2017

A novel privacy preserving user identification approach for network traffic

Nathan Clarke
Edith Cowan University, n.clarke@ecu.edu.au

Fudong Li

Steven Furnell
Edith Cowan University, s.furnell@ecu.edu.au

10.1016/j.cose.2017.06.012
Clarke, N., Li, F., & Furnell, S. (2017). A novel privacy preserving user identification approach for network traffic. computers & security, 70, 335-350. Available here

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A novel privacy preserving user identification approach for network traffic

N. Clarke [a,b], F. Li [a,*], S. Furnell [a,b,c]

[a] Centre for Security, Communications and Network Research, University of Plymouth, Plymouth, United Kingdom
[b] Security Research Institute, Edith Cowan University, WA, Australia
[c] Centre for Research in Information and Cyber Security, Nelson Mandela Metropolitan University, Port Elizabeth, South Africa

ARTICLE INFO

Article history:
Received 23 January 2017
Received in revised form 30 May 2017
Accepted 26 June 2017
Available online 10 July 2017

Keywords:
Biometrics
Digital forensics
Network forensics
Network metadata
Traffic analysis
User identification

ABSTRACT

The prevalence of the Internet and cloud-based applications, alongside the technological evolution of smartphones, tablets and smartwatches, has resulted in users relying upon network connectivity more than ever before. This results in an increasingly voluminous footprint with respect to the network traffic that is created as a consequence. For network forensic examiners, this traffic represents a vital source of independent evidence in an environment where anti-forensics is increasingly challenging the validity of computer-based forensics. Performing network forensics today largely focuses upon an analysis based upon the Internet Protocol (IP) address – as this is the only characteristic available. More typically, however, investigators are not actually interested in the IP address but rather the associated user (whose account might have been compromised). However, given the range of devices (e.g., laptop, mobile, and tablet) that a user might be using and the widespread use of DHCP, IP is not a reliable and consistent means of understanding the traffic from a user. This paper presents a novel approach to the identification of users from network traffic using only the meta-data of the traffic (i.e. rather than payload) and the creation of application-level user interactions, which are proven to provide a far richer discriminatory feature set to enable more reliable identity verification. A study involving data collected from 46 users over a two-month period generated over 112 GBs of meta-data traffic was undertaken to examine the novel user-interaction based feature extraction algorithm. On an individual application basis, the approach can achieve recognition rates of 90%, with some users experiencing recognition performance of 100%. The consequence of this recognition is an enormous reduction in the volume of traffic an investigator has to analyse, allowing them to focus upon a particular suspect or enabling them to disregard traffic and focus upon what is left.

1. Introduction

During the past 15 years, Internet usage has experienced explosive growth and technological evolution – from a simple data network with around 500 million users to a multipurpose and multiservice platform with almost 3.2 billion users (Internetlivestats, 2015). Indeed, with the prevalence of various broadband network technologies, mobile devices, and the web, users can utilize a wide range of services to complete

[*] Corresponding author.
E-mail address: fudong.li@plymouth.ac.uk (F. Li).

http://dx.doi.org/10.1016/j.cose.2017.06.012

various personal and business tasks both in the office and on the move 24/7. Services include (but are not limited to) entertainment (e.g., watching online videos), communication (e.g., making VoIP calls), finance (e.g., online banking), data storage (e.g., cloud services), office applications (e.g., Google Docs), and even Operating Systems (e.g., ZeroPC). It is evidenced that these activities generate a tremendous amount of IP traffic – 60 Exabyte's globally per month in 2014, 40% of which originated from non-PC devices (Cisco, 2015).

While people and business take the full advantage of the Internet, malicious attackers use the same infrastructure to plot various cyberattacks (e.g., hacking, Denial of Service (DoS), insider misuse, and phishing) against user's information, corporation's servers, and Internet services. Indeed, there is an ever-increasing volume of literature reporting the scale and nature of the computer-related crime (both cyber and computer-assisted). For example, the FBI Internet Crime Complaint Center reported 269,422 self-reported incidents of cyberattacks (mainly fraud related) in 2014, with a total estimated loss of $800 million (FBI, 2015). The Verizon's 2015 Data Breach Investigations Report shows that almost 80,000 security incidents were discovered by 70 organizations around world in 2014, causing them an estimated financial loss of $400 million (Verizon, 2015).

With the aim to search evidence and testify against cybercriminals, images of suspects' digital devices are forensically created and examined by investigators. For instance, statistics from the FBI's Regional Computer Forensics Laboratory show that they had completed 6322 examinations and investigators testified in court and/or at hearing 88 times in 2014 (FBI, 2014). Due to the volatile nature of computer memory and the availability of anti-forensic techniques (e.g., data wiping), critical evidence can vanish when the computer is switched off or purposely destroyed by the suspects as they have direct access to their digital devices. As a result, a complete picture of how an attack is conducted might not be achievable. As such, independent sources of data, such as network traffic, provide investigators with an invaluable source of evidence, where the chance of the evidence being tampered or destroyed is minimized.

Many tools (both commercial and open source) have been designed and developed to assist network forensic examiners to conduct investigations. These tools include NIKSUN's NetDetector Suite (NIKSUN, 2016), RSA's Netwitness Suite (RSA, 2016), Wireshark (Wireshark, 2016), PyFlag (Cohen, 2008), and Xplico's Open Source Network Forensic Analysis Tool (Xplico, 2016). They all rely upon the IP address of the suspect's machine as a basis for the investigation, assuming IP is static and linkable to an individual. However, IPs are increasingly unreliable due to the mobile nature of devices and the use of dynamic allocation of IP addresses. As a result, beyond the detected attack (often a single IP packet or flow), it is a challenge to investigate the question of *what* has happened in terms of the wider attack and *who* was actually involved. Indeed, to identify and extract a specific user's traffic from the wider organization over a prolonged period is a particularly challenging task.

Biometrics is a proven method that identifies individuals based upon their physiological or behavioural traits. Several biometric techniques, such as face recognition, fingerprint identification, and speaker recognition, are already in wide use within the forensic domain (Vezzani et al., 2013). However, little research has been undertaken on identifying individuals using biometric design techniques within the network forensic domain. Existing research has largely focused upon merely providing network data (either in packet or flow forms) to identify anomalous behaviour (a two-class problem of benign and malicious traffic) rather than looking to identify particular individuals (which is an *n* class problem, where *n* is the size of the user population). Studies into behavioural profiling on desktop and mobile platforms have demonstrated the ability to verify an individual; however, deriving application-level interactions (such as which websites users visit and more importantly what they do whilst visiting – posting, chatting, listening to music or watching video) from low-level encrypted packet-based data has proven challenging. Furthermore, using these application-based interactions for identification rather than verification introduces a need for stronger discriminative information. To this end, this paper describes an experimental study which proposes and investigates user identification through user's application-level activities based solely on the metadata of network packets.

The remainder of the paper is structured as follows. Section 2 reviews existing network traffic analysis methods and the prior art in behavioural profiling. Section 3 presents a novel feature extraction approach to deriving user oriented application level activities. The research methodology and the formation of the user activity dataset are presented in Section 4, followed by a full experimental study to evaluate the approach in Section 5. Section 6 discusses the proposed approach and its impact, while the conclusions and future work are highlighted in Section 7.

2. Prior art in network and behavioural profiling

In order to fully understand the relationship between user's application level interactions and their corresponding network signals, a detailed review on existing network traffic analysis is discussed. The work into network traffic analysis can be traced back to 1990s (Claffy et al., 1995; Debar et al., 1999) and is utilized by network administrators in various domains, including management, prioritization, performance, accounting, application behaviour analysis, and security (Hofstede et al., 2014). Depending upon the granularity of the analysis, network traffic can be analysed by two approaches: packet based (finer grained) or flow based (coarser grained). The packet based method is mainly used to examine the content (i.e., the payload) of individual IP packets, while flow based approach is utilized to analyse the summary of multiple IP packets that share similar characteristics over a period of time (i.e., IP flows). Details of these two methods, including their working principles, existing research, and advantages and disadvantages, are described. In addition, whilst behavioural profiling has not been applied to network traffic, an analysis of the prior art is presented to provide a baseline understanding of the technique and the typical levels of performance that can be expected.

2.1. *Packet based network analysis method*

It is well-known that malicious attackers can use various data fields of the IP packet (e.g., header and payload) to conceal information and plot different attacks via the network, including malware, intrusions, and data exfiltration (Ahmed and Lhee, 2011; He et al., 2014; Wang and Stolfo, 2004). In order to detect and deter these attacks, the packet based method (also known as Deep Packet Inspection (DPI)) is utilized to examine the content (primarily the payload) of IP packets that passes a network monitoring point on the fly. The IP packet content (including IP addresses, port numbers, and payload information) can be directly obtained from the network through either hardware or software applications. The content is then instantly analysed and compared with existing rulesets by using pattern matching techniques (e.g., string matching or Self-Organizing Map) (Bolzoni et al., 2006; Zanero, 2005). Any matches would indicate the presence of an attack. Depending upon the application in which the packet based analysis method is used, a number of actions could be taken regarding the incident, such as blocking the traffic or raising an alarm. Under this principle, a number of studies were conducted to counter different threats; examples of these studies, including their applications, matching methods, and performance, are summarized in Table 1. Due to the time consuming nature of the pattern matching technique and the increasing amount of data that network devices need to process (e.g., Giga bits per second), a wide range of research have been carried out to enhance the performance and effectiveness of the packet based network analysis based applications, including packet sampling techniques (Jurga and Hulboj, 2007), hardware based solutions (Cho et al., 2002; Dharmapurikar et al., 2004; Smith et al., 2009; Sourdis et al., 2005), and novel algorithms (Dharmapurikar and Lockwood, 2006; Liu et al., 2004; Lu et al., 2006; Smallwood and Vance, 2011; Sun et al., 2011; Yu et al., 2006).

The packet based network analysis method can be effective against different network related attacks, such as malware distribution, data exfiltration, DoS, and network intrusions. Despite a wide range of methods being devised to improve the performance, the packet inspection approach is still a time consuming process due to the bit-by-bit comparison nature; indeed, to analyse gigabits of corporate network traffic within a fraction of a second is a challenging task. Also, the widespread use of SSL/TLS results in encrypted traffic requiring further efforts to access the payload. Even in cases where it is possible to obtain the payload in the plain text format, the decryption process itself will introduce additional delays and be a compromise on the confidentiality of the data. Moreover, it can be challenging and laborious to interpret the raw payload data into meaningful user related information (e.g., sending an email).

2.2. *Flow based network analysis approach*

The flow based analysis approach relies upon the concept of IP flows to detect various network attacks. A flow is the summary of a group of IP packets that share a set of common properties (e.g., source and destination IP addresses and port numbers) passing a network observation point during a certain timeframe (Claise, 2008). As a result, a typical flow record normally contains the following attributes: the time and date stamps (that indicate the start and end of the flow), the IP addresses of the source and the destination, their port numbers, the total size of the packet payload, the total number of IP packets, and the type of protocols (e.g., TCP or UDP). The network flow record is created by a flow generation application rather than obtained directly from the raw traffic. There are several well-known flow generation applications in the field, including Cisco's NetFlow, sFlow, Juniper's J-Flow, and IETF's IPFIX (Cisco, nd; Claise, 2008; Juniper, 2015; Sflow, 2015). Depending upon the implementation and configuration of each individual flow generation application, a flow record can be completed when certain criteria are met, such as when the flow is idle for a period time (e.g., 10 seconds), or the FIN or RST flags are seen in the TCP traffic. Within a flow based network monitoring system, the current traffic flow information is quickly compared with historical flow data by using various pattern classification techniques (e.g., neural networks or statistical models); any deviation (e.g., amount of data being sent or the type of traffic being observed on a particular port number) can be considered as a basis for alerting. According to this theory, many methods and tools have been proposed and devised within the flow based network analysis domain; a number of selected examples, along with their applications, pattern classification techniques and performance, are analysed in Table 2. With the aim to deal with high speed network traffic, a number of sampling techniques have been proposed for the flow based analysis approach, including Estan and Varghese (2003), Duffield et al. (2004), Duffield et al. (2005), Androulidakis et al. (2007) and Canini et al. (2009).

Due to its ability to handle large volume of network traffic (both encrypted and unencrypted) in a timely fashion, the flow based network analysis approach has become the primary method for security analysts to investigate different network

Table 1 – Examples of packet based network analysis works.

Studies	Applications	Performance
Mahoney and Chan (2001)	Anomaly based network IDS	65% detection rate
Wang and Stolfo (2004)	Anomaly based network IDS	100% detection rate with 0.1% false positive for port 80 traffic
Zanero (2005)	Anomaly based network IDS	66.7% detection rate with 0.03% false positive
Wang et al. (2005)	Zero-day worm detection	Over 95% of detection rate with less than 0.5% of false positive
Bolzoni et al. (2006)	Anomaly based network IDS	73.2% (detection rate) with less than 1% of false positive
Wang et al. (2006)	Buffer overflow attack blocker	100% detection rate on HTTP traffic
Ahmed and Lhee (2011)	Malware detection	False negative (4.69%) and false positive (2.53%)
Al-Bataineh and White (2012)	Detection of data exfiltration	99.97% detection rate on HTTP traffic
He et al. (2014)	Detection of encrypted data exfiltration	Close to 90% detection rate with less than 1% of false positives

Table 2 – Examples of flow based network analysis studies.

Studies	Applications	Performance
Kim et al. (2004)	Anomaly network detection	The method could detect scanning and flooding attacks
Pao and Wang (2004)	Signature based network IDS	The method was able to detect the Ping sweep, DoS and port scan attacks.
Crotti et al. (2006)	Traffic classification	99.4% and 97.5% detection rate for server's and client's HTTP traffic respectively
Song et al. (2006)	Anomaly based network IDS	94% detection rate with 0.2% false positive
Muraleedharan et al. (2010)	Anomaly based network IDS	100% detection rate on 15 attacks but with long delays
Braga et al. (2010)	Anomaly based network IDS	99.11% detection rate and 0.46% false alarm rate
Winter et al. (2011)	Anomaly based network IDS	98% detection rate
Tegeler et al. (2012)	Malware detection system	90% detection rate with 0.1% false positive rate
Jadidi et al. (2013)	Anomaly based network IDS	99.43% accuracy rate
Hofstede et al. (2013)	Anomaly based network IDS	95% detection rate with 1% false positive rate

related issues, including malware distributions, network intrusions, DDoS attacks, and traffic classifications. However, as it is a higher-level of abstraction and with the use of sampling techniques, the flow based network analysis approach losses the granularity of the traffic, such as the number of files being downloaded by an attacker during a given period, or the number of messages being sent by a user. Moreover, the performance of the flow based network analysis approach could be affected by the deployment of the DHCP, the usage of wireless technologies (e.g., Wi-Fi and 4G), the use of network proxies by users (e.g., the Onion Router network), and the utilization of IP spoofing technique.

As demonstrated above, the analysis of network traffic is a well-established domain and has been widely used to detect various network related attacks. Nonetheless, existing approaches experience an increasing set of challenges due to multiple facets of the network environment. It is envisaged that identifying users via their network activities/interactions rather than IP traffic would offer additional information to network forensic analysts when an incident occurs, such as *how* an attack was formed, *what* actions the attacker carried out, and more importantly *who* the attacker was.

2.3. *Biometric-based behavioural profiling*

Behavioural profiling is a biometric method that identifies users based upon the way in which they interact with services/systems on their IT devices. In general, user interactions can be extracted from two levels: service/application-level (local) or system-level (global). At the service level, user interactions are more application specific. For example, within a telephony application (e.g., Skype), the following user interactions can be observed, including caller's number, the time of calling, and the duration of calling, while for a word-processing application, user interactions include the name of the document, the editing time, and the overall length of the document. System-level user interactions are more generic, such as the name of the running applications, the time of individual application usage, and the amount of processing power being used. Within a behavioural profiling system, user's interactions are compared with existing profile(s) that are created based upon the historical service usage with their identity being verified/identified based upon the comparison result.

Aupy and Clarke (2005) proposed a non-intrusive and continuous authentication method by monitoring user's daily PC usage (e.g., applications opened and websites browsed). Their empirical study employed a private dataset, containing 21 users' data over a two-month period. By using Feed Forward Multi-Layered Perceptron neural network as the classifier method, an Equal Error Rate (EER) of 7.1% was obtained. Song et al 2013 suggested an identification model by studying user's system level behaviour (e.g., process creation, registry key changes and file system actions). Their evaluation was carried out on a dataset with 18 users over four weeks on a Windows based computer. Their system accuracies were in the range of 50%–90% via their Gaussian mixture model classifier. Deutschmann and Lindholm (2013) demonstrated a continuous authentication system by observing user's mouse movement, keystrokes and application usage within an office environment. Their dataset was captured from 99 users over a 10 week period. The developed trust model can detect an incorrect user in just over 1.5 minutes and allow correct users to work through a regular working day. Li et al. (2014) presented an active authentication method for mobile devices by utilizing user's application interactions. Their system obtained an EER of 9.8% by using the combination of a rule-based classifier, a dynamic profiling technique and a smoothing function on a 76-user dataset.

As demonstrated above, studies have demonstrated that users can be identified/verified via their interactions; and a good level of performance is also observed. However, existing studies were focused upon capturing information from the device itself (i.e., via an agent installed on the computer or mobile device). In terms of user network service interactions, Conti et al. (2016) have suggested that user's application actions can be identified by analysing the network traffic of their Android mobiles; however, their analysis is based specifically on the analysis of a very limited set of apps rather than an analysis of web-based traffic – the nature of the resulting network signal being very different. Furthermore, the study did not seek to use this information as a basis for identifying users.

3. Deriving user interactions from network metadata

The underlying premise or hypothesis of the proposed approach is that different user interactions within network-based applications would result in network connections whose composition would reveal a deterministic (or worst-case probabilistic) measure that would intrinsically map to a specific user action. For instance, when users chat via an instant messenger application, it is intuitive that a longer datagram could

Table 3 – Services and user interactions.

Services	User interactions
BBC	Page navigation, watch video clips/TV programs, listen to audio clips/radio, comment, news sharing
Dropbox	Download/upload files, share files/folders, folder navigations
Facebook; Twitter	Post, comment, share, find friends, attach files, chat, typing message, like
Google	Keyword search, page navigation, create/edit/share/delete online documents
Hotmail	Download/upload file attachments, compose/reply/forward/delete an email, insert recipients, read emails
Skype	Send text messages, file transfer, click on contacts, audio/video call, change online presence
YouTube	Search, watch videos, listen to songs, upload songs/videos, like, dislike, comment
Wikipedia	Search, read/modify an article, upload media files

indicate a larger amount of characters being sent during the conversation. Using flow-based network analysis is not sufficient because more than one user interaction might exist within a single flow. For example, with TLS-based connections, the TLS session remains open for a period of time (to remove the overhead in creating the session) which results in many user interactions being possible within a single flow. Conversely, a packet-based approach does not provide the necessary abstraction required to understand the nature of the user interaction. Understanding what a user is actually doing provides the basis for subsequently using the information to profile and identify them.

Based upon this working hypothesis, a series of experiments were conducted to investigate the relationship between user activities/interactions and the corresponding network signals that result within various Internet based services. To aid this process and provide a starting point for the analysis, nine of the most popular Internet services were identified (Alexa, 2016). In each application, all possible user interactions were identified – resulting in Table 3. A deductive methodology was then applied to identify whether each interaction identified resulted in a unique network signature and if it did, what the discerning parameters were. This was performed on a manual basis – using Wireshark to capture the traffic and perform the analysis. To provide a level of rigor and verification in this process, each interaction was captured at least 10 times (with particular interactions such as chatting resulting in a larger number of captures to investigate the role of the length of the message on the resulted network traffic). This was also then repeated by three independent researchers working in a blind mode (i.e., they were not privy to the results of the observations made by the other researchers). The results were then combined and verified.

A list of the network metadata parameters utilized for the analysis was as follows: the date and time stamp, the IP address of the source and destination, the port numbers, protocol ID (either TCP or UDP), the length of a datagram, and several TCP flags (e.g., SYN, FIN, ACK and PUSH). According to the packet sending rate, user actions over a secured communication (e.g., HTTPS) can be represented in three forms via their network metadata: a single packet, multiple packets, and a stream of packets. User's Internet activities over a plain-text traffic can be observed according to the SYN and FIN flags of the TCP protocol; as a result, the majority of user's unsecured traffic will be interpreted as a stream of packets. Both directions of network traffic were analysed, as it was envisaged that the nature of the returning traffic from the Internet service could also provide meaningful information as to the nature of the interaction.

In order to aid in understanding the nature of the analysis, three examples are provided that represent the single, multiple, and streamed packet contexts.

Example 1: single packet analysis

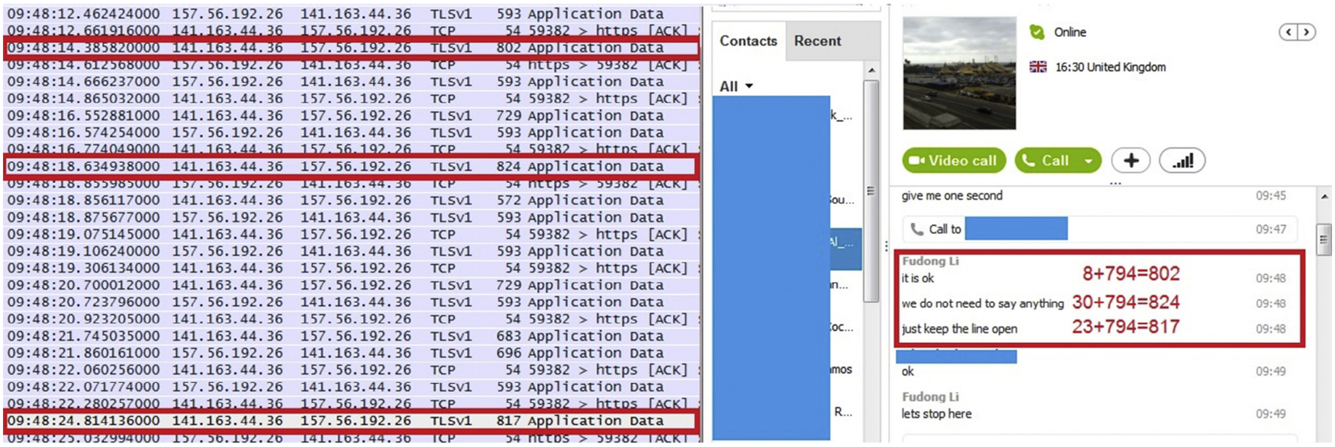
For certain interactions, when a user activity is carried out, a single packet with the same payload length and TCP flag status occurs. For example, the length of a chatting message can be identified within the Skype service. The packet length is composed of a baseline set of characters. In the example illustrated in Fig. 1, this is 794 bytes with three messages with 8, 30, 23 characters being sent from the client to the receiving party. Upon repeating this experiment, it is noticed that the baseline set does vary between users and thus a threshold will need to be identified on a per-user basis in order to identify this interaction. Another example, a single TCP packet with a length of 937-byte is sent by the Hotmail server to the client when a recipient is added to or removed from an email conversation (including new, reply, and forward). A third example observed shows that two individual TCP packets with length of 60 bytes and 95 bytes are sent from the Dropbox server to the client when a document is shared.

Example 2: multiple packet analysis

User activities can also be observed in the form of multiple packets (normally between 2 and 4 packets). These packets are being sent in less than a millisecond timeframe. For instance, within the chatting service of Facebook, two TCP packets with lengths of 1434 and 68 bytes are sent from the client to the server when the typing activity is commenced (as illustrated in Fig. 2). In the same figure, different lengths of packets (i.e., 1169 bytes and 333 bytes) are observed when the same user activity is carried out. Nonetheless, the total length of the two sets of packets is the same (i.e. 1502 bytes). The baseline for chatting on Facebook is a total of 2625 bytes (i.e. 1434 + 1191). For example, when a 4-character word is sent to the server, a total of 2629 bytes appeared on the network. Another example is that when a user edits the Spreadsheet application of Google Docs online, two TCP packets with lengths of 1434 and 912 bytes are sent from the client to the server; and the server replies with two packets with lengths 111/112 and 159 bytes (as illustrated in Fig. 3).

Example 3: streamed packet analysis

The third user interaction pattern can be presented in the style of a stream of packets when certain user activities are performed.



```
09:48:12.462424000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:12.661916000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:14.385820000 141.163.44.36  157.56.192.26  TLSv1   802 Application Data
09:48:14.612568000 157.56.192.26  141.163.44.36  TCP      54 https > 59382 [ACK]
09:48:14.666237000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:14.865032000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:16.552881000 141.163.44.36  157.56.192.26  TLSv1   729 Application Data
09:48:16.574254000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:16.774049000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:18.634938000 141.163.44.36  157.56.192.26  TLSv1   824 Application Data
09:48:18.855985000 157.56.192.26  141.163.44.36  TCP      54 https > 59382 [ACK]
09:48:18.856117000 141.163.44.36  157.56.192.26  TLSv1   572 Application Data
09:48:18.875677000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:19.075145000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:19.106240000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:19.306134000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:20.700012000 141.163.44.36  157.56.192.26  TLSv1   729 Application Data
09:48:20.723796000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:20.923205000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:21.745035000 141.163.44.36  157.56.192.26  TLSv1   683 Application Data
09:48:21.860161000 157.56.192.26  141.163.44.36  TLSv1   696 Application Data
09:48:22.060256000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:22.071774000 157.56.192.26  141.163.44.36  TLSv1   593 Application Data
09:48:22.280257000 141.163.44.36  157.56.192.26  TCP      54 59382 > https [ACK]
09:48:24.814136000 141.163.44.36  157.56.192.26  TLSv1   817 Application Data
09:48:25.032994000 157.56.192.26  141.163.44.36  TCP      54 https > 59382 [ACK]
```

give me one second 09:45

Call to [redacted] 09:47

Fudong Li
it is ok 8+794=802 09:48
we do not need to say anything 30+794=824 09:48
just keep the line open 23+794=817 09:48

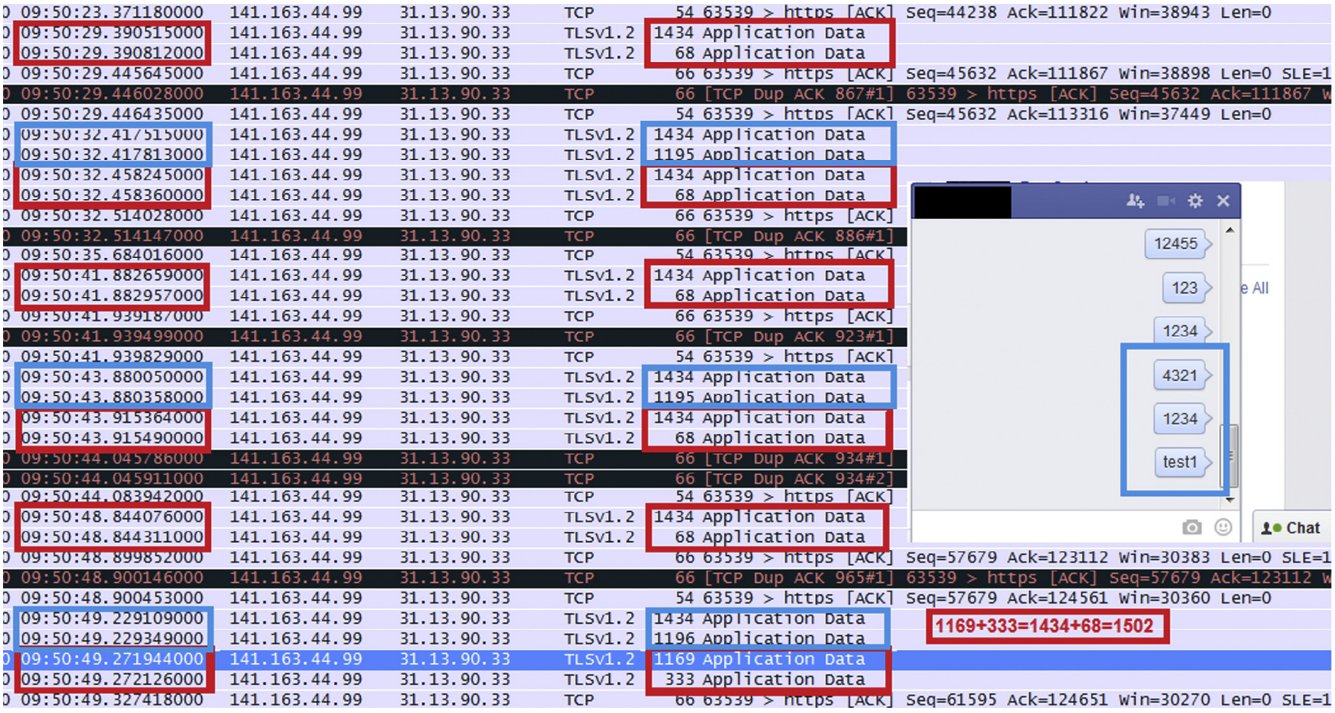
[redacted]
ok 09:49

Fudong Li
lets stop here 09:49

Fig. 1 – User chatting over Skype.

These packets are sent within the range of microseconds (e.g., 50 microseconds) from each other in a continuous fashion; and normally they have same or similar payload lengths. The most common example is user's uploading or downloading activities: in the case of uploading, streams of Maximum Transmission Unit (MTU) size packets are sent from the user's computer to the server; while the server only replies with acknowledgement packets. Another example is when users have a video-conference via Skype. As demonstrated in Fig. 4, the connection is set up directly between two clients via UDP ports (i.e., not via the Skype server). During the video conference, one client was sending larger size packets (e.g., 1166–1360 byte video frames) while the other was sending much smaller packets (e.g., 138–149 bytes voice frames) as the former client had the camera turned on while the latter did not.

A summary of the experimental results on selected common user interactions from the aforementioned Internet service is presented in Table 4. Obviously many user actions can be performed via thousands of Internet based services and this study only manually analysed a number of common user interactions from 9 popular applications. Nevertheless, these results in Table 4 along with the information illustrated in Figs 1–4 provide enough evidence suggesting that user's application level activities can be presented by their corresponding network level metadata, without needing to check the payload, hence protecting user's privacy and reducing the computation power required. It should also be highlighted that the nature of the resulting signatures are focused upon the core user interaction (i.e., the messaging component, the payload of a voice call) rather than utilizing a wider set of network flows/interactions



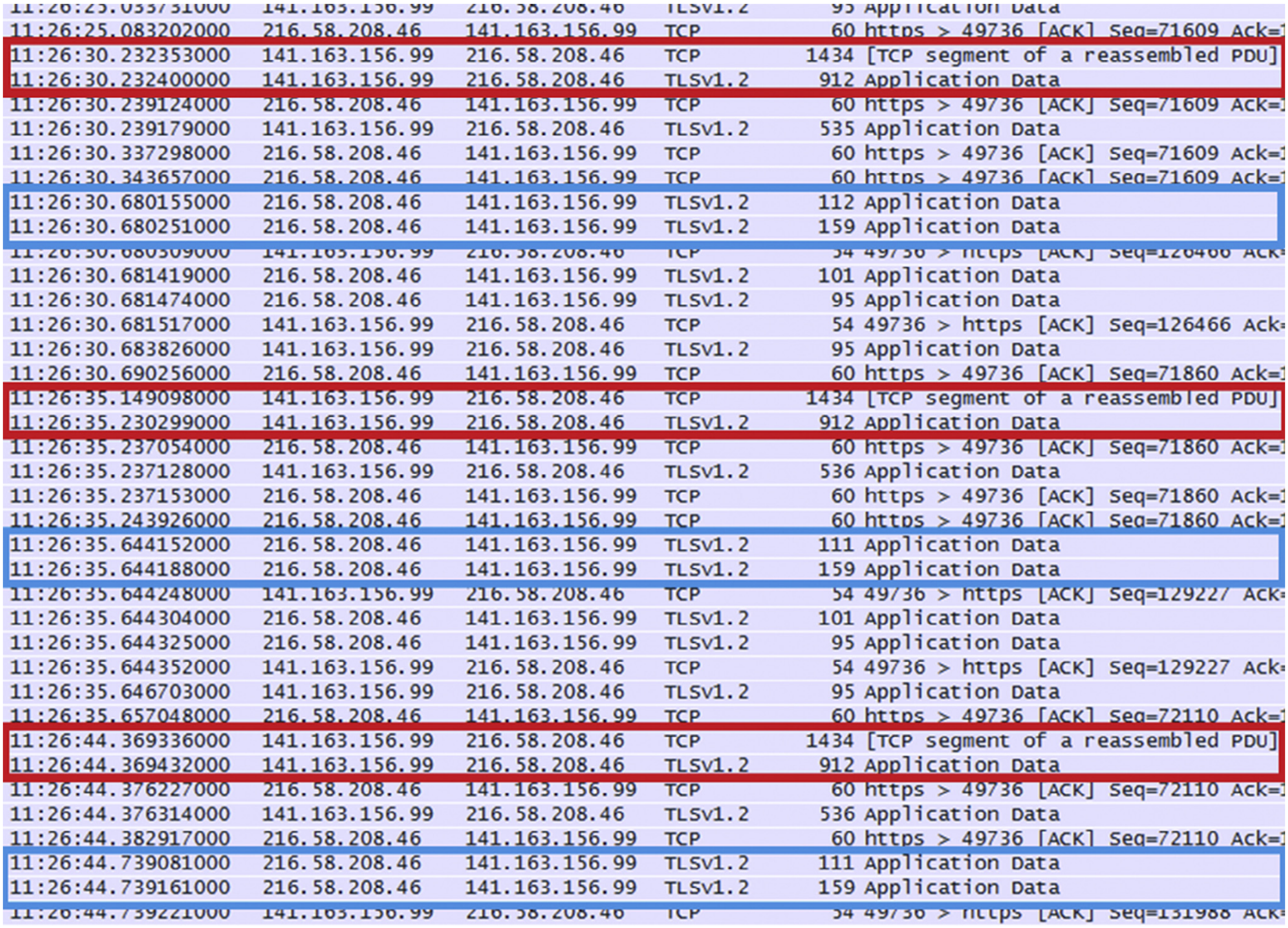
```
09:50:23.371180000  141.163.44.99  31.13.90.33  TCP      54 63539 > https [ACK] Seq=44238 Ack=111822 Win=38943 Len=0
09:50:29.390515000  141.163.44.99  31.13.90.33  TLSv1.2 1434 Application Data
09:50:29.390812000  141.163.44.99  31.13.90.33  TLSv1.2   68 Application Data
09:50:29.445645000  141.163.44.99  31.13.90.33  TCP      66 63539 > https [ACK] Seq=45632 Ack=111867 Win=38898 Len=0 SLE=1
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=111867 W
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]
09:50:32.514147000  141.163.44.99  31.13.90.33  TCP      66 [TCP Dup ACK 886#1]
09:50:35.684016000  141.163.44.99  31.13.90.33  TCP      54 63539 > https [ACK]
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]
09:50:41.939499000  141.163.44.99  31.13.90.33  TCP      66 [TCP Dup ACK 923#1]
09:50:41.939829000  141.163.44.99  31.13.90.33  TCP      54 63539 > https [ACK]
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]
09:50:44.045911000  141.163.44.99  31.13.90.33  TCP      66 [TCP Dup ACK 934#2]
09:50:44.083942000  141.163.44.99  31.13.90.33  TCP      54 63539 > https [ACK]
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 SLE=1
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=123112 W
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 SLE=1
```

[redacted]

12455
123
1234
4321
1234
test1

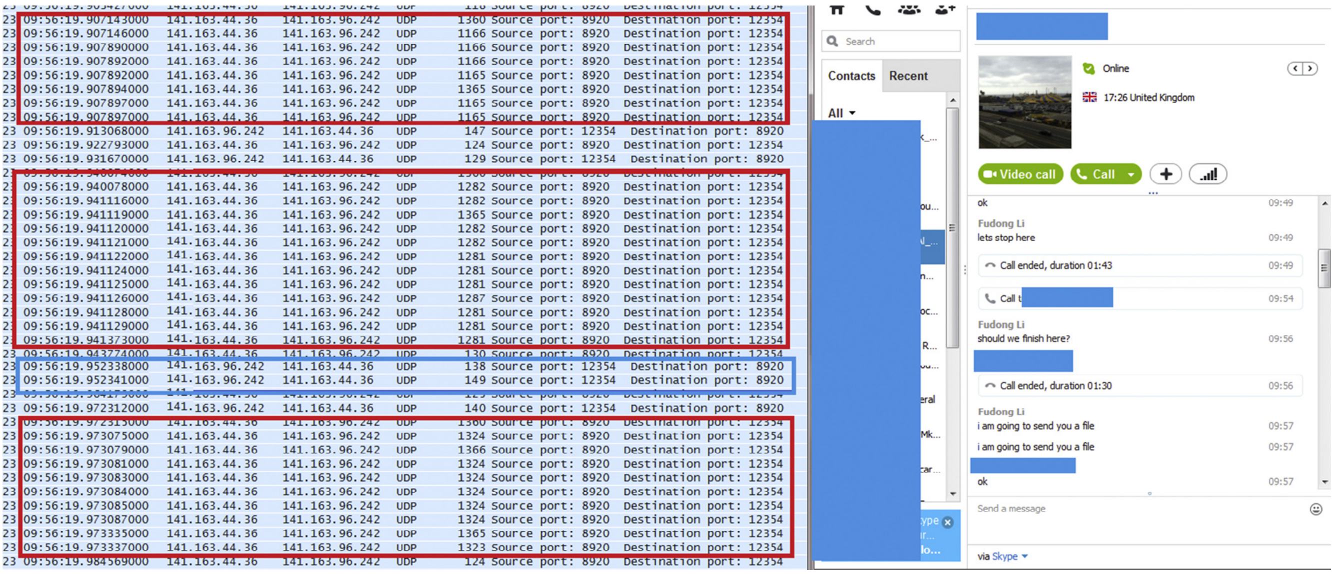
1169+333=1434+68=1502

Fig. 2 – User chatting on Facebook.



```
11:26:25.033731000  141.163.156.99  216.58.208.46   TLSv1.2    95 Application Data
11:26:25.083202000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71609 Ack=1
11:26:30.232353000  141.163.156.99  216.58.208.46   TCP      1434 [TCP segment of a reassembled PDU]
11:26:30.232400000  141.163.156.99  216.58.208.46   TLSv1.2   912 Application Data
11:26:30.239124000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71609 Ack=1
11:26:30.239179000  141.163.156.99  216.58.208.46   TLSv1.2   535 Application Data
11:26:30.337298000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71609 Ack=1
11:26:30.343657000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71609 Ack=1
11:26:30.680155000  216.58.208.46   141.163.156.99  TLSv1.2   112 Application Data
11:26:30.680251000  216.58.208.46   141.163.156.99  TLSv1.2   159 Application Data
11:26:30.680309000  141.163.156.99  216.58.208.46   TCP        54 49736 > https [ACK] Seq=126466 Ack
11:26:30.681419000  216.58.208.46   141.163.156.99  TLSv1.2   101 Application Data
11:26:30.681474000  216.58.208.46   141.163.156.99  TLSv1.2    95 Application Data
11:26:30.681517000  141.163.156.99  216.58.208.46   TCP        54 49736 > https [ACK] Seq=126466 Ack
11:26:30.683826000  141.163.156.99  216.58.208.46   TLSv1.2    95 Application Data
11:26:30.690256000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71860 Ack=1
11:26:35.149098000  141.163.156.99  216.58.208.46   TCP      1434 [TCP segment of a reassembled PDU]
11:26:35.230299000  141.163.156.99  216.58.208.46   TLSv1.2   912 Application Data
11:26:35.237054000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71860 Ack=1
11:26:35.237128000  141.163.156.99  216.58.208.46   TLSv1.2   536 Application Data
11:26:35.237153000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71860 Ack=1
11:26:35.243926000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=71860 Ack=1
11:26:35.644152000  216.58.208.46   141.163.156.99  TLSv1.2   111 Application Data
11:26:35.644188000  216.58.208.46   141.163.156.99  TLSv1.2   159 Application Data
11:26:35.644248000  141.163.156.99  216.58.208.46   TCP        54 49736 > https [ACK] Seq=129227 Ack
11:26:35.644304000  216.58.208.46   141.163.156.99  TLSv1.2   101 Application Data
11:26:35.644325000  216.58.208.46   141.163.156.99  TLSv1.2    95 Application Data
11:26:35.644352000  141.163.156.99  216.58.208.46   TCP        54 49736 > https [ACK] Seq=129227 Ack
11:26:35.646703000  141.163.156.99  216.58.208.46   TLSv1.2    95 Application Data
11:26:35.657048000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=72110 Ack=1
11:26:44.369336000  141.163.156.99  216.58.208.46   TCP      1434 [TCP segment of a reassembled PDU]
11:26:44.369432000  141.163.156.99  216.58.208.46   TLSv1.2   912 Application Data
11:26:44.376227000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=72110 Ack=1
11:26:44.376314000  141.163.156.99  216.58.208.46   TLSv1.2   536 Application Data
11:26:44.382917000  216.58.208.46   141.163.156.99  TCP        60 https > 49736 [ACK] Seq=72110 Ack=1
11:26:44.739081000  216.58.208.46   141.163.156.99  TLSv1.2   111 Application Data
11:26:44.739161000  216.58.208.46   141.163.156.99  TLSv1.2   159 Application Data
11:26:44.739221000  141.163.156.99  216.58.208.46   TCP        54 49736 > https [ACK] Seq=131988 Ack
```

Fig. 3 – User editing a Spreadsheet on Google Docs.



```
23 09:56:19.905427000  141.163.44.36   141.163.96.242  UDP   118 Source port: 8920  Destination port: 12354
23 09:56:19.907143000  141.163.44.36   141.163.96.242  UDP  1360 Source port: 8920  Destination port: 12354
23 09:56:19.907146000  141.163.44.36   141.163.96.242  UDP  1166 Source port: 8920  Destination port: 12354
23 09:56:19.907890000  141.163.44.36   141.163.96.242  UDP  1166 Source port: 8920  Destination port: 12354
23 09:56:19.907892000  141.163.44.36   141.163.96.242  UDP  1166 Source port: 8920  Destination port: 12354
23 09:56:19.907892000  141.163.44.36   141.163.96.242  UDP  1165 Source port: 8920  Destination port: 12354
23 09:56:19.907894000  141.163.44.36   141.163.96.242  UDP  1365 Source port: 8920  Destination port: 12354
23 09:56:19.907897000  141.163.44.36   141.163.96.242  UDP  1165 Source port: 8920  Destination port: 12354
23 09:56:19.907897000  141.163.44.36   141.163.96.242  UDP  1165 Source port: 8920  Destination port: 12354
23 09:56:19.913068000  141.163.96.242  141.163.44.36   UDP   147 Source port: 12354  Destination port: 8920
23 09:56:19.922793000  141.163.44.36   141.163.96.242  UDP   124 Source port: 8920  Destination port: 12354
23 09:56:19.931670000  141.163.96.242  141.163.44.36   UDP   129 Source port: 12354  Destination port: 8920
23 09:56:19.940074000  141.163.44.36   141.163.96.242  UDP  1360 Source port: 8920  Destination port: 12354
23 09:56:19.940078000  141.163.44.36   141.163.96.242  UDP  1282 Source port: 8920  Destination port: 12354
23 09:56:19.941116000  141.163.44.36   141.163.96.242  UDP  1282 Source port: 8920  Destination port: 12354
23 09:56:19.941119000  141.163.44.36   141.163.96.242  UDP  1365 Source port: 8920  Destination port: 12354
23 09:56:19.941120000  141.163.44.36   141.163.96.242  UDP  1282 Source port: 8920  Destination port: 12354
23 09:56:19.941121000  141.163.44.36   141.163.96.242  UDP  1282 Source port: 8920  Destination port: 12354
23 09:56:19.941122000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.941124000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.941125000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.941126000  141.163.44.36   141.163.96.242  UDP  1287 Source port: 8920  Destination port: 12354
23 09:56:19.941128000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.941129000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.941373000  141.163.44.36   141.163.96.242  UDP  1281 Source port: 8920  Destination port: 12354
23 09:56:19.943774000  141.163.44.36   141.163.96.242  UDP   130 Source port: 8920  Destination port: 12354
23 09:56:19.952338000  141.163.96.242  141.163.44.36   UDP   138 Source port: 12354  Destination port: 8920
23 09:56:19.952341000  141.163.96.242  141.163.44.36   UDP   149 Source port: 12354  Destination port: 8920
23 09:56:19.964179000  141.163.44.36   141.163.96.242  UDP   129 Source port: 8920  Destination port: 12354
23 09:56:19.972312000  141.163.96.242  141.163.44.36   UDP   140 Source port: 12354  Destination port: 8920
23 09:56:19.972313000  141.163.44.36   141.163.96.242  UDP  1360 Source port: 8920  Destination port: 12354
23 09:56:19.973075000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973079000  141.163.44.36   141.163.96.242  UDP  1366 Source port: 8920  Destination port: 12354
23 09:56:19.973081000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973083000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973084000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973085000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973087000  141.163.44.36   141.163.96.242  UDP  1324 Source port: 8920  Destination port: 12354
23 09:56:19.973335000  141.163.44.36   141.163.96.242  UDP  1365 Source port: 8920  Destination port: 12354
23 09:56:19.973337000  141.163.44.36   141.163.96.242  UDP  1323 Source port: 8920  Destination port: 12354
23 09:56:19.984569000  141.163.44.36   141.163.96.242  UDP   124 Source port: 8920  Destination port: 12354
```

[redacted]

[redacted]

Online

17:26 United Kingdom

ok 09:49

Fudong Li
lets stop here 09:49

Call ended, duration 01:43 09:49

Call t[redacted] 09:54

Fudong Li
should we finish here? 09:56

[redacted]

Call ended, duration 01:30 09:56

Fudong Li
i am going to send you a file 09:57
i am going to send you a file 09:57

[redacted]

ok 09:57

Send a message

via Skype ▾

Fig. 4 – Video conferencing via Skype.

Table 4 – User interactions – key characteristics.

Services	User interactions	No. of packets	Total length (bytes)	Main direction
BBC	Page navigation	Stream	Various	Server > Client
	Watching video clips	Stream	MTU (approx)	Server > Client
	Listening to audio clips or radio	Stream	MTU (approx)	Server > Client
Dropbox	Download files	Stream	MTU (approx)	Server > Client
	Upload files	Stream	MTU (approx)	Client > server
	Folder navigations	Multiple	155	Server > Client
Facebook	Page loading	Stream	Various	Server > Client
	Attach files	Stream	MTU (approx)	Client > server
	Chat	Multiple	2625+	Client > Server
	Typing	Multiple	1502	Client > Server
Hotmail	File attachments	Stream	MTU (approx)	Client > server
	Compose an email	One	981	Server > Client
	Insert a recipient	One	937	Server > client
	Remove a recipient	One	937	Server > client
Google Docs	Editing document	Multiple	2309	Client > Server
	Editing Spreadsheet	Multiple	2346	Client > Server
Google Search	Page navigation	Multiple	268	Server > Client
Skype	Text messages	One	794+	Client > Server
	Audio calls	Stream	Approx 140	Both Clients
	Video calls	Stream	Approx 1250	Both Clients
	File transfer	Stream	MTU (approx)	Sending Client > receiving client
	Click on contacts	One	731	Client > Server
	Idle	One	572	Client > Server
Twitter	Page loading	Stream	Various	Server > Client
	Uploading Photo	Steam	MTU (approx)	Client > Server
	Twitter	One	1267+	Client > Server
Wikipedia	Page loading	Stream	Various	Server > Client
YouTube	Watch videos	Stream	MTU (approx)	Server > Client
	Video upload	Stream	MTU (approx)	Client > Server

that exist within the majority of web applications. This results in a targeting upon the flows that actually link to the user, rather than to advertisements or supplementary data. This also helps to alleviate any issues resulting from plugins that might block content (e.g., AdBlock). Furthermore, in order to identify individuals from the network traffic, it is not necessary to profile every Internet-based application but merely those that represent a significant volume of the user's traffic – which the top applications do. It should be highlighted that whilst it is expected that web-based interactions will be identical when using desktop/laptop web browsers, the interactions coming from mobile applications (e.g., Facebook via a web browser and via a mobile app) are likely to be a little different, which would result in needing to also profile popular mobile applications.

Given the integrated nature of many web applications, it is also expected that one service might connect to another. For example, watching a YouTube video on the BBC website. In this approach, these two activities will be separately identified as a BBC and YouTube interaction, rather than a series of BBC interactions. Whilst a level of granularity will be lost, the fundamental nature of the interaction or behaviour is still being captured and therefore it should have minimal impact. Furthermore, whilst in some instances it was not possible to uniquely identify the interaction through a deterministic approach. It would however still be possible to apply a probabilistic approach to these, as each interaction can only belong to a subset of possibilities. Understanding the likelihood of each would provide a basis for labelling the interaction. However, this research did not continue to perform the probabilistic analysis, as the deterministic results have provided sufficient user-based information from which to derived and apply behavioural profiling.

However, furthering this line of research could provide an invaluable approach to providing forensic investigators with a rich and abstracted set of information about what users are actually doing on their computers using an independent source of evidence.

4. Network data collection dataset

To provide a basis for evaluating whether the aforementioned user actions can provide a reliable basis for identifying users, a dataset is required. To provide scientific rigour and statistical reliability, the following criteria were established:

(a) The dataset must contain a sufficient number of participants to provide a basis for identifying them;

(b) The dataset must contain sufficient samples across a prolonged period in order to ensure identification performance can be maintained;

(c) All network traffic meta-data from all participants is to be collected;

(d) The IP address and user must be fixed for the complete duration in order to provide a ground truth to which to label the interactions and calculate the performance.

Unfortunately, existing public datasets, including the DARPA datasets (MIT, 2015), the Cyber Defence Exercise dataset (Sangster et al., 2009), the Kyoto dataset (Song et al., 2011), the SimpleWeb's (2015) datasets, and the University of New Brunswick (UNB) ISCX datasets (UNB, 2015), are either dated or were created for specific purposes (e.g., network intrusion detection or malware detection) rather than user identification. Therefore, a new dataset was required.

4.1. *Data collection*

In total, 46 users completed the data collection process during the period of 12 November 2014 to 20 January 2015. 18 users' data were collected via a network-based approach as they were full-time PhD students within the same research laboratory. All traffic on the network metadata was monitored and captured. To ensure the relationship between the IP addresses and users, DHCP was disabled and all IPs were fixed for the duration of the study. The remaining participants were recruited from the local student population via a standalone application installed on their personal machines. Whilst the sample population is not representative of the general population, it was specifically selected as graduate and post-graduate students share a common purpose (i.e. studying) are therefore more likely to share similarities in their Internet use than would be expected more generally. As such, arguably this population represents a more difficult classification task than would be anticipated from a wider population. It also more accurately reflects the behaviours you would expect of staff working for the same organization. All participants were explicitly asked not to share their systems with other people during the collection period. Due to the nature of the study (i.e., collection of user's network activities), ethical approval was obtained and approved in advance of any data collection.

At the end of the collection process, a total of 112 Gigabytes of IP header information was accumulated. Each user's data were stored in an individual SQLite database file, and each record contains the following fields: a date and time stamp (e.g., 2015.01.15.22:38:05.587341), sender's IP address, sender's port number, receiver's IP address, receiver's port number, the packet length, the type of the traffic (e.g., TCP) and the flags (e.g., SYN and FIN).

4.2. *Data pre-processing*

The raw network traffic metadata was processed by applying the signatures obtained of the user interactions. This was achieved by identifying the nine core services using IP look up and subsequently applying the signatures to the resulting metadata. Table 5 presents an overview of the resulting interactions per service across the total population. As illustrated in the table, not all users exhibited use across all nine services (as would be expected); however, it is clear that a sufficient number of users and interactions do exist to provide a basis for performing a study on user identification.

Ultimately, the objective for pre-processing is to focus upon the user-based interaction data, whilst removing machine-to-machine network protocol traffic, with the specific purpose of extracting the user discriminative information. This results in a dataset with a higher proportion of user related information and significantly reduces the volume of data that needs to be analysed further – reducing the time and computational overhead of processing every packet for identification. The 112 GBs of metadata represents a total of 1.38 billion packets. Once the feature extraction is applied, this reduces to a total of 5,417,823 interactions, representing a 96.1% reduction in data.

Table 5 – An overview of user interactions dataset.

Services	Users	Total no. of interactions	% of interactions
BBC	30	44,847	0.8%
Dropbox	31	116,989	2.2%
Facebook	46	1,619,651	29.9%
Google	46	878,418	16.2%
Hotmail	45	303,088	5.6%
Skype	31	260,611	4.8%
Twitter	46	231,639	4.3%
Wikipedia	44	17,046	0.3%
YouTube	45	1,945,534	35.9%
	Total:	5,417,823	100%

5. User identification via network interactions

The purpose of the experiment is to determine whether the use of user interactions provided a basis to successfully identify users. It is typical in many biometric studies to investigate the nature of the classifier, in particular focusing upon the optimization problem to minimize the resulting classification errors. However, given the large volume of data within the dataset, an exhaustive iterative classifier optimization design methodology was not deemed appropriate (or would not be completed in a timely manner). As such, a preliminary experiment focusing upon subsets of the dataset was undertaken to determine appropriate classifier configurations – whilst not optimal, the subsequent results do still provide a strong indication as to the overall performance that can be achieved.

5.1. *Preliminary experiment: classification configuration*

In order to successfully identify users through their service interactions to answer the *who* question for network forensic investigations, a classifier that can discriminate individual users based upon their behavioural patterns is required. Several Artificial Intelligence (AI) techniques, such as Feedforward Multi-Layer Perceptron (FF MLP) neural network, Radial Basis Function neural network, and Self-Organizing Map, can all be utilized in the pattern classification domain (Jain et al., 1996). Amongst them, FF MLP is often chosen as the default classifier as previous studies demonstrated that a better performance was obtained over other techniques within the biometric authentication domain (Clarke and Furnell, 2006; Iranmanesh et al., 2014; Li et al., 2014; Saevanee et al., 2015; Sibai et al., 2013; Svozil et al., 1997).

The two key parameters that required configuration with a FF MLP network are the training algorithm and the number of neurons. A third parameter involving a methodological aspect was also included in the investigation – the minimum number of interactions required by a participant in order to be included in the analysis. Including users with particularly low numbers of interactions for a service would not provide sufficient samples to perform training or testing of the classifier. To reduce the overhead of processing all data, the BBC service was selected – as a service with lower volumes of interaction data but with sufficient data to make an analysis meaningful. The data were randomly split 60/40 into training and test datasets respectively. Whilst random sampling does not reflect real-world use, for these purposes it was deemed appropriate to obtain an overview of the complete dataset for both training and testing. Given typically high levels of variability in feature vectors of behavioural-based biometrics, this removes the need to consider template retraining.

A total of 315 tests were conducted through varying the following parameters:

- Minimum numbers of interaction – 50, 100, 150, 200 and 250
- Training algorithms – Levenberg–Marquardt, BFGS Quasi-Newton, Resilient Backpropagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell/Beale Restarts, Fletcher–Powell Conjugate Gradient, Polak–Ribiére Conjugate Gradient, One Step Secant, and Variable Learning Rate Backpropagation
- Number of neurons – 20, 40, 60, 80, 100, 120, and 140

Due to a number of factors (such as the complexity of the problem and the number of data points), it is difficult to know which training algorithm from the 9 chosen would be suitable for a given problem (MathWorks, 2017). Nonetheless, an analysis of the training algorithms (as illustrated in Fig. 5) clearly shows the Levenberg–Marquardt backpropagation algorithm consistently outperforming the other training algorithms across a range of network sizes for solving the issue of user identification. Using this as a basis, Fig. 6 presents the results of varying network sizes against the minimum number of interactions. The performance of the classifier increases slightly when the network size gets larger for user interaction threshold settings 150, 200 and 250; however, under the same configurations, longer training times for the FF MLP neural network and the need for additional computational resources are also observed. The best performance of 75.5% True Positive Identification Rate (TPIR) is obtained under the network with 60 neurons and 250 minimal number of user interactions. This suggests that network size does not have a large impact upon the overall performance that can be achieved (a smaller network size is preferable due to computation overheads). What is clear is the number of interactions does have a direct impact, with a significant increase in performance being achieved – an increase of 16% in the identification rate is experienced between 50 and 250 interactions. Notably, the rate of improvement significantly reduces as the number of minimum interactions is increased. This suggests that whilst increasing the minimum number of interactions further still might result in better levels of performance, the level of improvement is likely only to be slight.

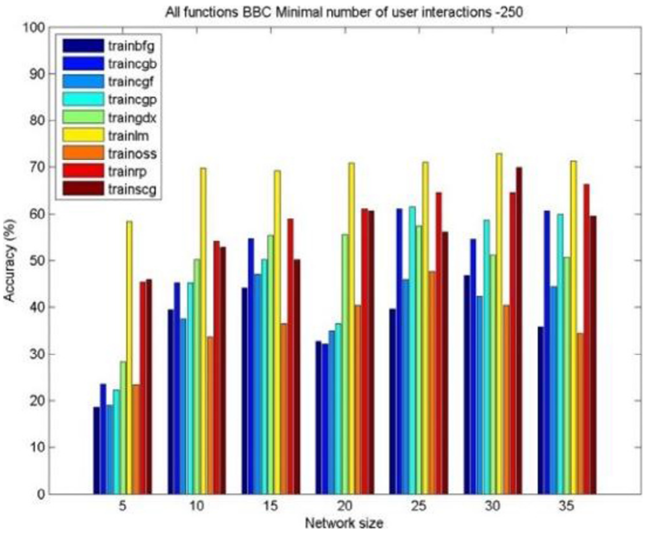


Fig. 5 – Performance achieved by varying the learning algorithm versus network size.

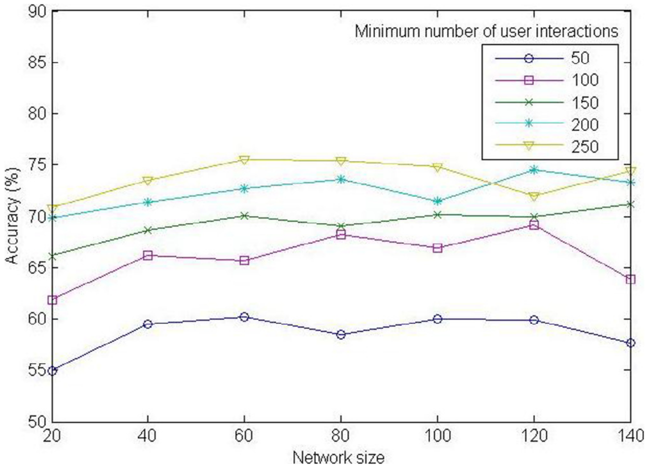


Fig. 6 – Comparison of performance with varying network sizes and minimum number of user interactions (via Levenberg–Marquardt backpropagation).

5.2. *Experiment*

Utilizing the configuration information obtained from the preliminary study, a complete evaluation was undertaken investigating the feasibility of identifying users across the nine Internet-based services. The study involved all 46 participants across the two-months of data collection period. For all 9 services, the Levenberg–Marquardt backpropagation is chosen as the training function, with a network size of 40 neurons and a minimum of 200 user interactions in order to achieve a reasonable level of performance (i.e., 71.4% TPIR) versus ensuring sufficient data are available within the dataset. The minimum number of user interactions is utilized to identify which users are eligible to be included within the identification model. Sixty percent of 200 samples are then randomly selected for use in training the classifier. All remaining samples (at least 40% of 200 but more in many users) are then used for testing the dataset to calculate the performance. At no stage is a sample used for both training and testing datasets. This methodological

Table 6 – Overall identification results for each user with all services.

User ID	Average	1	2	3	4	5	6	7
TNUI	5,285,384	9,835	82,165	27,666	441,831	104,086	58,469	72,374
Top 1 (%)	47.5	62.8	12.6	32.8	19.2	69.2	51.5	23.1
Top 3 (%)	60.5	74.1	46.7	47.0	51.0	71.4	70.8	42.1
Top 5 (%)	66.0	76.6	70.5	55.4	72.0	73.7	84.8	52.8
User ID	8	9	10	11	12	13	14	15
TNUI	83,873	310,364	34,310	40,496	171,949	50,776	34,337	147,776
Top 1 (%)	43.8	42.9	39.1	39.6	26.8	42.7	42.2	39.2
Top 3 (%)	57.5	57.0	44.6	42.7	45.4	56.7	47.4	55.6
Top 5 (%)	61.3	62.3	47.8	43.9	54.3	63.8	48.3	61.0
User ID	16	17	18	19	20	21	22	23
TNUI	8,492	1,992	24,701	71,610	118,655	28,328	732	64,955
Top 1 (%)	29.9	59.4	41.6	49.7	28.6	16.0	47.0	29.0
Top 3 (%)	36.9	62.3	59.3	59.6	34.9	35.5	51.1	38.1
Top 5 (%)	38.6	63.0	71.8	67.9	37.4	50.0	54.2	43.9
User ID	24	25	26	27	28	29	30	31
TNUI	97,688	16,936	36,248	21,828	24,232	296,048	65,628	319,239
Top 1 (%)	81.8	54.7	27.1	67.1	41.0	43.4	51.6	68.6
Top 3 (%)	90.1	65.6	40.5	70.7	54.5	58.5	63.5	87.9
Top 5 (%)	92.3	67.2	48.3	70.9	61.6	67.2	66.3	92.3
User ID	32	33	34	35	36	37	38	39
TNUI	961,357	206,768	95,501	150,750	26,297	113,918	137,543	2,270
Top 1 (%)	51.9	32.9	73.7	56.4	51.0	31.1	59.6	86.3
Top 3 (%)	68.8	40.4	94.4	73.6	53.4	46.0	83.6	88.8
Top 5 (%)	74.1	45.6	96.8	77.9	55.1	50.6	87.6	89.1
User ID	40	41	42	43	44	45	46	
TNUI	104,722	16,662	13,952	97,848	290,472	87,754	111,951	
Top 1 (%)	56.7	73.9	46	64.1	63.5	67.4	45.9	
Top 3 (%)	69.5	87.0	47.8	73.7	79.1	86.8	72.8	
Top 5 (%)	73.8	90.7	48.4	76.7	81.3	90.0	77.8	

approach permitted a fixed training dataset where each user was given an identical number of training samples – this would reduce any skewing effects that would exist if the model simply permitted 60% of a user's interactions to be used for training (e.g., user X would have 600 training samples versus user Y with only 120 samples, assuming user X has 1000 samples while user Y has 200 samples in total). The test dataset however is variable based upon the remaining samples that are present (with a minimum of 80 samples (40% of 200) but in practice going up to user X with 840 samples). This provides for a fairer assessment of the overall performance that can be achieved. Three levels of TPIRs, top 1, 3 and 5, are set for ranking the performance accuracy. For instance, a top 3 result indicates that a user's interaction was identified within the highest 3 of the classifier's output results.

For each service, a single FF MLP network was created with 9 inputs, 40 hidden neurons and 46 outputs – with the highest value amongst the 46 positions indicating which the network deemed to be the user belonging to that sample. Networks were trained until the training conditions were met (e.g., 1000 epochs had been completed or 6 maximum validation failures had been reached). The inputs to the network were:

- Start time of interaction: 2014.12.09.10:45:23.769053
- End time of interaction: 2014.12.09.10:45:23.817927
- Source port number: 59477
- Service IP address: 212.58.246.93
- Service port number: 443
- Number of packets sent from source to destination: 2
- Total size of packets sent from source to destination: 1850
- Number of packets sent from destination to source: 10
- Total size of packets sent from destination to source: 13,419

The source IP address is not utilized at all in the identification process. This is a key differentiating factor over existing packet and flow based detection studies.

The overall results of the identification are presented in Table 6, with an average TPIR (Top 1) rate of 48% increasing to 66% (TPIR Top 5). Whilst in terms of identification systems, this is unacceptably low, its purpose in this context is to reduce the volume of traffic an investigator needs to analyse. Removing even 50% will have a huge impact. The table also includes the number of interactions this results are based upon (i.e., the size of the test dataset), showing this was based upon an average of 115,000 interactions per user – over 5 million in total.

In terms of individual users, the best top 1 ranking performances are 86.3%, 81.8% and 73.9% for users 39, 24 and 41 respectively, with the best rank 5 results of 96.8%, 92.3% and 90.7% for users 34, 24/31 and 41. The worst performing users included users 2 (12.6%), 4 (19.2%) and 21 (16%) rank 1 TPIR.

Table 7 – The overall performance for each individual service.

Services	Number of users	Top 1 (%)	Top 3 (%)	Top 5 (%)
BBC	11	73.1	83.7	88.5
Dropbox	19	42.2	54.7	62.8
Facebook	43	49.8	62.8	68.3
Google	44	53.8	67.5	72.6
Hotmail	28	68.3	72.5	74.0
Skype	23	90.3	92.5	93.2
Twitter	42	43.5	50.2	53.3
Wikipedia	17	63.5	71.3	74.7
YouTube	44	44.0	58.7	65.2

However, reflecting upon their performance at rank 5 shows an increase to 70.5% (user 2), 72% (user 4) and 50% (user 21), demonstrating a reasonable to high level of identification rate.

An analysis of the results per service rather than individual shows Skype, BBC and Hotmail as the top three in terms of recognition performance. Notably, in all three cases, the population sample is lower than in other services such as Google, Facebook and YouTube, which achieved lower rank 1 performances (of between 44 and 54%). Again, reflecting upon their rank 5 performance and the percentages increase to encouraging levels (65–73%). It is also worth highlighting that population size is only one of the factors, with the nature and range of the feature vector being another. Where services have performed well (in rank 1) they tend to have a richer set of possible user interactions (when compared to the interactions presented in Table 4).

Rather than using all services, a forensic investigator might wish to restrict the filtering of network traffic based upon the best performing services, in order to be more confident about the identity of the traffic. Table 7 presents the performance and name of the service for the top three services. Thirty-five of the forty-six participants have a first service performance over 80% or greater, providing a rigorous approach. These highly positive identifications can be used to confirm the user's IP address(es) which can in turn be used to filter and analyse the traffic.

An analysis of the second and third top services (from Table 8) also shows a number of users with significantly high identifications rates – 19 users with 80% or greater with second and 9 users in third. Skype, Hotmail, Facebook, BBC and Google are the most recurrent services listed in the top three services amongst all users.

6. Discussion

The need to identify and map network traffic to individuals is a key requirement in order to be able to investigate the *who*. Or indeed, even when the *who* is known, this research would enable the identification of relevant traffic in an IP-independent and more timely fashion, enabling the *what* and *how*. The contribution of this paper is not to present a complete solution where traffic is identified and merely extracted for investigation but rather to provide an additional layer of analysis where the traffic can be labelled and prioritized for analysis by the investigator (i.e., more or less likely to belong to a particular user). The solution is seeking to reduce the time taken and cognitive load upon an investigator. As such, the research question becomes less about just the recognition performance and more about how this leads to a reduction in the analysis required, although of course the two aspects are linked.

The results have presented a series of analyses that show that the use of user interactions is a reliable means of creating a behavioural profile. The performance achieved in these experiments are certainly in line with if not exceeding in many cases previous research conducted in behavioural profiling (Fridman et al., 2016; Li et al., 2014). Unfortunately, a direct comparison to prior research in flow-based approaches is not possibly due to the research having focused upon anomaly detection (a 2-class problem) rather than identification (a *n*-class problem). Additional, prior work is focused upon using the user's IP address within the classifier – something this research specifically sought not to do. However, to provide a basis for understanding the value of interactions versus flows, a further experiment was conducted. The data flows and interactions from the Dropbox service were extracted and processed (inline with the aforementioned experimental methodology in Section 5). The results of the experiment are presented in Table 9. As can be seen, the interaction-based approach outperforms the flow-based approach by approximately 10% at rank 1 TPIR, with the gap shrinking for tops 3 and 5. This suggests the interaction-based processing provides for a more discriminative feature-set than the flow-based approach.

The use of interactions introduces a number of benefits over existing approaches that include:

- Reduction in the volume of traffic that needs to be processed by the classification system over packet-based approaches and an increase in granularity over flow-based approaches
- Focusing upon the user rather than the packet or flow, enabling a clear picture of what the user is doing at the application-level
- Privacy preserving as only metadata is captured and analysed, although in practice an investigation might wish to inspect the traffic (if possible)
- Encryption independent – linked with privacy, the lack of DPI techniques removes the restriction and need to decrypt traffic prior to processing (which can be a significant processing burden)

Whilst the preliminary study has resulted in a promising set of results, two key areas require further thought and consideration: scalability and training data. The model utilized in this experiment had to deal with identifying a maximum of 46 users. In practice, it would be expected that the approach would need to successfully differentiate between hundreds (and with particularly large centralized Enterprise organizations, thousands). It is unlikely a single FF MLP as designed in this experiment would be sufficient to provide the necessary classification. Whilst the results in Table 8 show that the recognition performance of the top service is on average 87%, the increased volume of data and the need for the classifier to process and discriminate would place a significant burden upon the system. Further work is required on designing models that

Table 8 – Top 3 services per user.

User ID	Top 1	TPIR (%)	Top 2	TPIR (%)	Top 3	TPIR (%)
1	Google	90.0	Twitter	63.0	YouTube	57.7
2	Wiki	83.0	BBC	77.0	Dropbox	41.8
3	Hotmail	99.4	Google	76.5	Twitter	60.0
4	Hotmail	92.4	Google	58.3	Dropbox	41.4
5	YouTube	71.7	Google	62.9	Facebook	52.6
6	Skype	99.4	Hotmail	94.5	Google	82.6
7	Google	48.3	YouTube	18.5	Facebook	16.9
8	Twitter	57.9	Google	50.8	Facebook	43.7
9	Skype	97.6	Wiki	92.4	Hotmail	75.6
10	Skype	100.0	Hotmail	83.3	Facebook	55.2
11	Skype	97.2	Hotmail	65.6	Facebook	56.9
12	Skype	99.6	BBC	88.9	Dropbox	83.1
13	Facebook	62.3	Dropbox	55.5	Google	37.9
14	Skype	96.4	Hotmail	76.9	Twitter	51.9
15	Google	60.4	Wiki	58.9	YouTube	42.7
16	Wiki	94.5	YouTube	81.6	Google	75.4
17	YouTube	61.8	Google	57.5	–	–
18	YouTube	87.2	BBC	84.8	Facebook	69.4
19	Skype	88.2	Hotmail	78.3	BBC	51.8
20	Facebook	56.8	Wiki	54.1	Google	44.5
21	Hotmail	99.5	Twitter	88.8	Facebook	77.0
22	Google	47.8	YouTube	47.2	–	–
23	Wiki	69.0	Twitter	68.3	Google	63.8
24	Skype	97.1	Hotmail	94.8	BBC	93.0
25	Dropbox	99.6	Google	81.8	Facebook	51.6
26	Google	76.9	Twitter	51.5	YouTube	31.5
27	Skype	98.4	Hotmail	96.0	Twitter	70.6
28	Hotmail	98.1	Skype	93.8	Google	90.4
29	Skype	82.8	YouTube	61.8	BBC	55.8
30	Skype	95.2	Facebook	66.0	Hotmail	61.7
31	Skype	98.6	Hotmail	87.7	Google	87.4
32	Skype	77.5	Facebook	71.5	BBC	58.9
33	Skype	98.4	Wiki	66.8	Hotmail	55.8
34	Skype	100.0	Dropbox	94.8	BBC	93.1
35	Skype	98.0	Wiki	79.4	Hotmail	71.8
36	Google	84.7	Hotmail	76.1	YouTube	63.8
37	BBC	80.1	YouTube	71.0	Hotmail	50.2
38	Skype	94.9	BBC	89.1	Hotmail	78.9
39	Google	92.6	Dropbox	92.5	Twitter	90.7
40	Skype	96.8	YouTube	94.5	BBC	85.1
41	Twitter	89.3	Google	77.7	YouTube	63.1
42	Skype	100.0	Hotmail	88.1	Google	80.1
43	Skype	98.7	Twitter	92.6	Hotmail	51.5
44	Skype	90.9	Wiki	74.0	Google	72.6
45	Hotmail	90.0	Google	83.8	Facebook	74.2
46	Skype	100.0	Hotmail	59.2	YouTube	57.2

operate in more of a scalable-independent fashion. For example, a new system utilises a verification model per user (i.e., a two-class problem) with the resultant output from each user-based verification classifier feeding into a fusion model to determine the most likely user ID. In this manner, the discriminative effort is distributed through the use of N two-class classifiers with a single fusion engine looking to merge the result.

Table 9 – Performance comparison between flow and interaction based approaches on user's Dropbox traffic.

	Top 1 (%)	Top 3 (%)	Top 5 (%)
Flow based approach	49.3	64.9	74.2
Interaction based approach	59.8	70.3	76.3

This leads to the second area for further work – how to train the classifier. In this study, an assumption was made that training data were available in order to perform supervised training of the classifier. Whilst for the purposes of this research – to evaluate the usefulness of user interactions being derived from network metadata – it was normal to assume and label the traffic, in practice the question of where do these data come from exists. Indeed, this is a common problem across behavioural profiling studies and the wider anomaly detection work performed within Intrusion Detection Systems (IDSs). Typically, researchers merely refer to the requirement that at some point (typically at the beginning of use) data are collected and assumed to belong (either to a user or is not anomalous) (Chun, 2016). For behavioural studies where the system is merely looking to verify an individual, whilst still open to error, this is acceptable (as is the case for anomaly detection in IDSs). However, it is not possible to do so in an identification task – it would require the organization to fix IP addresses to static – which if merely left in place would remove the need for the approach as IP would be a reliable indicator of the user. As such, further work needs to be undertaken in investigating approaches that can cluster interactions through a mix-model approach of both unsupervised and supervised learning.

Whilst the focus of this research has been the classification of users without using the source (or user's) IP address, because of mobile devices and DHCP, in practice, the IP address is likely to be static in small time windows (up to a couple of minutes). Therefore, through the use of services with a high recognition performance, to provide a strong confidence of a particular user's IP address, this could then be used to associate any other traffic coming from or to that address within the time window – identifying further traffic which was not identifiable being successfully labelled. The intelligence gleaned from the recognition provides a strong indicator to enable investigators to examine certain aspects of the network traffic.

7. Conclusion and future work

This paper has presented and evaluated a novel feature extraction approach for network traffic that provides robust user identification. Through the removal of non-user related information and the transformation of features to focus specifically on application-level user interactions the study has shown this provides sufficient discriminative capacity to reliably identify individuals – particularly in services where a richer interaction environment exists. This information will provide an invaluable source of intelligence for a forensic investigator/ incident analyst to more effectively filter, refine and identify relevant network traffic.

As the discussion highlighted, scalability and identifying initial training data are both areas for future work. This will lead to the development of a next generation Network Forensic Analysis Tool (NFAT) that is capable of analysing network traffic and presenting filtering upon users (rather than IP addresses), abstracting the user actions to permit an investigator to appreciate what the nature of the traffic is from an application perspective.

Acknowledgement

This research was undertaken with the support of the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/K03345X/1 "Identifying and Modelling Victim, Business, Regulatory and Malware Behaviors in a Changing Cyber Threat Landscape".

REFERENCES

Ahmed I, Lhee K. Classification of packet contents for malware detection. J Comput Virol 2011;7(4):279–95.

Al-Bataineh A, White G. Analysis and detection of malicious data exfiltration in web traffic. In: 7th International conference on malicious and unwanted software, Fajardo, PR; 2012. p. 26–31 doi:10.1109/MALWARE.2012.6461004.

Alexa. Top sites in United Kingdom; 2016. Available from: http://www.alexa.com/topsites/countries/GB.

Androulidakis G, Chatzigiannakis V, Papavassiliou S. Using selective sampling for the support of scalable and efficient network anomaly detection. In: 2007 IEEE Globecom workshops; 2007. p. 1–5.

Aupy A, Clarke N. User authentication by service utilisation profiling. In: Proceedings of the ISOneWorld 2005, Las Vegas, USA; 2005.

Bolzoni D, Etalle S, Hartel P. POSEIDON: a 2-tier anomaly-based network intrusion detection system. In: Fourth IEEE international workshop on information assurance (IWIA'06), London; 2006. p. 10–156 doi:10.1109/IWIA.2006.18.

Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: 2010 IEEE 35th conference on local computer networks (LCN); 10–14 Oct, 2010. p. 408–15. doi:10.1109/LCN.2010.5735752.

Canini M, Fay D, Miller DJ, Moore AW, Bolla R. Per flow packet sampling for high-speed network monitoring. In: Communication systems and networks and workshops, 2009. COMSNETS 2009. First international; 5–10 Jan, 2009. p. 1–10 doi:10.1109/COMSNETS.2009.4808888.

Cho YH, Navab S, Mangione-Smith WH. Specialized hardware for deep network packet filtering. In: Proceeding of the 12th international conference, Montpellier, France; September 2–4, 2002. p. 452–61.

Chun SY. Single pulse ECG-based small scale user authentication using guided filtering. In: 2016 international conference on biometrics (ICB), Halmstad; 2016. p. 1–7 doi:10.1109/ICB.2016.7550065.

Cisco. Cisco visual networking index: forecast and methodology, 2014–2019 white paper; 2015. Available from: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html.

Cisco. NetFlow version 9; nd. Available from: http://www.cisco.com/c/en/us/products/ios-nx-os-software/netflow-version-9/index.html.

Claffy KC, Braun HW, Polyzos GC. A parameterizable methodology for Internet traffic flow profiling. IEEE J Sel Area Comm 1995;13(8):1481–94.

Claise B. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. IETF, RCF5101, January 2008.

Clarke NL, Furnell SM. Authenticating mobile phone users using keystroke analysis. Int J Inf Secur 2006;6(1):1–4.

Cohen MI. PyFlag – an advanced network forensic framework. Digit Invest 2008;5:112–20.

Conti M, Mancini L, Spolaor R, Verde N. Analyzing android encrypted network traffic to identify user actions. IEEE Trans Inf Forensics Secur 2016;11(1):114–25.

Crotti M, Gringoli F, Pelosato P, Salgarelli L. A statistical approach to IP-level classification of network traffic. In: IEEE international conference on communications, 2006. ICC '06, vol. 1. 2006. p. 170–6 doi:10.1109/ICC.2006.254723.

Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion-detection systems. Comput Netw 1999;31:805–22.

Deutschmann I, Lindholm J. Behavioral biometrics for DARPA's active authentication program. In: 2013 international conference of the BIOSIG special interest group (BIOSIG), Darmstadt; 2013. p. 1–8.

Dharmapurikar S, Lockwood JW. Fast and scalable pattern matching for network intrusion detection systems. IEEE J Sel

Areas Commun 2006;24(10):1781–92. doi:10.1109/JSAC.2006.877131.

Dharmapurikar S, Krishnamurthy P, Sproull TS, Lockwood JW. Deep packet inspection using parallel bloom filters. IEEE Micro 2004;24(1):52–61. doi:10.1109/MM.2004.1268997.

Duffield N, Lund C, Thorup M. Flow sampling under hard resource constraints, Proceedings of the joint international conference on Measurement and modeling of computer systems, pp. 85–96, ACM New York, NY, USA; 2004. doi:10.1145/1005686.1005699.

Duffield N, Lund C, Thorup M. Learn more, sample less: control of volume and variance in network measurement. IEEE Trans Inf Theory 2005;51(5):1756–75. doi:10.1109/TIT.2005.846400.

Estan C, Varghese G. New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice. ACM Trans Comput Syst 2003;21(3):270–313.

FBI. The RCFL program's annual report for Fiscal year 2014; 2014. Available from: https://www.rcfl.gov/downloads/documents/fiscal-year-2014.

FBI. 2014 internet crime report; 2015. Available from: https://pdf.ic3.gov/2014_IC3Report.pdf.

Fridman L, Weber S, Greenstadt R, Kam M. Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. IEEE Syst J 2016;(99):1–9. doi:10.1109/JSYST.2015.2472579.

He G, Zhang T, Ma Y, Xu B. A novel method to detect encrypted data exfiltration. In: Proceedings of the 2014 second international conference on advanced cloud and big data (Proceeding CBD '14). Washington, DC, USA: IEEE Computer Society; November 20–22, 2014. p. 240–6.

Hofstede R, Bartos V, Sperotto A, Pras A. Towards real-time intrusion detection for NetFlow and IPFIX. In: 2013 9th international conference on network and service management (CNSM); 14–18 Oct, 2013. p. 227–34 doi:10.1109/CNSM.2013.6727841.

Hofstede R, Čeleda P, Trammell B, Drago I, Sadre R, Sperotto A, et al. Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. IEEE Commun Surv Tutorials 2014;16(4):2037–64. doi:10.1109/COMST.2014.2321898.

Internetlivestats. Internet users; 2015. Available from: http://www.internetlivestats.com/internet-users/.

Iranmanesh V, Ahmad S, Adnan W, Yussof S, Arigbabu O, Malallah F. Online handwritten signature verification using neural network classifier based on principal component analysis. Scientific World Journal 2014;2014:Article ID 381469. doi:10.1155/2014/381469.

Jadidi Z, Muthukkumarasamy V, Sithirasenan E, Sheikhan M. Flow-based anomaly detection using neural network optimized with GSA algorithm. In: 2013 IEEE 33rd international conference on distributed computing systems workshops (ICDCSW); 8–11 July, 2013. p. 76–81 doi:10.1109/ICDCSW.2013.40.

Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: a tutorial. Computer 1996;29(3):31–44. doi:10.1109/2.485891.

Juniper. Juniper flow monitoring; 2015. Available from: http://www.juniper.net/us/en/local/pdf/app-notes/3500204-en.pdf.

Jurga RE, Hulboj MM. Packet sampling for network monitoring. In: Technical report; 2007. Available from: https://openlab-mu-internal.web.cern.ch/openlab-mu-internal/03_documents/3_technical_documents/technical_reports/2007/rj-mm_samplingreport.pdf.

Kim M, Kong H, Hong S, Chung S, Hong JW. A flow-based method for abnormal network traffic detection. In: Network operations and management symposium, 2004 (NOMS 2004), vol. 1. IEEE/IFIP. 23–23 April, 2004. p. 599–612.

Li F, Clarke NL, Papadaki M, Dowland PS. Active authentication for mobile devices utilising behaviour profiling. Int J Inf Secur 2014;13(3):229–44. ISSN:1615-5262.

Liu R, Huang N, Kao C, Chen C, Chou C. A fast pattern-match engine for network processor-based network intrusion detection system. In: International conference on information technology: coding and computing, 2004. Proceedings. ITCC 2004, vol. 1. 5–7 April, 2004. p. 97–101 doi:10.1109/ITCC.2004.1286432.

Lu H, Zheng K, Liu B, Zhang X, Liu Y. A memory-efficient parallel string matching architecture for high-speed intrusion detection. IEEE J Sel Areas Commun 2006;24(10):1793–804. doi:10.1109/JSAC.2006.877221.

Mahoney M, Chan PK. PHAD: packet Header Anomaly Detection for Identifying Hostile Network Traffic, Department of Computer Sciences, Florida Institute of Technology, Melbourne, FL, USA, Technical Report CS-2001-4, April 2001. Available from: https://cs.fit.edu/~mmahoney/paper3.pdf.

MathWorks. Choose a multilayer neural network training function; 2017. Available from: http://uk.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html.

MIT. DARPA intrusion detection data sets; 2015. Available from: http://www.ll.mit.edu/ideval/data/.

Muraleedharan N, Parmar A, Kumar M. A flow based anomaly detection system using chi-square technique. In: 2010 IEEE 2nd international Advance computing conference (IACC); 19–20 Feb, 2010. p. 285–9 doi:10.1109/IADCC.2010.5422996.

NIKSUN. NIKSUN NetDetector suite; 2016. Available from: https://www.niksun.com/product.php?id=112.

Pao T, Wang P. NetFlow based intrusion detection system. In: 2004 IEEE international conference on networking, sensing and control, vol. 2. 2004. p. 731–6 doi:10.1109/ICNSC.2004.1297037.

RSA. Threat detection and response; 2016. Available from: https://www.rsa.com/en-us/products-services/threat-detection-and-response.

Saevanee H, Clarke NL, Furnell SM, Biscioneb V. Continuous user authentication using multi-modal biometrics. Comput Secur 2015;53:234–46.

Sangster B, O'Connor TJ, Cook T, Fanelli R, Dean E, Morrell C, et al. Toward instrumenting network warfare competitions to generate labeled datasets. In: Proceedings of the 2nd conference on cyber security experimentation and test. Berkeley, CA, USA: USENIX Association; 2009.

Sflow. Sflow; 2015. Available from: http://www.sflow.org.

Sibai FN, Nuaimi A, Maamari A, Kuwair R. Ear recognition with feed-forward artificial neural networks. Neural Comput Appl 2013;23:1265. doi:10.1007/s00521-012-1068-1.

SimpleWeb. Traces; 2015. Available from: https://www.simpleweb.org/wiki/index.php/Main_Page.

Smallwood D, Vance A. Intrusion analysis with deep packet inspection: increasing efficiency of packet based investigations. In: 2011 international conference on cloud and service computing (CSC); 12–14 Dec, 2011. p. 342–7 doi:10.1109/CSC.2011.6138545.

Smith R, Goyal N, Ormont J, Sankaralingam K, Estan C. Evaluating GPUs for network packet signature matching. In: IEEE international symposium on performance analysis of systems and software, 2009. ISPASS 2009; 26–28 April, 2009. p. 175–84 doi:10.1109/ISPASS.2009.4919649.

Song J, Takakura H, Okabe Y, Eto M, Inoue D, Nakao K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In: Proceedings of the first workshop on building analysis datasets and gathering experience returns for security. New York, NY, USA: ACM; 2011. p. 29–36.

Song S, Li L, Manikopoulo CN. Flow-based statistical aggregation schemes for network anomaly detection. In: Proceedings of the 2006 IEEE international conference on networking, sensing and control, 2006. ICNSC '06; 2006. p. 786–91 doi:10.1109/ICNSC.2006.1673246.

Song Y, Salem MB, Hershkop S, Stolfo SJ. System level user behavior biometrics using Fisher features and Gaussian mixture models. In: 2013 IEEE security and privacy workshops. San Francisco, CA; 2013. p. 52–9 doi:10.1109/SPW.2013.33.

Sourdis I, Pnevmatikatos D, Wong S, Vassiliadis S. A reconfigurable perfect-hashing scheme for packet inspection. In: International conference on field programmable logic and applications, 2005; 24–26 Aug, 2005. p. 644–7 doi:10.1109/FPL.2005.1515804.

Sun Y, Valgenti VC, Kim M. NFA-based pattern matching for deep packet inspection. In: 2011 proceedings of 20th international conference on computer communications and networks (ICCCN); July 31 2011–Aug. 4, 2011. p. 1–6 doi:10.1109/ICCCN.2011.6006095.

Svozil D, Kvasnicka V, Pospichal J. Introduction to multi-layer feed-forward neural networks. Chemom Intell Lab Syst 1997;39(1):43–62.

Tegeler F, Fu X, Vigna G, Kruegel C. BotFinder: finding bots in network traffic without deep packet inspection. In: Proceeding of the 8th international conference on emerging networking experiments and technologies. New York, NY, USA: ACM; 2012. p. 349–60.

UNB. The ISCX data repository; 2015. Available from: http://www.unb.ca/research/iscx/dataset/index.html.

Verizon. 2015 data breach investigations report; 2015. Available from: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report_2015_en_xg.pdf.

Vezzani R, Baltieri D, Cucchiara R. People reidentification in surveillance and forensics: a survey. ACM Comput Surv 2013;46(2):29.

Wang K, Stolfo SJ. Anomalous payload-based network intrusion detection. In: Proceeding of the 7th international symposium RAID 2004, Sophia Antipolis, France; September 15–17, 2004. p. 203–22.

Wang K, Cretu G, Stolfo SJ. Anomalous payload-based worm detection and signature generation. In: Proceedings of the 8th international conference on recent advances in intrusion detection. Berlin, Heidelberg: Springer-Verlag; 2005. p. 227–46 doi:10.1007/11663812_12.

Wang X, Pan C, Liu P, Zhu S. SigFree: a signature-free buffer overflow attack blocker. In: Proceedings of the 15th conference on USENIX security symposium, Vancouver, BC, Canada; July 31–August 04, 2006.

Winter P, Hermann E, Zeilinger M. Inductive intrusion detection in flow-based network data using one-class support vector machines. In: 2011 4th IFIP international conference on new technologies, mobility and security (NTMS); 7–10 Feb, 2011. p. 1–5 doi:10.1109/NTMS.2011.5720582.

Wireshark. Wireshark; 2016. Available from: https://www.wireshark.org/.

Xplico. Open source network forensic analysis tool (NFAT); 2016. Available from: http://www.xplico.org/.

Yu F, Chen Z, Diao Y, Lakshman TV, Katz RH. Fast and memory-efficient regular expression matching for deep packet inspection. In: ACM/IEEE symposium on Architecture for networking and communications systems, 2006; 3–5 Dec, 2006. p. 93–102.

Zanero S. Analyzing TCP traffic patterns using self organizing maps. In: Proceedings of 13th international conference, Cagliari, Italy; September 6–8, 2005. p. 83–90 doi:10.1007/11553595_10.

Nathan L. Clarke is a Professor of Cyber Security and Digital Forensics at Plymouth University. His research interests reside in the area of information security, biometrics, and forensics and has over 180 outputs consisting of journal papers, conference papers, books, edited books, book chapters and patents. He is the Chair of the IFIP TC11.12 Working Group on the Human Aspects of Information Security & Assurance. Prof. Clarke is a chartered engineer, a fellow of the British Computing Society and a senior member of the IEEE and the author of Transparent Authentication: Biometrics, RFID and Behavioural Profiling published by Springer.

Fudong Li is a Senior Research Fellow for within the Centre for Security, Communications and Network Research (CSCAN) at the Plymouth University, where he previously completed a BSc(Hons.) degree in Computer System and Networks, an MRes degree on the subject of Network Systems Engineering and a PhD degree in Behaviour profiling for mobile devices. His research interests are behaviour profiling, user authentication, intrusion detection techniques, digital forensics, and biometrics.

Steven M. Furnell is the head of the Centre for Security, Communications & Network Research at Plymouth University (UK). His interests include mobile device security, cybercrime, user authentication, and security usability. Prof. Furnell is the author of over 260 papers in refereed international journals and conference proceedings, as well as books including Cybercrime: Vandalizing the Information Society (2001) and Computer Insecurity: Risking the System (2005). Steve is a Fellow of the BCS, a Senior Member of the IEEE, and a Board Member of the IISP. Steve has also produced a variety of security podcasts, available via www.cscan.org/podcasts.