the voice and IM chatting applications as it is amongst the most popular. Skype based traffic is encrypted but an analysis of the network traces – at packet rather than flow level does reveal a number of user interactions. A number of user activities were tested against their corresponding network metadata signals, including chatting, video conferencing and file sharing. The analysis shows that chatting is handled directly from the client to the Skype server (IP 157.56.192.26 via TCP port 443) – as illustrated in Figure 2. The baseline for sending characters was 794 bytes on the network. For example, a 19-character sentence the corresponding network signal was one 813-byte frame. Upon repeating this experiment, it was noticed that the baseline did vary between users and thus a threshold will need to be identified on a per-user basis in order to identify this interaction.



```
6717 2014-09-19 12:05:08.445129000  141.163.44.109  157.56.192.26  TLSv1   729 Application Data
6757 2014-09-19 12:05:28.981687000  141.163.44.109  157.56.192.26  TLSv1   813 Application Data
6772 2014-09-19 12:05:30.766920000  141.163.44.109  157.56.192.26  TLSv1   734 Application Data
6786 2014-09-19 12:05:37.812083000  141.163.44.109  157.56.192.26  TLSv1   683 Application Data
6856 2014-09-19 12:06:06.005827000  141.163.44.109  157.56.192.26  TLSv1   795 Application Data
6864 2014-09-19 12:06:06.801425000  141.163.44.109  157.56.192.26  TLSv1   734 Application Data
6868 2014-09-19 12:06:07.991729000  141.163.44.109  157.56.192.26  TLSv1   684 Application Data
7014 2014-09-19 12:06:18.583718000  141.163.44.109  157.56.192.26  TLSv1   796 Application Data
7110 2014-09-19 12:06:18.940921000  141.163.44.109  157.56.192.26  TLSv1   735 Application Data
7128 2014-09-19 12:06:29.322355000  141.163.44.109  157.56.192.26  TLSv1   730 Application Data
7145 2014-09-19 12:06:37.381325000  141.163.44.109  157.56.192.26  TLSv1   797 Application Data
7155 2014-09-19 12:06:37.601085000  141.163.44.109  157.56.192.26  TLSv1   573 Application Data
```

just stop for a sec 19+794=813 12:05

1 12:06

1 12:06

test 4+794=798 12:06

[redacted]

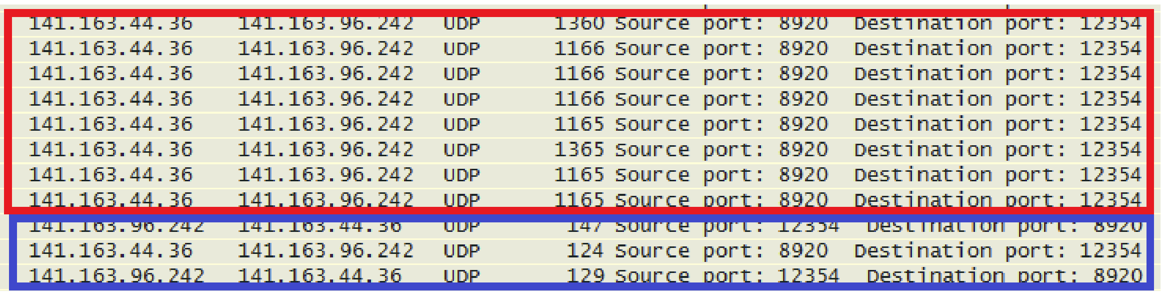
123456789 12:06

123456789 12:06

Figure 2: IM via Skype

Both the video-conference and file sharing were set up directly between two clients via UDP ports (i.e. not via the Skype server). For the video conference, one client was sending video frames with larger size packets (e.g.

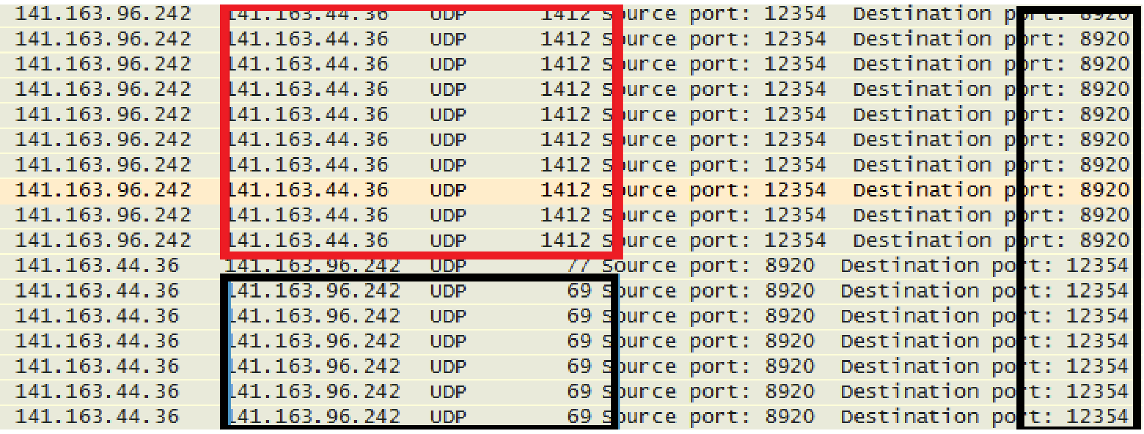
1166-1360 bytes) while the other was sending audio in smaller packets (e.g. 129-149 bytes) as only the former client had the camera turned on while the latter did not (as illustrated in Figure 3).



```
141.163.44.36    141.163.96.242   UDP    1360 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1166 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1166 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1166 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1165 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1365 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1165 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP    1165 Source port: 8920   Destination port: 12354
141.163.96.242   141.163.44.36    UDP     147 Source port: 12354   Destination port: 8920
141.163.44.36    141.163.96.242   UDP     124 Source port: 8920   Destination port: 12354
141.163.96.242   141.163.44.36    UDP     129 Source port: 12354   Destination port: 8920
```

Figure 3: Video conferencing via Skype

Regarding file sharing, an analysis of the subsequent network traces identify (as illustrated in Figure 4) that packets sizes reflexed the maximum packet size that the network can handle (the MTU) (i.e. 1412 bytes in this example) and these were sent by the sender while receiving little traffic from the receiver (merely replying with acknowledgments) (i.e. 69 bytes Ethernet frames). Furthermore, both traffic flows for video-conferencing and file sharing was sent in a continuous manner - thus would manifest themselves as a series of packets lasting more than a few seconds.



```
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.96.242   141.163.44.36    UDP    1412 Source port: 12354   Destination port: 8920
141.163.44.36    141.163.96.242   UDP      77 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
141.163.44.36    141.163.96.242   UDP      69 Source port: 8920   Destination port: 12354
```

Figure 4: File Sharing via Skype

Based up these observations it is possible to derived signatures for the network traffic that would identify the user interactions within Skype. As illustrated in Table 3, even though the service is completely encrypted it is still possible to identify a range of interactions – providing information to an investigator about how many times the suspect/victim uses skype for instant messaging, video/audio calling and file uploads.

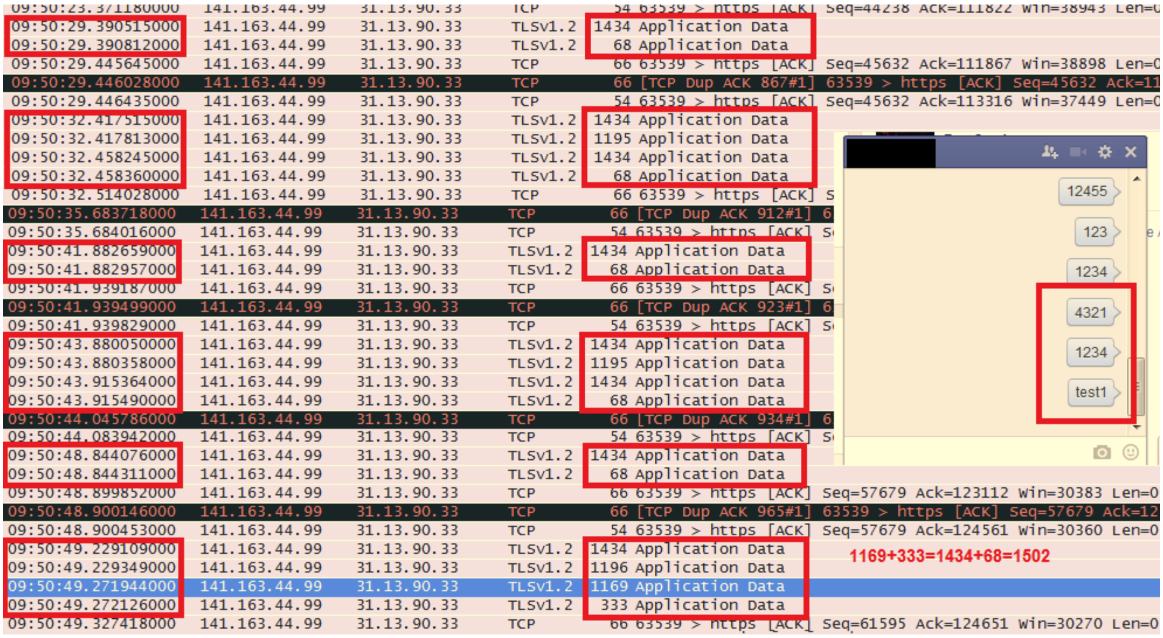
Table 3: User Interactions Derived from Skype

Actions	**Protocol**	**Destination Port**	**Frame length (bytes)**	**Number of packets**	**Directions**
Chat	TCP	443	794+	1	Client >server
File sharing	UDP	Random port	MTU (Almost)	Many	Sending client > receiving client
File sharing	UDP	Random port	69	Many	Receiving client > Sending client
Video conference	UDP	Random port	1165-1365	Many	Both clients
Audio call	UDP	Random port	129-147	Many	Both clients
Idle	TCP	443	572	1	Client >server
Click on contacts	TCP	443	731	1	Client >server

Social Networking

Facebook was chosen for the social network analysis. A number of user activities were performed and compared against their corresponding network signals. The network signals that are created when accessing the site are large in number and also encrypted which results in a more complex analysis. With more dynamic and complex web services, it is clear a significant degree of additional traffic is generated – which serves to complicate the analysis. For example, Facebook has an idle signal that is sent periodically – this is functionality that helps to ensure the webpage is up to date. Without care, this signal could be misrepresented as a user interaction.

When the typing activity started in the chatting dialog box, a network signal with a total of 1502 bytes (i.e. 1434+68) was sent from client to the Facebook server; the same pattern also occurred directly after a message being sent. Moreover, the baseline for chatting in Facebook is a total of 2,625 bytes (i.e. 1434+1191). For example, when a 4-character word is sent to the server, a total of 2,629 bytes appeared on the network. This pattern is illustrated in the analysis presented in Figure 5.



```
09:50:23.371180000  141.163.44.99  31.13.90.33  TCP      54 63539 > https [ACK] Seq=44238 Ack=111822 Win=38943 Len=0
09:50:29.390515000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:29.390812000  141.163.44.99  31.13.90.33  TLSv1.2    68 Application Data
09:50:29.445645000  141.163.44.99  31.13.90.33  TCP        66 63539 > https [ACK] Seq=45632 Ack=111867 Win=38898 Len=0
09:50:29.446028000  141.163.44.99  31.13.90.33  TCP        66 [TCP Dup ACK 867#1] 63539 > https [ACK] Seq=45632 Ack=11
09:50:29.446435000  141.163.44.99  31.13.90.33  TCP        54 63539 > https [ACK] Seq=45632 Ack=113316 Win=37449 Len=0
09:50:32.417515000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:32.417813000  141.163.44.99  31.13.90.33  TLSv1.2  1195 Application Data
09:50:32.458245000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:32.458360000  141.163.44.99  31.13.90.33  TLSv1.2    68 Application Data
09:50:32.514028000  141.163.44.99  31.13.90.33  TCP        66 63539 > https [ACK] S
09:50:35.683718000  141.163.44.99  31.13.90.33  TCP        66 [TCP Dup ACK 912#1] 6
09:50:35.684016000  141.163.44.99  31.13.90.33  TCP        54 63539 > https [ACK] S
09:50:41.882659000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:41.882957000  141.163.44.99  31.13.90.33  TLSv1.2    68 Application Data
09:50:41.939187000  141.163.44.99  31.13.90.33  TCP        66 63539 > https [ACK] S
09:50:41.939499000  141.163.44.99  31.13.90.33  TCP        66 [TCP Dup ACK 923#1] 6
09:50:41.939829000  141.163.44.99  31.13.90.33  TCP        54 63539 > https [ACK] S
09:50:43.880050000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:43.880358000  141.163.44.99  31.13.90.33  TLSv1.2  1195 Application Data
09:50:43.915364000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:43.915490000  141.163.44.99  31.13.90.33  TLSv1.2    68 Application Data
09:50:44.045786000  141.163.44.99  31.13.90.33  TCP        66 [TCP Dup ACK 934#1] 6
09:50:44.083942000  141.163.44.99  31.13.90.33  TCP        54 63539 > https [ACK] S
09:50:48.844076000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:48.844311000  141.163.44.99  31.13.90.33  TLSv1.2    68 Application Data
09:50:48.899852000  141.163.44.99  31.13.90.33  TCP        66 63539 > https [ACK] Seq=57679 Ack=123112 Win=30383 Len=0
09:50:48.900146000  141.163.44.99  31.13.90.33  TCP        66 [TCP Dup ACK 965#1] 63539 > https [ACK] Seq=57679 Ack=12
09:50:48.900453000  141.163.44.99  31.13.90.33  TCP        54 63539 > https [ACK] Seq=57679 Ack=124561 Win=30360 Len=0
09:50:49.229109000  141.163.44.99  31.13.90.33  TLSv1.2  1434 Application Data
09:50:49.229349000  141.163.44.99  31.13.90.33  TLSv1.2  1196 Application Data
09:50:49.271944000  141.163.44.99  31.13.90.33  TLSv1.2  1169 Application Data
09:50:49.272126000  141.163.44.99  31.13.90.33  TLSv1.2   333 Application Data
09:50:49.327418000  141.163.44.99  31.13.90.33  TCP        66 63539 > https [ACK] Seq=61595 Ack=124651 Win=30270 Len=0
```

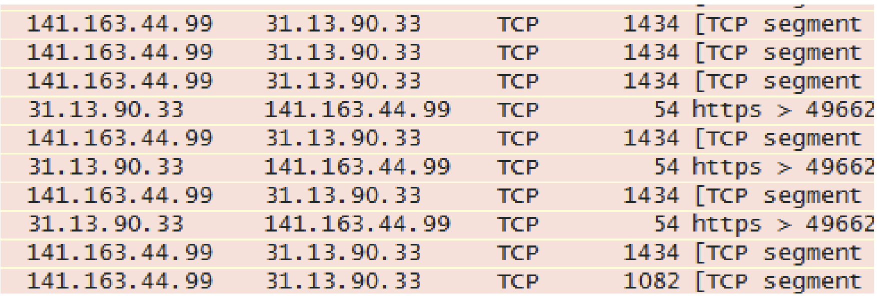
[redacted]

12455
123
1234
4321
1234
test1

1169+333=1434+68=1502

Figure 5: Chatting on Facebook

An examination of the photo uploading activity (as illustrated in Figure 6), shows a stream of almost full size Ethernet frames (i.e. the MTU which in this example 1434 bytes) were sent from the client to the Facebook server while the Facebook server simply acknowledged the receiving of the data (indicated by TCP ACK flag).



```
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
31.13.90.33     141.163.44.99   TCP       54 https > 49662
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
31.13.90.33     141.163.44.99   TCP       54 https > 49662
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
31.13.90.33     141.163.44.99   TCP       54 https > 49662
141.163.44.99   31.13.90.33     TCP     1434 [TCP segment
141.163.44.99   31.13.90.33     TCP     1082 [TCP segment
```

Figure 6: Photo uploading on Facebook

Based upon this signals analysis, the Table 4 presents a series of user interactions and the subsequent information required to detect the interaction from network traffic.

Table 4: User Interactions Derived from Facebook

Actions	Protocol	Destination Port	total length (bytes)	Number of packets	Directions
Chat	TCP	443	2,625+	2	Client >server
Typing	TCP	443	1502	2	Client >server
File uploading	TCP	443	MTU (Almost)	Many	Client >server
Idle	TCP	443	149	2	Client >server
Page load / viewing wall	TCP	Random port	Various	Various	Server >client

UO-NFAT FRAMEWORK

A network forensic analysis tool needs to develop the ability to operate in a similar fashion to computer forensic case management tools – i.e. provide a basis for abstracting/parsing low-level data into higher-level information that aids in easy analysis and interpretation by a forensic examiner. This includes the ability to identify and reconstruct protocols, reduce non relevant traffic, identify and extract objects (e.g. images, documents), reconstruct user events and correlate activities and data. The purpose of having such a NFAT is to reduce the cost, the time spent on the investigation and the cognitive load placed on the investigator in terms of their ability to correlate evidence.

The User-Oriented (UO) NFAT illustrated in Figure 7 seeks to provide a robust, extensible and investigator-friendly tool that provides the necessary core analysis functionality and the additional case management tasks such as reporting. Its focus is placed upon cyber and computer-assisted crimes that involve people – i.e. insider misuse, hacking and IP theft – rather than on the identification and analysis of malware. The central component of the framework, *UO-NFAT Manager*, acts as the case manager and central point from which analysis of the traffic can take place - interconnecting different modules and analysis tools. It is the UO-NFAT Manager that the investigator interacts with in order to provide initial case intelligence (e.g. nature of the crime, suspect/victim information) upon which analysis can subsequently focus to provide more detailed information. For example, this can help in phases such as pre-processing, protocol analysis, pattern/keyword searching and data carving. Having a specific set of case intelligence can contribute towards reducing the volume of data to be analysed.

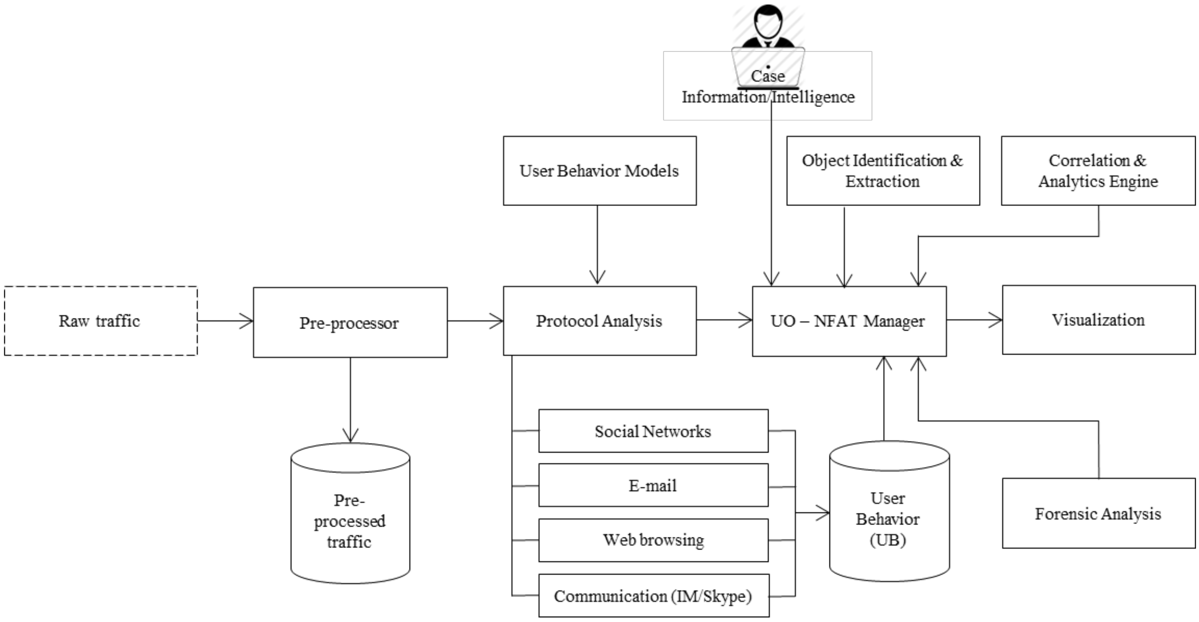


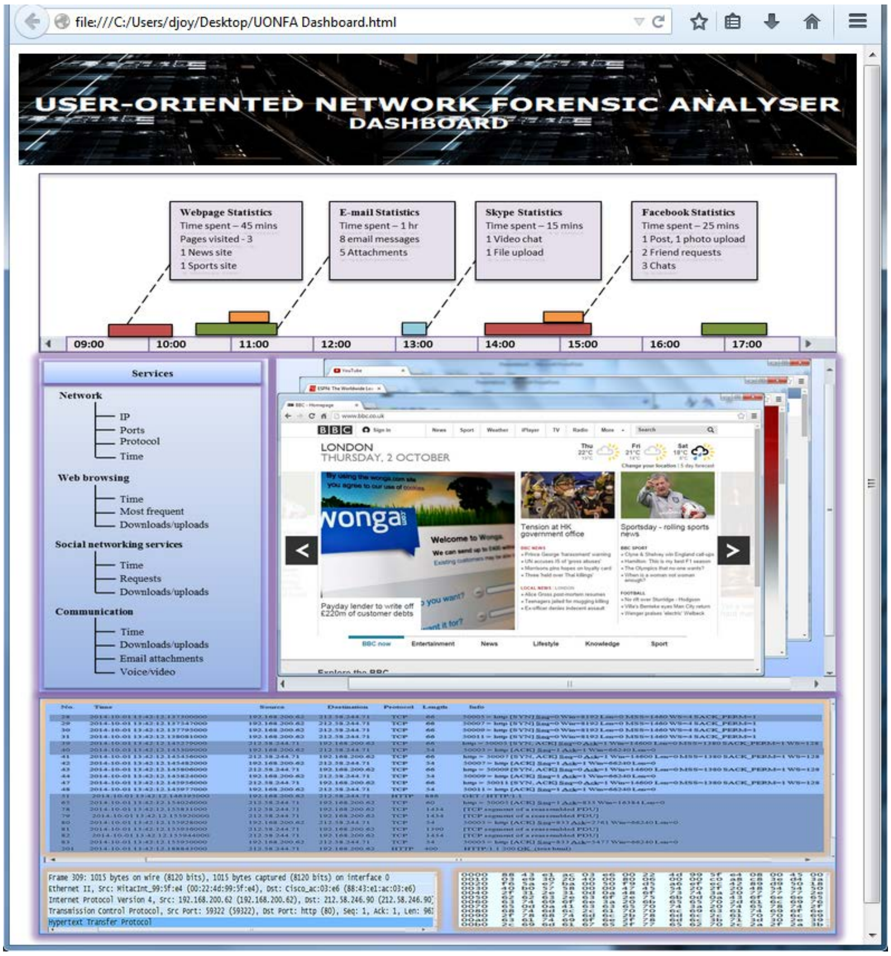
Figure 7: User-Oriented Network Forensic Analyser Design

The *Pre-Processor* module takes the raw traffic as its input and the purpose of this module is to reduce the amount of network data to be analysed by eliminating any noise. Considering the nature of the investigation, particular subsets of network protocols can be removed from the analysis – for example, when investigating what user activity is present, all protocols involved in machine-to-machine communication such as DHCP and ICMP can be removed. The pre-processor will also abstract the data out to TCP flows from the packet level to aid in reducing the complexity, noise and volume of data to be analysed. Once completed, the pre-processed traffic is stored in a database and is forwarded to the protocol analysis module.

The *Protocol Analysis* segment performs the detailed examination of the pre-processed traffic to generate the high-level user directed behavioural information in order to assist an investigator in appreciating what the user(s) are doing within online services. All abstracted user interactions are subsequently stored in the *User Behaviour* database. The *Object Identification & Extraction* module which operates on the information stored in the 'user behaviour' database and pre-processed data to identify and extract any high level objects (such as images, documents, PDFs, emails, instant message chats) and correlate these with the derived user interactions. The outcome of this process is to provide an investigator with an overview of the activities undertaken by users and direct access to data being transferred.

The *Forensic Analysis* module attempts to incorporate some features exhibited by the established computer forensic tools such as hashing (for identifying file signatures, known and notable files), data carving, regular expressions and keyword searching. The *Correlation & Analytics* engine uses data correlation and analytic techniques in order to provide further investigation and analysis between user activities, data objects and data flows. For example, the relationship between different IP addresses those are responsible for the suspicious traffic and movement of data.

Based on the all the obtained information, the *Visualization* module presents the investigated user activities in a visual and an interactive manner. An example of the user interface dashboard is shown in Figure 8. The tool provides an interface where investigators get access to the user interactions and summary information in a chronological manner (top window). Selection of any of the interactions, results in the data objects being loaded and thus viewable by the forensic examiner (middle right window). For integrity and bookmarking purposes, when the investigator selects an object, the associated network traffic, which this data is derived from, is then highlighted (in the bottom window) – this ensures all evidence is directly mapped back to the raw evidence. The interface also provides the investigator with a range of filtering options based upon protocols, services and activities (middle left window).



Webpage Statistics
Time spent – 45 mins
Pages visited - 3
1 News site
1 Sports site

E-mail Statistics
Time spent – 1 hr
8 email messages
5 Attachments

Skype Statistics
Time spent – 15 mins
1 Video chat
1 File upload

Facebook Statistics
Time spent – 25 mins
1 Post, 1 photo upload
2 Friend requests
3 Chats

Services

Network
- IP
- Ports
- Protocol
- Time

Web browsing
- Time
- Most frequent
- Downloads/uploads

Social networking services
- Time
- Requests
- Downloads/uploads

Communication
- Time
- Downloads/uploads
- Email attachments
- Voice/video

```
No.  Time                           Source          Destination     Protocol Length Info
28   2014-10-01 13:42:12.137300000  192.168.200.62  212.58.244.71   TCP      66     50005 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
29   2014-10-01 13:42:12.137347000  192.168.200.62  212.58.244.71   TCP      66     50007 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
30   2014-10-01 13:42:12.137793000  192.168.200.62  212.58.244.71   TCP      66     50009 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
31   2014-10-01 13:42:12.138081000  192.168.200.62  212.58.244.71   TCP      66     50011 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
39   2014-10-01 13:42:12.145279000  212.58.244.71   192.168.200.62  TCP      66     http > 50005 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM=1 WS=128
40   2014-10-01 13:42:12.145300000  192.168.200.62  212.58.244.71   TCP      54     50005 > http [ACK] Seq=1 Ack=1 Win=66240 Len=0
41   2014-10-01 13:42:12.145436000  212.58.244.71   192.168.200.62  TCP      66     http > 50007 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM=1 WS=128
42   2014-10-01 13:42:12.145482000  192.168.200.62  212.58.244.71   TCP      54     50007 > http [ACK] Seq=1 Ack=1 Win=66240 Len=0
43   2014-10-01 13:42:12.145806000  212.58.244.71   192.168.200.62  TCP      66     http > 50009 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM=1 WS=128
44   2014-10-01 13:42:12.145824000  192.168.200.62  212.58.244.71   TCP      54     50009 > http [ACK] Seq=1 Ack=1 Win=66240 Len=0
47   2014-10-01 13:42:12.145936000  212.58.244.71   192.168.200.62  TCP      66     http > 50011 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM=1 WS=128
48   2014-10-01 13:42:12.145977000  192.168.200.62  212.58.244.71   TCP      54     50011 > http [ACK] Seq=1 Ack=1 Win=66240 Len=0
51   2014-10-01 13:42:12.148393000  192.168.200.62  212.58.244.71   HTTP     688    GET / HTTP/1.1
67   2014-10-01 13:42:12.154026000  212.58.244.71   192.168.200.62  TCP      60     http > 50005 [ACK] Seq=1 Ack=835 Win=16384 Len=0
78   2014-10-01 13:42:12.155831000  212.58.244.71   192.168.200.62  TCP      1434   [TCP segment of a reassembled PDU]
79   2014-10-01 13:42:12.155920000  212.58.244.71   192.168.200.62  TCP      1434   [TCP segment of a reassembled PDU]
80   2014-10-01 13:42:12.155928000  192.168.200.62  212.58.244.71   TCP      54     50005 > http [ACK] Seq=835 Ack=2761 Win=66240 Len=0
81   2014-10-01 13:42:12.155936000  212.58.244.71   192.168.200.62  TCP      1390   [TCP segment of a reassembled PDU]
82   2014-10-01 13:42:12.155944000  212.58.244.71   192.168.200.62  TCP      1434   [TCP segment of a reassembled PDU]
83   2014-10-01 13:42:12.155950000  192.168.200.62  212.58.244.71   TCP      54     50005 > http [ACK] Seq=835 Ack=5477 Win=66240 Len=0
301  2014-10-01 13:42:12.188843000  212.58.244.71   192.168.200.62  HTTP     400    HTTP/1.1 200 OK (text/html)
```

```
Frame 309: 1015 bytes on wire (8120 bits), 1015 bytes captured (8120 bits) on interface 0
Ethernet II, Src: MitacInt_99:5f:e4 (00:22:4d:99:5f:e4), Dst: Cisco_ac:03:e6 (88:43:e1:ac:03:e6)
Internet Protocol Version 4, Src: 192.168.200.62 (192.168.200.62), Dst: 212.58.246.90 (212.58.246.90)
Transmission Control Protocol, Src Port: 59322 (59322), Dst Port: http (80), Seq: 1, Ack: 1, Len: 96
Hypertext Transfer Protocol
```

Figure 8: User-Oriented Network Forensic Analyser Dashboard

CONCLUSION AND FUTURE WORK

The paper addresses the need for network forensics, giving specific importance to the requirement of generating high-level information from the raw network traffic and applying correlation techniques to better appreciate the relationship between artefacts. The paper has demonstrated that it is possible to derive user interactions from online services from low-level network metadata and has subsequently proposed a tool then proceeds to analyse and visualise these in order to reduce the cognitive load upon the investigator and make it easier to identify and map an incident.

Our future work seeks to develop an empirical basis for extracting user interactions from a complete range of application services, and to develop the UO-NFAT tool that will incorporate this into a functional and usable forensic tool. Given the processing and storage-heavy nature of network forensics, more research will be conducted to investigate how to handle and process large volumes of traffic and look to identify relevant technology platforms that this can be deployed to.

REFRENCES

AccessData Corp. ‘Forensic Toolkit [Online] http://accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk [November, 2014]

AccessData Corp. ‘SilentRunner Sentinel’ [Online] http://accessdata.com/solutions/cybersecurity/silentrunner-sentinel [July, 2013]

Cisco “Cisco Visual Networking Index: Forecast and Methodology, 2013-2018”, [online], http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html [November 2014]

Cohen, M. I. (2008) ‘PyFlag – An advanced network forensic framework’. *Digital Investigation: The International Journal of Digital Forensics & Incident Response,* **5**, 112-120

DFRWS ‘A Road Map for Digital Forensic Research: Report from the first Digital Forensic Research Workshop (DFRWS) – August, 2001’ [Online] http://www.dfrws.org/2001/dfrws-rm-final.pdf [November, 2014]

Garfinkel, S. (2007) ‘Anti-Forensics: Technique, Detection and Countermeasures’. *2nd International Conference on i-Warfare and Security*, 77-84

Garfinkel, S L. (2010) ‘Digital Forensics Research: The Next 10 Years’. *Digital Investigation*, 7, 64-73

Internet Crime Complaint Center (IC3) ‘Internet crime report’ [Online] http://www.ic3.gov/media/annualreport/2012_IC3Report.pdf [July, 2013]

ISO ‘ISO/IEC 27037:2012 – Information Technology – Security Techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence’ [Online] https://www.iso.org/obp/ui/#iso:std:iso-iec:27037:ed-1:v1:en [November, 2014]

Lyonesse Software ‘Iris Network Traffic Analyser’ [Online] http://www.lyonware.co.uk/Iris.htm [February, 2013]

Merkle, L.D. (2008) ‘Automated Network Forensics’. *Proceedings of the conference on genetic and evolutionary computation (GECCO 2008)*, 1929-1932

Mukkamala, S. and Hung, A.W. (2003) ‘Identifying Significant Features for Network Forensic Analysis using Artificial Intelligence Techniques’. *International Journal of Digital Evidence,* **1**(4), 1-17

NETRESEC ‘NetworkMiner’ [Online] http://www.netresec.com/?page=NetworkMiner [July, 2013]

Netscout ‘nGenius InfiniStream Appliance’ [Online] http://www.netscout.com/library/Data%20sheets/NetScout_DS_nGenius_InfiniStream_Appliance_SP.pdf [July, 2013]

NIKSUN ‘NetDetector Alpine’ [Online] https://www.niksun.com/product.php?id=4 [September, 2013]

NIKSUN ‘NetIntercept’ [Online] https://www.niksun.com/collateral/NIKSUNDatasheet_NetIntercept_1010.pdf [September, 2013]

Pajek, P. and Pimenidis, E. (2009) ‘Computer Anti-Forensics Methods and Their Impact on Computer Forensic Investigation’. *Global Secuirty, Safety and Sustainability Communications in Computer and Information Science*, 45, 145-155

Pilli, E S., Joshi, R C. and Niyogi, R. (2010) ‘Network Forensic Frameworks: Survey and research challenges’. *Digital Investigation,* **7**, 14-27

RSA ‘RSA NetWitness Overview’ [Online] http://uk.emc.com/collateral/data-sheet/rsa-netwitness-nextgen.pdf [September, 2013]

RSA ‘The Current State of Cybercrime 2014 – An inside look at the changing threat landscape’ [Online] http://www.emc.com/collateral/white-paper/rsa-cyber-crime-report-0414.pdf [August 2014]

SC Magazine 'NetWitness NextGen v9.5' [Online] http://www.scmagazine.com/netwitness-nextgen-v95/review/3469/ [September, 2013]

SC Magazine 'RSA NetWitness' [Online] http://www.scmagazine.com/rsa-netwitness/review/3650/ [September, 2013]

Solera Networks 'DeepSee 5150 – Comprehensive Network Forensics Appliance' [Online] http://www.soleranetworks.co.jp/resources/datasheet5150_web.pdf [February, 2014]

Stoffel, K., Cotofrei, P. and Han, D. (2010) 'Fuzzy Methods for Forensic Data Analysis'. *2010 International Conference of Soft Computing and Pattern Recognition (SoCPar),*23-28

Tamos 'NetResident – Network Content Monitoring' [Online] http://www.tamos.com/docs/nr_datasheet.pdf [February, 2014]

United Nations Crime and Justice Information Network 'International review of criminal policy - United Nations Manual on the prevention and control of computer-related crime' [Online] http://www.uncjin.org/Documents/EighthCongress.html [December 2012]

WildPackets 'OmniPeek Network Analyzer' [Online] http://www.wildpackets.com/elements/omnipeek/OmniPeek_Network_Analyzer_datasheet.pdf [February, 2014]

Wireshark 'Wireshark User's Guide' [Online] https://www.wireshark.org/docs/wsug_html_chunked/ [September 2013]

Xplico 'Xplico Features' [Online] http://www.xplico.org/about [September, 2013]