THE JOURNAL OF
DIGITAL FORENSICS,
SECURITY AND LAW

Journal of Digital Forensics,
Security and Law

Volume 15

Article 1

August 2020

A Two-Stage Model for Social Network Investigations in Digital Forensics

Anne David
Cranfield University, a.david@cranfield.ac.uk

Sarah Morris
Cranfield University, s.l.morris@cranfield.ac.uk

Gareth Appleby-Thomas
Cranfield University, g.thomas@cranfield.ac.uk

Recommended Citation

David, Anne; Morris, Sarah; and Appleby-Thomas, Gareth (2020) "A Two-Stage Model for Social Network Investigations in Digital Forensics," *Journal of Digital Forensics, Security and Law*: Vol. 15 , Article 1.
DOI: https://doi.org/10.15394/jdfsl.2020.1667
Available at: https://commons.erau.edu/jdfsl/vol15/iss2/1

A TWO-STAGE MODEL FOR SOCIAL NETWORK INVESTIGATIONS IN DIGITAL FORENSICS

Anne David[1], Sarah Morris[2], Gareth Appleby-Thomas[3]
[1,2]Centre for Electronic Warfare, Information, and Cyber
[3]Centre for Defence Engineering
Cranfield University, Shrivenham, UK
{a.david,s.l.morris,g.thomas}@cranfield.ac.uk

ABSTRACT

This paper proposes a two-stage model for identifying and contextualizing features from artefacts created as a result of social networking activity. This technique can be useful in digital investigations and is based on understanding and the deconstruction of the processes that take place prior to, during and after user activity; this includes corroborating artefacts. Digital Investigations are becoming more complex due to factors such as, the volume of data to be examined; different data formats; a wide range of sources for digital evidence; the volatility of data and the limitations of some of the standard digital forensic tools. This paper highlights the need for an approach that enables digital investigators to prioritize social network artefacts to be further analysed; determine social connections in the context of an investigation e.g. a user's social relationships, how recovered artefacts came to be, and how they can successfully be used as evidence in court.

Keywords: digital evidence, digital forensics, social networking, relationship attribution

1. INTRODUCTION

The proliferation of social networking sites has improved ways in which users engage and form relationships with people of similar interests; going beyond the days of email communications to the use of social networking applications to share messages, photos and videos. It has also provided the opportunity for some users to perpetrate unlawful activities.

Social networking presents challenges to digital forensic investigations for example, content posted may not always be written to permanent storage media. In addition, communication content can be altered or deleted after the fact. There is a need for digital forensic investigators to be able to recover such messages or other evidence which may be used to infer user activity and sufficiently attribute an action/actions to a user.

Evidence from social networking activity may be required in different types of criminal or corporate investigations. The type of evidence recovered helps the investigator obtain useful information that could:

- Guide the initial stages of an investigation e.g. determining if a based on the evidence recovered, suspect is worth investigating further.

- Generate new leads which may lead to the:
  - Identification of other persons, places or items of interest.
  - Identification of other potential sources of digital evidence to facilitate decision making.

1.1 Contribution

The key contribution of this paper is the proposal of a two-stage model for evidence recovery and investigations involving social networking activities. It aims to help investigators prioritize digital evidence and maximize efficiency where resources are limited by focusing on extracting meaningful information from social networking artefacts. It is focused on the prompt identification and interpretation of associated artefacts and is aimed at enabling the analyst to quickly determine whether to expand or narrow the scope of an investigation.

1.2 Paper Structure

The rest of this paper is structured as follows: related work is discussed in Section 2. Section 3 proposes a two-stage model for social networking investigation in digital forensics. Section 4 discusses the experimental and analysis methodologies for this work. The research results and the implementation of the proposed model is presented in Section 5. Finally, the conclusions and potential future work stemming from this research are presented in Section 6.

2. LITERATURE REVIEW

This section provides a background on related work in digital forensics and social networking investigations. It discusses the requirements for admissibility of digital evidence and the evidential value of data generated as a result of user social networking activity.

2.1 Digital Evidence

Digital evidence can be described as any data that can be used to determine intent, culpability, how an event occurred, and the parties involved. It is useful in the investigation of a range of computer crimes and non-computer related crimes where evidence from a digital device may be used to link a suspect to an offence (Casey, 2011).

Casey defines digital evidence as “data stored or transmitted using a computer, that supports or refutes a theory on how an event occurred”. Digital evidence is crucial in digital investigation and thus must be acquired in a forensically sound manner (McKemmish, 2008) to ensure that its admissibility in a court of law.

ACPO (2012)’s definition of digital evidence encompasses a range of artefacts that can be found on digital devices for example system log files, application logs etc. Multiple devices with various artefacts, whilst ideal sources of digital evidence, present the challenge of “weeding out” information not directly relevant to the case. When time is of the essence, an investigator needs to be able to adequately identify devices that contain evidence pertinent to the case, and use the intelligence obtained from those devices to progress the investigation.

Although the processing of digital evidence varies across jurisdictions, there are a few requirements digital evidence needs to meet before it is deemed admissible in court

(Casey, 2005, 2002; Murr, 2007; Sommer, 1999):

- Evidence submitted must be **relevant** to the case.
- The evidence must be **reliable**.
- The methods used to produce the evidence must be **repeatable** and should produce the same results when applied independently by a third party.
- The evidence must be **authentic** (genuine) and can be verified using hash values generated prior to and after imaging a device.
- The evidence must be **valid** and error free. In exceptional circumstances where evidence acquisition from an active device is required, the process must be accurately documented, and any alteration accounted for.
- The evidence must be **trustworthy** and believable beyond reasonable doubt.

In order to be admissible in court, evidence from social networking activity must satisfy these requirements.

2.2 Social Networking

Social networking has been defined as "the activity of sharing information and communicating with groups of people using the internet, especially through websites that are specially designed for this purpose" (Cambridge University Press, 2019). It enables users to connect with others and to form personal or business relationships.

In the context of digital evidence acquisition, social network activity provides a plethora of digital evidence to investigators. Artefacts from web browser history, cache, cookies etc. can be used to determine and infer a relationship between a user and a social network account or another user and may also be used to attribute an action to a user. This includes, but is not limited to, determining dates and times of access, usernames, session information etc.

In spite of the advantages presented by social networking applications (instant messaging, sharing personal events, micro blogging, personal or corporate marketing, advertising etc.), it has also been known to present a means for a small minority of users to engage in disagreeable or criminal activities (Bello & DiBlasio, 2013; Jonsson, 2011; Osborne, 2010; Richards, 2007; Rankin, 2010; Select Committee on Communications, 2014; BBC News, 2012; Moore, 2014; McGuire, 2019a)

Investigating a user's (or suspect's) social network activity may be required for several reasons such as the collection of evidence to be used in court for the prosecution of an offender or for use in disciplinary actions taken against employees who abuse corporate Acceptable Use Policy (AUP) (Taylor, Haggerty, Gresty, Almond, & Berry, 2014).

The Crown Prosecution Service (CPS) (2018) proceed to trial once they are satisfied with the evidence obtained during investigations involving social networking activity. However, there are no defined guidelines for digital forensic investigators with regards to prioritizing evidence collection and the management of artefacts related to social networking.

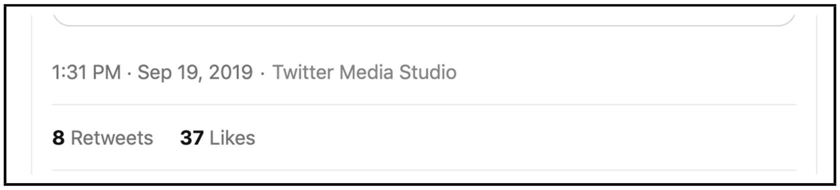
Taylor et al. (2014) suggests that although there are no specific guidelines for the forensic investigation of social networking applications, ACPO Guidelines can be used as a starting point for the investigation of offences committed through or with a social networking application. It can thus be inferred that the lack of defined guidelines often results in such investigations being broadly categorized under 'web browser forensics' (Cusack & Son, 2012) due to the nature of access on a computer (while comparable, access through mobile devices is not considered here as it is considered outside the scope of this paper). However, it is suggested that focus on features specific to social networks e.g. user IDs, profile IDs, etc. can also be used as a viable technique for evidence acquisition using methods tailored to web browser forensics.

Keyvanpour, Moradi, and Hasanzadeh (2014) presents a three-phase framework for social network forensics however, the specifics and potential location of artefacts of interest and techniques for recovery were not discussed. Oh, Lee, and Lee (2011) proposed an integrated method for the collection and analysis of web browser evidence where multiple web browsers have been used in the commission of an offence. This is based on the need to recover and utilize data created and stored on disk when a user accesses social networking sites using a web browser. It is important to note that due to the nature and flexibility of social networking applications, materials posted or shared can be later modified or deleted. In such situations, the service provider may be in a position to provide the evidence required to determine the author, when content was modified (or deleted) and reconstruct events. However, activities such as evidence retrieval from service providers is considered out of scope for this paper as most practitioners, for example in the UK, will not have access to the service provider hence the reliance on "dead disk" forensics which is more common and accessible.

Although at the time of writing there is a research gap in the use of social network artefacts as digital evidence (Powell & Haynes, 2019; Huber et al., 2011; Jang & Kwak, 2015; Zainudin, Merabti, & Llewellyn-Jones, 2011; Shaw, Das, & Mehdi, 2016; Arshad, Jantan, & Omolara, 2019; Taylor et al., 2014), there are a number of reported cases where evidence from social networking activity has successfully led to prosecution (Wood, 2018; Agency, 2015; Press Association, 2014; Bowcott, Carter, & Clifton, 2011; Haroon & Carter, 2010; BBC, 2010).

In discussing evidence collection from social network activity, Arshad et al. (2019) grouped social network artefacts into four distinct classes, User, Activity, Network and Content:

- User: consists of user data such as profile information, name, email address, phone numbers etc.
- Activity: consists of a timeline of user actions logged by the service provider on the server side e.g. dates and time of activity, location information, source of post e.g. phone, tablet, third party app etc. These types of artefacts are created as a result of user actions on the social networking site for example, when a user posts a comment about the service at a restaurant, the service provider tags the comment with the date and time it's posted (see Figure 1). The location may also be included if geolocation is enabled. The user is unable to directly modify these types of artefacts.
- Network: consists of personal social connections such as individuals or groups following or being followed by a user.
- Content: consists of materials published



1:31 PM · Sep 19, 2019 · Twitter Media Studio

8 Retweets **37** Likes

Figure 1. Activity time, date, source of tweet, number of retweets and likes illustrating server-side elements (source: `https://www.twitter.com`)

by a user such as photos, videos, tweets, retweets, shares etc.

Artefacts from each group can independently be used to infer user activity and when combined can be used to corroborate other related artefacts found on disk.

It is important to note that there is currently a knowledge gap with regards to formalizing the acquisition and analysis of social network artefacts. Some of the existing (traditional) digital forensic tools are not wholly designed for social networking investigations and the capability for targeted searches or the ability of the tool to interpret and present the evidence in a human readable format may be limited (Cusack & Son, 2012). For example, artefacts like Windows Registry Hives, Event logs, SQLite databases may need to be extracted and analyzed with a third-party tool. The objective of this paper is consequently to propose an approach that can be applied as a formal technique for social network investigations.

2.3 Extracting Features from Social Networking Artefacts

Due to the proliferation of devices and the volume of data investigators need to process, it is often crucial to quickly identify content of interest prior to a detailed analysis of a seized device. Feature extraction is an approach that enables investigators to process vast quantities of data in an efficient manner (see Garfinkel (2006, 2013) for more on feature extraction).

A user's interaction with a digital device (computer) is a two-way process aptly explained by Locard's Exchange Principle (Chisum & Turvey, 2000) which states that every contact leaves a trace (Locard 1934, pp. 7-8 as cited in Chisum and Turvey (2007, pp. 23-24)). With regards to digital evidence, this principle can be adopted to explain the existence of artefacts created as a result of user activity. For example, creating a user account; installing an application such as a web browser; or visiting social networking sites, all leave traces that can be used to infer what a user has done.

In the context of this paper, feature identification (and extraction) is described as the process of identifying and extracting artefacts containing key information about a user's social networking activities. Features in this context can be extracted from the absolute path of a given URL or other related artefacts such as HTML or JSON data, using pattern matching methods such as Regular Expressions (RegEx). The reoccurrence of a given feature can thus be attributed to a user's repeated access to a resource on a social networking site (Garfinkel, 2006).

2.3.1 Identify Features in URLs

Every website visited by a user has a URL which indicates where resources are located, and the protocol used in accessing those resources. RFC 1738 describes URL as a compact string representation for a resource available via the Internet (Berners-Lee, Masinter, & McCahill, 1994). It also describes the URL syntax as being made up of the following components:

`<scheme>:<scheme-specific-part>`

The <scheme> part of the syntax defines the protocol used e.g. ftp, http; while the <scheme-specific-part> varies and is dependent on the protocol used. For example, two typical protocols comprise of:

`ftp://user@host.domain/directory/filename`

or

`http://domain_name/path/query-searchpart-parameter-fragment`

An example of a HTTP URL with three parts is shown in Figure 2. The query or searchpart of a URL may also be complex, having several parameters as seen in Figure 3. Some URLs may also specify subdomains or port numbers.

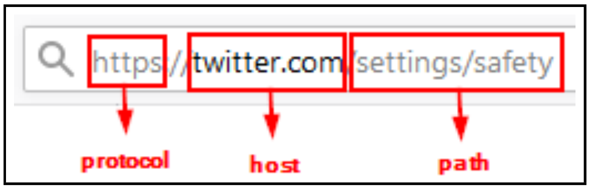
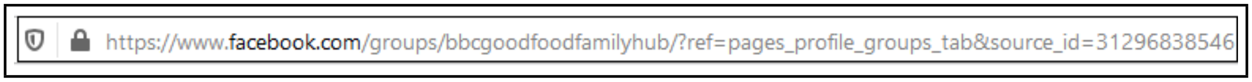


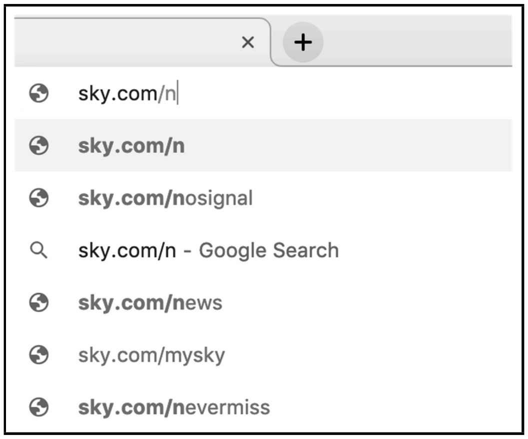
Figure 2. An example showing the parts of a HTTP URL



https://www.facebook.com/groups/bbcgoodfoodfamilyhub/?ref=pages_profile_groups_tab&source_id=31296838546

Figure 3. Example HTTP URL with multiple query or searchpart parameters

URLs can be generated in several ways for example, clicking on a link in an email or a web page; clicking a bookmark or a shortcut; typing an address in the browser address bar, or using the autocomplete feature in the browser as shown in Figure 4.



sky.com/n
sky.com/n
sky.com/nosignal
sky.com/n - Google Search
sky.com/news
sky.com/mysky
sky.com/nevermiss

Figure 4. An example of the URL autocomplete/suggestion feature

URLs can provide a lot of information about what a user has been doing online – from pages accessed, to content searched for or shared. For this research, considering that URL structure is not browser dependent, identifying features in URLs involves decomposing the URL and decoding everything after the protocol part of the URL (i.e. <scheme-specific-part>) and in some instances, everything after the domain name part. This includes the path, parameters and their values and fragments.

For investigations involving social network activities, finding and recovering actionable intelligence/evidence will help narrow the scope and refocus an investigation, thus maximizing the use of the investigator's time. Although there has been research in the forensic investigation of online activity and web browser artefacts, there is knowledge gap with regards to the deconstruction of individual URLs in relation to social networking activity. There is also limited research specifically focused on the modelling of social network forensic investigations processes and the extraction of features from deconstructed URLs.

The research described in this paper highlights the need for a concise model for the investigation and analysis of such social media related artefacts. It also highlights the importance of features in understanding user intent and the attribution of actions to a user.

3. PROPOSED MODEL

Building on the identified need for a new approach to forensic analysis of social media data, there are two key areas to consider; understanding user activity and the types of artefacts which this can generate, as well as the links between the two.

3.1 Understanding "User Activity"

In general, there are three basic ways a user interacts with a computer:

1. Create/modify: This includes the creation of new files e.g. documents; modifying existing files; installing applications, file upload or download.

2. Read/access: This includes accessing files with read, write, execute permissions. This type of interaction includes actions like launching a web browser (execute); generating an entry in the browser history (write); opening a folder or file (read).

3. Delete/remove files: This includes deliberate or unintentional removal of a file or application e.g. uninstall, delete.

In the context of this paper, user activity falls into three broad groups comprised of a mix of the basic interactions described above:

- Pre-browser activities:
  - These set of activities create (or ***modify***) artefacts that can be used to infer user activity. For example, powering on a device (computer) or a login would ***create*** an entry in the event log. Installing an application would ***create*** artefacts associated with setup and configuration. Launching the web browser would ***read*** (with execute permission), the executable file to start the browser; ***create*** a file, or an entry in the associated browser profile directory, which can be used to determine when a browser session was initiated.
- Browser activities:
  - These set of activities ***create*** (or ***modify***), read (***access***) artefacts that can be used in inferring user activity. For example, navigating to a website by typing a URL in the address bar; clicking on a website shortcut; selecting an autocomplete URL suggestion in browser may ***modify*** the existing state of the browser history (e.g. the SQLite database) by creating new records based on the user's activity.
  - Activities such as accessing resources on a website, e.g. posts, uploads, downloads; would also ***create/modify*** entries in the session files, browser history, cache, etc. These activities also ***read*** (***access***) the browser cache to check that the associated content requested is in date and would ***modify*** the cache to update the content as required.
- Post-browser activities:
  - These set of activities also perform ***create*** (or ***modify***), ***read*** (***access***, ***write*** or ***execute***) operations. For example, exiting/-closing the web browser, may cause some of the contents in memory to be written to disk; logging off or powering off the device would ***create*** a corresponding entry in the event logs.
  - In some instances, ***delete*** (***remove***) operations may occur where a browser is configured to"clear" content when the browser is closed. Delete operations as part of user activities are considered out of scope for this paper.

At each point during any form of interaction with a device (pre-browser, browser and post-browser), artefacts are created allowing inference of what occurred. In any investigation where there is more than one user account on the device, to prove an action was initiated by a user, it is important to identify and highlight artefacts (including corroborating evidence) that determine whether the user being investigated was at the keyboard of the device in question within the timeframe of an incident.

3.2 Linking an Artefact to User Activity

Artefacts found on a digital device following the range of activities described in Section 3.1 can be broadly identified as system generated and user generated artefacts (Mabuto & Venter, 2012). This highlights the distinction between an artefact created by the system in the course of user activity and an artefact created as a result of direct input/interaction from a user as discussed below:

- System generated artefacts: these are artefacts created by the operating system (OS) or an application on the computer when core OS functions are performed by the OS or when a user performs tasks that require or trigger core OS services. These artefacts can be described as context artefacts because they can be used to support other artefacts recovered during an investigation and they also give context to the type of user activities that may have led to their creation. Examples include:
  - Event logs created during user login/logout activity (pre-browser; post-browser activities) could be used to infer a user had access and credentials for a device, and to connect a user to a timeframe of interest.
  - Setup or configuration artefacts created as local files or as records when an application or device driver is installed; including artefacts that are modified when a function is enabled/disabled by a user (pre-browser activity) could be used to provide context to the existence of an application (e.g. a web browser). The artefact (e.g. a Prefetch file, Registry entry) can thus be attributed to user activity (i.e. installing the web browser).
- User generated artefacts: these are artefacts created as a result of a user's direct interaction with an application. These artefacts can be described as case-specific artefacts and form the basis of inferences made about user behavior. They often create associated system generated artefacts which can be used to contextualize the activity. In the context of this paper, these refer to artefacts created during pre-browser, browser and post-browser activities. Examples include:
  - Installing a web browser (pre-browser activity) would create artefacts such as browser profile directories where the user's browser activities are stored. These artefacts can be attributed to user actions such as running an executable file to install the web browser. This can be linked to the system generated artefacts which are created as a result of this type of activity.
  - Accessing a website (browser activity) involves direct input by the user through typing in a URL or clicking on hyperlinks as discussed in Section 3.1. Artefacts indicating

visits to websites can be attributed to a user launching a web browser and providing the input (URL or click) required to access the website(s).

  - File downloads from a visited website is another example of an artefact that could be attributed to user activity. The sequence of activity and user clicks on hyperlinks, as recorded in the browser history, and leading up to the file download may be used to corroborate the existence of a local copy.
  - User activity may also create artefacts inferring the deletion/ removal of files; and the uninstallation of applications and drivers. These activities are also deemed out of scope for this paper.

3.2.1 Sources of Determination for Investigators

There are several ways an investigator can build a picture or reconstruct an event and attribute actions to a user based on the system and user generated artefacts recovered. These include but are not limited to using:

- Local Files: Created, modified, accessed dates and times can be determined when files resident on the device are analyzed.
- URLs: By deconstructing the URL, it is possible to determine what sites have been visited and what the user had typed in the address bar or search box e.g.: `search?q=statlerwaldorf\&src=typd`
- System setup or configuration logs: This infers when an application was installed; a file was created, number of times run or accessed, last time an application was run, or a file was accessed. It may also contain the path to Event logs.
- System Event logs: Logging is a way for the OS to record information about system activities that occur. This includes date and time of event; hostname/computer name of device involved; username of who was logged in to the machine when the event occurred; the program that triggered the event etc. In Windows, an identification number is assigned to each event type (Ultimate IT Security, 2014).

Attribution in the context of this paper requires an approach that links a user (or user account) to the web activity being investigated. The two-stage model proposed in this paper can be used to achieve this. This model is intended to produce case-specific and context artefacts in a social networking investigation:

i Case-specific artefacts: these can be described as artefacts that can be directly attributed to an action/activity of interest. For example, this feature "A" from the URL "Y" infers that the user clicked on the *"X"* tab on Twitter.

ii Context artefacts: these can be described as artefacts that provide an explanation of how another artefact occurred. For example, clicking on a tab or link on a web page generates a new URL however, HTTP headers or the JSON artefacts for the browser will indicate if a link was clicked as well as if there is a referrer URL. Context artefacts may also include artefacts that are expected to occur as a result of a user's activity on a social networking site. For example, the existence of photos from a social media profile in the cache could be attributed to a visit to that profile. This is discussed further in Section 5.1.2.

3.3 Two-Stage Model

Details of the proposed model are presented Figure 5.

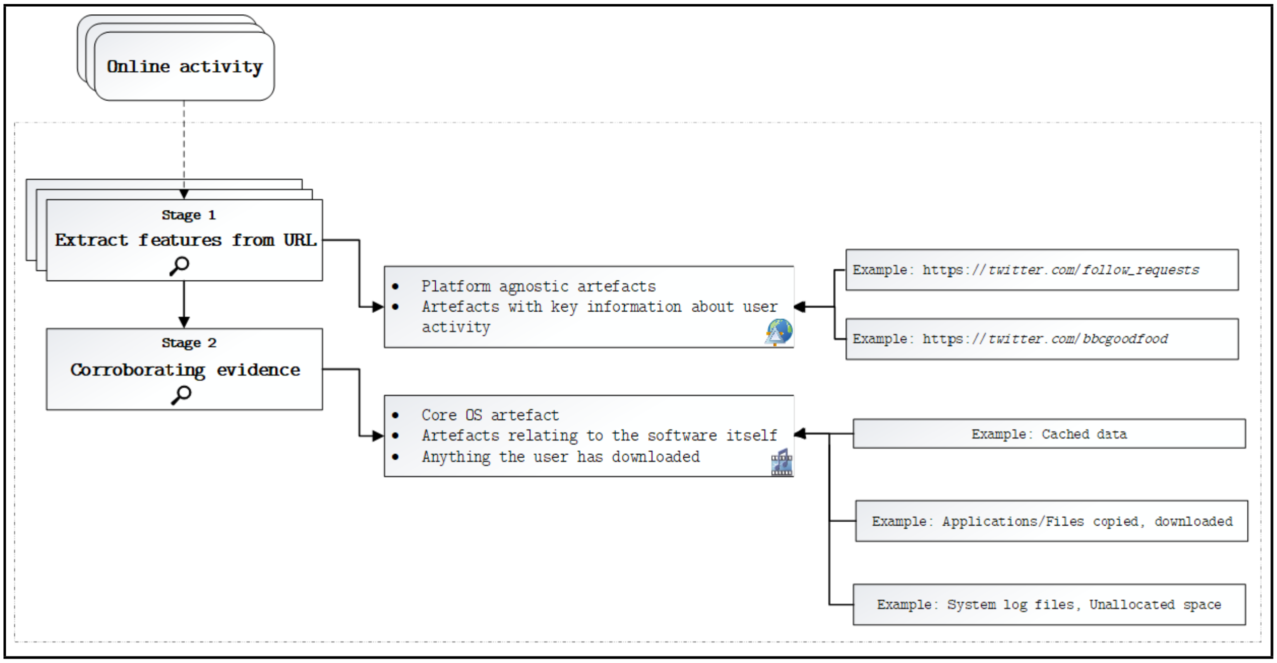

Online activity
Stage 1
Extract features from URL
Stage 2
Corroborating evidence
Platform agnostic artefacts
Artefacts with key information about user activity
Example: https://twitter.com/follow_requests
Example: https://twitter.com/bbcgoodfood
Core OS artefact
Artefacts relating to the software itself
Anything the user has downloaded
Example: Cached data
Example: Applications/Files copied, downloaded
Example: System log files, Unallocated space

Figure 5. A schematic illustration of the proposed two-stage model for the investigations of social network activity

Stage 1: URL feature extraction The first stage of this model involves the identification and recovery of URLs from disk and the extraction of features from the URLs. The URL in this instance is the main/core source of features for online activity. For example, the social network site visited, or the actions performed by the user (search, follow). It is important to note that URLs are not platform dependent, so this approach can be applied to any platform (i.e. OS or browser).

Features are extracted using a combination of RegEx and the sqlite3 module in Python. Artefacts recovered are stored in CSV files containing the dates and times of activity, the full URL, and extracted feature(s) which can be used to infer user activity or allude to the user's intent.

Stage 2: Corroborating evidence Corroborating artefacts validate each piece of evidence found during an investigation. In the context of this paper, corroborating artefacts provide both confirmation and supplementary information about the artefacts recovered during the URL feature extraction stage.

This stage of the model involves the identification and recovery of artefacts that validate what a URL feature indicates. These types of artefacts provide context to the features extracted from a URL. For example, the HTTP header information in the cache may show a URL that contains "/settings/account" was created as a result of clicking on the "account" link in the page "settings" causing the browser to respond by updating the URL and rendering the requested content. In addition to information derived from HTTP headers found in the browser cache (or unallocated space), metadata from the web page HTML could be useful in understanding the user's interaction with the social networking site.

This stage also involves the recovery of core OS artefacts that backup what has been inferred of the user's activity. For example, downloaded files associated with the recovered URLs; a local copy of uploaded data (associated with a "`www.domain-name/upload`" URL); artefacts indicating that a downloaded application was installed and run *"X"* number of times including the physical path of the application; artefacts verifying application paths.

The proposed model is useful for both the recovery of actionable intelligence and for focusing and ensuring a structured investigation. Having a "URL feature extraction" stage takes the bulk of URL artefacts and extracts meaningful information from them. This is useful because the digital forensic investigator needs a clear understanding of the URL structure in order to extract usable information from it. When artefacts from the "URL" stage have been extracted, corroborating (supplementary) artefacts are used to contextualize events and help digital evidence meet the requirement to be beyond reasonable doubt.

This paper presents work that improves on existing research on the forensic investigation of social network activities. It provides context by identifying and highlighting the importance of artefacts that corroborate or supplement the extracted feature(s). This includes data in URLs which ordinarily may be missed due to the volume of information returned by conventional digital forensics tools; data from HTTP headers and general browser artefacts.

4. RESEARCH METHODOLOGY

This section presents the data generation and analysis methodology for this paper.

4.1 Data Generation

The experiments for this paper were conducted in virtual environments running Windows 7 and Windows 10 respectively with Firefox browser installed via an executable in a network shared folder. Firefox version 61 was installed with rolling updates up to version 69.

The purpose of these experiments was to simulate real life activity on a social networking site and to create a feasible model for investigating such activities. The need to ensure the repeatability of the experiments made it necessary to use a virtual environment.

During the experimental phase, Fiddler (Telerik, 2018) was used to capture HTTP requests in order to understand how individual parts of a URL can be deconstructed; what was sent to the web server and the response returned to the client (web browser); what was cached eventually irrespective of "no-cache" options in the header etc.

Data generation for this paper involved creating a local user account in the Windows virtual machine (VM) and creating a user account on Twitter using Firefox. Normal user activity was simulated by conducting a variety of the activities listed below over the course of the experiment:

- Power on the VM and log in
- Launch the web browser

- Login to the test user account on Twitter
- Searching for users to follow
- Sending tweets
- Reply and retweet
- Viewing and sending Direct Messages (DMs)
- Send follow requests to other test user accounts
- Viewing and accepting follow requests
- Updating the test user account privacy settings
- Continuous scrolling and viewing the test user account timeline

During the experiments, all activity was logged in a contemporaneous note as a means of verifying the user activity against the results found during the analysis.

4.2 Data Analysis

The analysis was conducted in a Windows 10 desktop environment using existing digital forensics tools. This was done to identify and understand artefacts of interest in a social networking investigation and to help with the implementation of the proposed model. The data analysis covered core OS artefacts and user generated artefacts, and a multi-tool analysis technique was employed. The following tools were used:

- General examination tool:
  - WinHex (X-Ways Software Technology, AG, 2018): this was used to view the virtual disk contents and for the extraction of artefacts to be analyzed with third-party tools (listed below). It was also used for simultaneous keyword search. WinHex simultaneous search function allows a list of search terms (one per line) to be searched at once. It is important to note that a keyword search was run across the whole disk image and WinHex would have only picked up hits (data) that was visible to it; this limitation means that it would not have captured the contents of compound or encrypted files.
- Tools used for individual artefact analysis:
  - Registry Decoder (Case & Marziale, n.d.): this was used to analyze the registry hives.
  - RegRipper (Carvey, 2018): this was used to validate the results from Registry Decoder.
  - DB Browser for SQLite (DB4S Project, n.d.): used for the analysis of user browser SQLite databases in the user's Firefox profile.
  - MZCacheView (NirSoft, 2018b): used for the analysis of the Firefox cache artefacts.
  - FullEventLogView (NirSoft, 2018a): used for the analysis of Windows event logs.
  - Prefetch Forensics (Woan, 2013): used for the analysis of Prefetch artefacts.
  - Python 3 (Python.org, 2019): python scripts were used to convert the sessionstore files to JSON format and to extract features from the recovered URLs.

5. RESULTS AND DISCUSSION

This section discusses the experimental results, following the proposed two-stage model approach. These are categorized into URL feature artefacts and corroborating artefacts.

5.1 Implementation

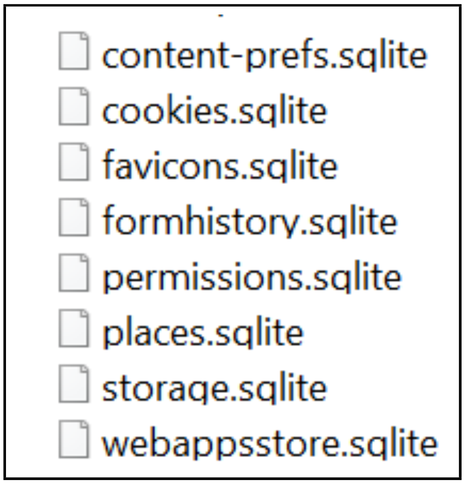
Artefacts of interest in this research include user and system generated artefacts inferring user activity on Twitter. This includes URLs generated as a result of Twitter activity (login, searching for followers, viewing followers, tweeting etc.) and system generated artefacts that give context to other artefacts and a user's activity.

5.1.1 URL Feature Artefacts

As discussed in previous sections, features from URLs can be used to give context to or infer user action. URL artefacts were recovered from the user's Firefox Profile using DB Browser for SQLite. It is important to differentiate between the History location and the Cache location as both folders share the same name. In this section, when the Firefox Profile folder is mentioned, it refers to the folder containing the browser history. The Cache is discussed in Section 5.1.2.3.

The Firefox profile can be found in: `%APPDATA%\Mozilla\Firefox\Profiles`; where `%APPDATA%` is the variable name for `%SYSTEMROOT%\Users\<username>\AppData\Roaming`.

The profile folder contains a subdirectory with a *.default* extension (e.g. *oeds8ys7.default*) which contains the SQLite database files (shown in Figure 6), session information files, other files and directories used by Firefox.

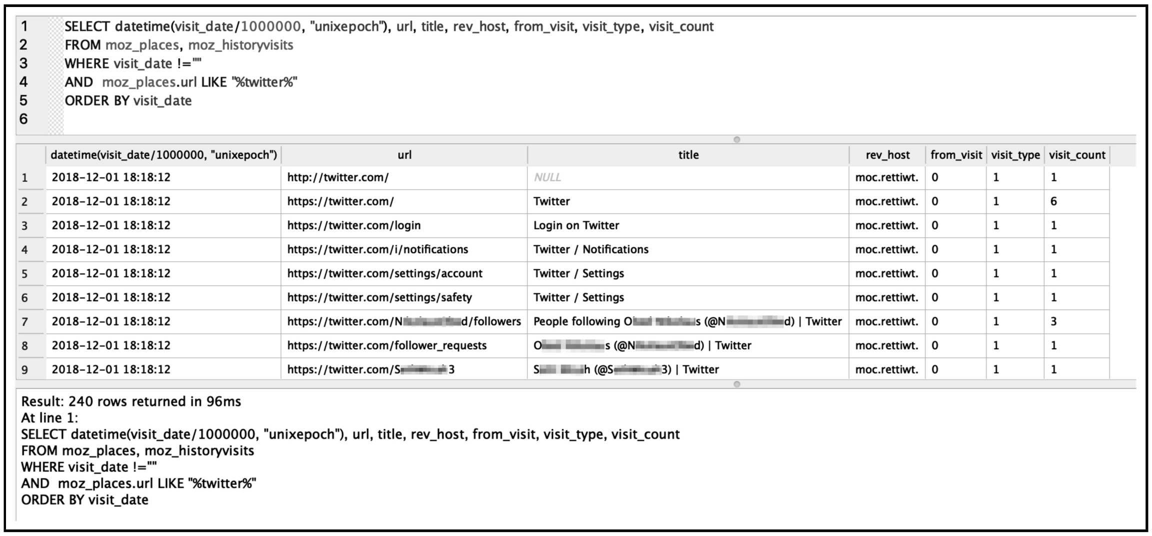


content-prefs.sqlite
cookies.sqlite
favicons.sqlite
formhistory.sqlite
permissions.sqlite
places.sqlite
storage.sqlite
webappsstore.sqlite

Figure 6. SQLite database files in the Firefox Profile folder

History The browser history is written to places.sqlite and was analyzed using DB Browser for SQLite. The query was focused on URLs indicative of accessing Twitter. The returned URLs include login, timeline, search, profile views, and follow/follower activities.

Extracts of the results are shown in Figure 7 (although the experimental user is fictitious, certain parts of the URL have been modified to remove identifying information). The extracts show standard Twitter activity URLs as displayed in the browser address bar while navigating to different pages and as recorded by Fiddler during the experiments.



```
SELECT datetime(visit_date/1000000, "unixepoch"), url, title, rev_host, from_visit, visit_type, visit_count
FROM moz_places, moz_historyvisits
WHERE visit_date !=""
AND  moz_places.url LIKE "%twitter%"
ORDER BY visit_date
```

	datetime(visit_date/1000000, "unixepoch")	url	title	rev_host	from_visit	visit_type	visit_count
1	2018-12-01 18:18:12	http://twitter.com/	NULL	moc.rettiwt.	0	1	1
2	2018-12-01 18:18:12	https://twitter.com/	Twitter	moc.rettiwt.	0	1	6
3	2018-12-01 18:18:12	https://twitter.com/login	Login on Twitter	moc.rettiwt.	0	1	1
4	2018-12-01 18:18:12	https://twitter.com/i/notifications	Twitter / Notifications	moc.rettiwt.	0	1	1
5	2018-12-01 18:18:12	https://twitter.com/settings/account	Twitter / Settings	moc.rettiwt.	0	1	1
6	2018-12-01 18:18:12	https://twitter.com/settings/safety	Twitter / Settings	moc.rettiwt.	0	1	1
7	2018-12-01 18:18:12	https://twitter.com/N[redacted]d/followers	People following O[redacted]s (@N[redacted]d) | Twitter	moc.rettiwt.	0	1	3
8	2018-12-01 18:18:12	https://twitter.com/follower_requests	O[redacted]s (@N[redacted]d) | Twitter	moc.rettiwt.	0	1	1
9	2018-12-01 18:18:12	https://twitter.com/S[redacted]3	S[redacted]h (@S[redacted]3) | Twitter	moc.rettiwt.	0	1	1

```
Result: 240 rows returned in 96ms
At line 1:
SELECT datetime(visit_date/1000000, "unixepoch"), url, title, rev_host, from_visit, visit_type, visit_count
FROM moz_places, moz_historyvisits
WHERE visit_date !=""
AND  moz_places.url LIKE "%twitter%"
ORDER BY visit_date
```

Figure 7. URLs showing user activity (recovered from places.sqlite)

The results from DB Browser for SQLite were exported and saved in *.csv* format ready for URL deconstruction.

Using the contemporaneous notes, the recovered URLs were grouped based on the user activities carried out during the experiments. Tables 1 - 3 show examples of artefacts indicative of profile/timeline view, clicking the "followers" hyperlink on the user's timeline and viewing follower requests, respectively.

Table 1. Twitter timeline URL and tab title

URL	Tab Title
`https://twitter.com/NxxxxxxxOxxx` *(User's home/timeline)*	`Oxxx Nxxxxxxx (@Nxxxxxxxxxxd) | Twitter` *(Tab title contains user's full name and (@user's handle))*

Table 2. URL generated when the "followers" link is clicked

URL	Tab Title
https://twitter.com/ NxxxxxxxOxxx/followers *(Created when "followers" is clicked)*	People following Oxxx Nxxxxxxx (@Nxxxxxxxxxxd) *(Tab title)*

Table 3. URL generated when the "followers" link is clicked to view follow requests

URL	Tab Title
https://twitter.com/ NxxxxxxxOxxx/follower_requests *("pending follower requests" is clicked.)*	Oxxx Nxxxxxxx (@Nxxxxxxxxxxd) | Twitter *(Tab title)*

During the experiments, the account security and privacy settings were updated. This includes making the account private etc. The user clicked on settings and was directed to the account page from where security and privacy settings can be modified. Tables 4 - 5 show URLs from this activity:

Table 4. URL indicative of account settings modification

URL
https://twitter.com/settings/account *(This URL takes the user to the 'Account page' from where the user account settings can be modified.)*

Table 5. URL indicative of privacy settings modification

URL
https://twitter.com/settings/safety *(When the user clicks on 'Privacy and safety' within settings, this URL is created. This page allows the user to set tweets as private, disable location tagging etc.)*

To attribute these artefacts to actions performed, it was necessary to review the HTTP request and response headers captured by Fiddler. This was done to determine if an unrelated activity could have caused the creation of these artefacts.

The next step was to break down the URLs into manageable components, the csv and re modules in Python 3 were used as seen in the example code extract below. The features including the date and time of access were written to a csv file using a version of the code illustrated below:

```
import csv
import re

# open the csv file
with open("history.csv") as hist:
    # read the csv file
    readCSV = csv.reader(hist, delimiter=',')
    for row in  csv file:
     for item in  row:
      if ‘twitter’ in item:
          pattern = your_regex_pattern
             regex_search = re.compile(pattern, re.IGNORECASE)
             matches = regexp.findall(item)

            # write regex match to a csv file
            with open("matched_patterns.csv", 'a') as mp:
             mp.write("{0}, {1}".format(row[0], ','.join(matches)) + '\n')
```

The examples below are some of the saved feature matches; they are intended to help an analyst make sense of how an event happened. For example, the user accessed Twitter on date and time, then the user navigated to this part of the page; the user searched for this other user, viewed their profile and status etc.

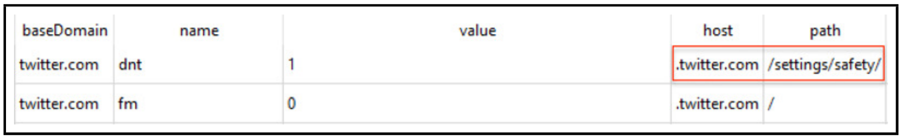
When sorted by the date and time of activity, it can be used to recreate a probable timeline of activity.

```
2016-01-17 21H:09M:49S, twitter.com, MissPiggy?ref_src=twsrc%5Egoogle
2018-12-01 18H:18M:12S, twitter.com, login
2018-12-01 18H:18M:13S, twitter.com, i, notifications
2018-12-01 18H:21M:14S, twitter.com, settings, account
2018-12-01 18H:21M:14S, twitter.com, settings, safety
2018-12-01 18H:21M:52S, twitter.com, Nxxxxxxxxxxd, followers
2018-12-01 18H:21M:52S, twitter.com, follower_requests
2018-12-01 18H:21M:52S, twitter.com, Sxxxxxxxh3
2018-12-01 18H:21M:52S, twitter.com, Nxxxxxxxxxxd
2019-11-08 11H:46M:15S, twitter.com, Pxxxxxxxxxd
2019-11-11 15H:14M:45S, twitter.com, DxxxxxKxxxx, with_replies
2019-12-24 11H:53M:39S, twitter.com, jxxxxhxxxxx, status, 544385844081987584
```

5.1.2 Corroborating Artefacts

Corroborating artefacts as discussed in Section 3.3 are artefacts that provide context to the URL artefacts recovered using the two-stage model proposed in this paper. This section discusses corroborating artefacts as they relate to the URL artefacts discussed in the preceding section.

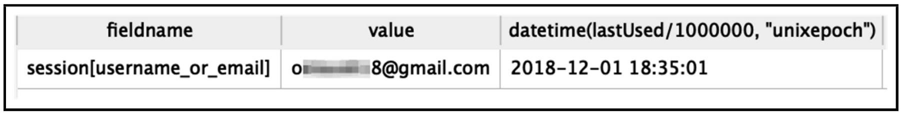
Other SQLite Database Files Cookies.sqlite records cookies set during a browsing session. It provided corroborating information for the features that were recovered from URLs in Section 5.1.1.1. Figure 8 shows a cookie set for the path “settings/safety”.



baseDomain	name	value	host	path
twitter.com	dnt	1	.twitter.com	/settings/safety/
twitter.com	fm	0	.twitter.com	/

Figure 8. Cookies set in cookies.sqlite indicates the setting/safety page was accessed on twitter.com

The cookie information can be used in conjunction with other artefacts such as login credentials shown in Figure 9, to determine and link the user account to the browsing session where the activity of interest occurred.



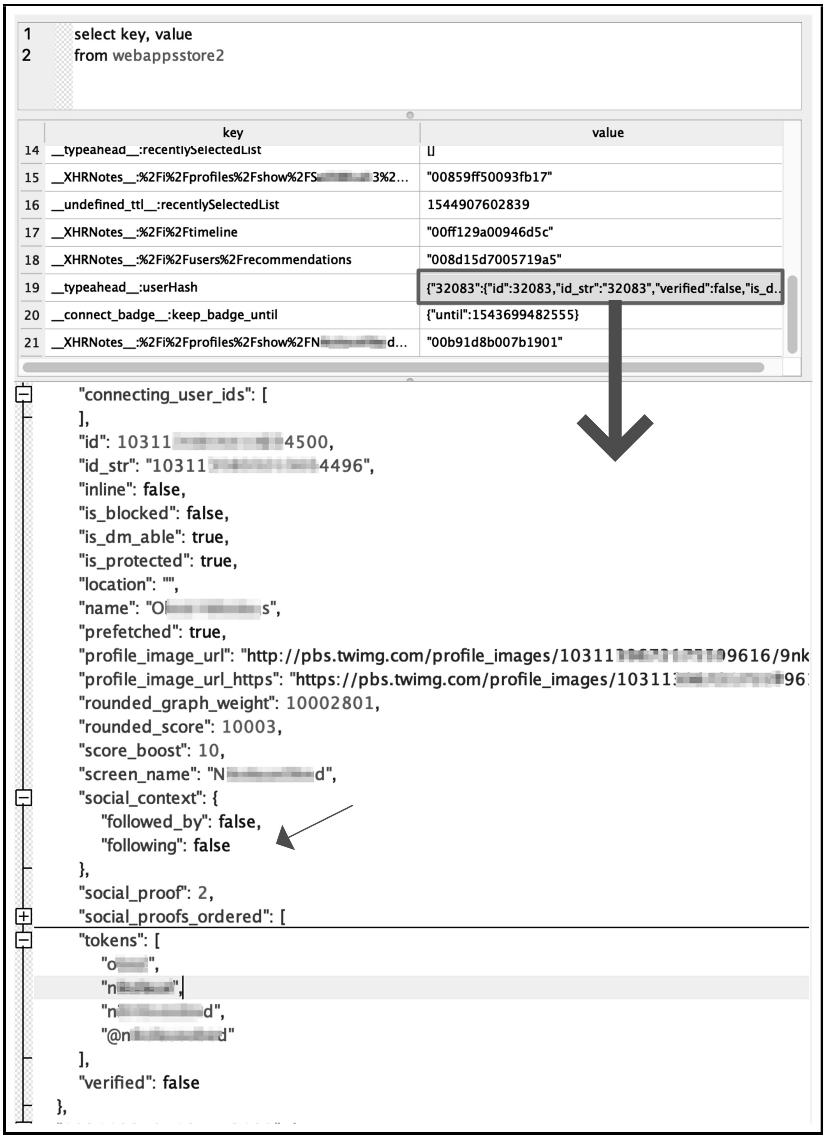
fieldname	value	datetime(lastUsed/1000000, "unixepoch")
session[username_or_email]	o[redacted]8@gmail.com	2018-12-01 18:35:01

Figure 9. Login credentials from formhistory.sqlite shows the last time the recorded email address was used

Formhistory.sqlite records entries made in form fields in Firefox. These are stored in *key:value* pairs where the fieldname is the key, and the entry typed into a text field is the value. Information recovered shows the login username for the user's Twitter account and the last time the username was used. This can be used to infer user login activity with regards to an activity and time frame of interest.

Webappsstore.sqlite stores data for websites in key:value pairs. In this instance, the value of the *__typeahead__:userHash* key contained information on the user's Twitter account including that of over 500 other Twitter users. This amount of information can be overwhelming however, using features such as *@user_handle*, *user_id*, *profile_id*, extracted from the URLs, it can be filtered down to a manageable size.

Artefacts from webappsstore such as the values of *"followed_by"* and *"following"* within *"social_context"* as seen in Figure 10, can be used to infer a user's social connection (relationship) to other users.



```sql
select key, value
from webappsstore2
```

	key	value
14	__typeahead__:recentlySelectedList	[]
15	__XHRNotes__:%2Fi%2Fprofiles%2Fshow%2FS[redacted]3%2...	"00859ff50093fb17"
16	__undefined_ttl__:recentlySelectedList	1544907602839
17	__XHRNotes__:%2Fi%2Ftimeline	"00ff129a00946d5c"
18	__XHRNotes__:%2Fi%2Fusers%2Frecommendations	"008d15d7005719a5"
19	__typeahead__:userHash	{"32083":{"id":32083,"id_str":"32083","verified":false,"is_d..
20	__connect_badge__:keep_badge_until	{"until":1543699482555}
21	__XHRNotes__:%2Fi%2Fprofiles%2Fshow%2FN[redacted]d...	"00b91d8b007b1901"

```json
    "connecting_user_ids": [
    ],
    "id": 10311[redacted]4500,
    "id_str": "10311[redacted]4496",
    "inline": false,
    "is_blocked": false,
    "is_dm_able": true,
    "is_protected": true,
    "location": "",
    "name": "O[redacted]s",
    "prefetched": true,
    "profile_image_url": "http://pbs.twimg.com/profile_images/10311[redacted]9616/9nk
    "profile_image_url_https": "https://pbs.twimg.com/profile_images/10311[redacted]96
    "rounded_graph_weight": 10002801,
    "rounded_score": 10003,
    "score_boost": 10,
    "screen_name": "N[redacted]d",
    "social_context": {
        "followed_by": false,
        "following": false
    },
    "social_proof": 2,
    "social_proofs_ordered": [
    "tokens": [
        "o[redacted]",
        "n[redacted]",
        "n[redacted]d",
        "@n[redacted]d"
    ],
    "verified": false
},
```

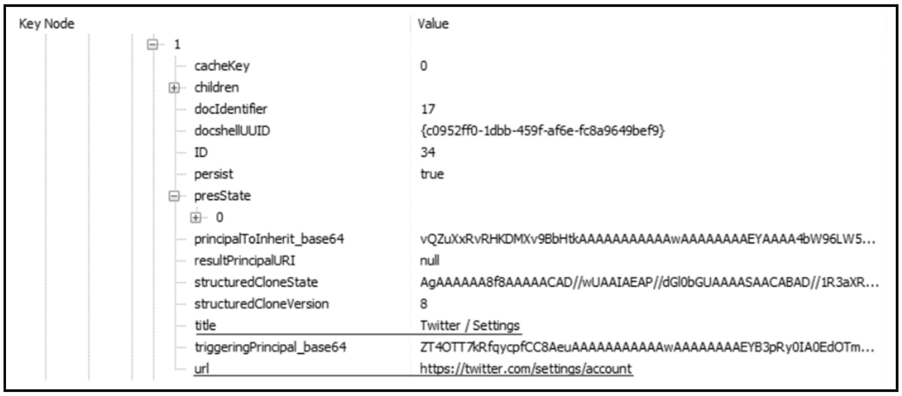
Figure 10. Extracts from webappsstore.sqlite contain JSON data that can be used to determine a user's follow/following status on Twitter

Session Information Artefacts supporting follow activity identified during the URL extraction stage were recovered from Firefox sessionstore. Sessionstore (as at the time of writing) is stored in a compressed file format (MOZLZ4/JSONLZ4) in the browser profile folder and is used by Firefox to manage the ability to restore currently open windows and tabs in the event of a crash or forced restart. It can also be used to open previous tabs on startup following a clean exit. The following files are also used to store session data: `%APPDATA%\Mozilla\Firefox\Profiles\<profile-folder>\sessionstore-backups\`

- previous.jsonlz4
- recovery.baklz4
- upgrade.jsonlz4-[datetimestamp] (session state before a browser version upgrade)

The information of interest in sessionstore include URL, page title, referrer URL; time a tab was last accessed or closed; the time a window was last accessed or closed; session start/last updated time; cookies associated with the session.

The sessionstore file was decompressed from LZ4 to JSON format, using the lz4 module in Python 3. When analyzed, it provided information on the user session where the account settings were accessed thus corroborating the activities inferred by the URLs (Section 5.1.1) as shown in Figures 11 - 14.

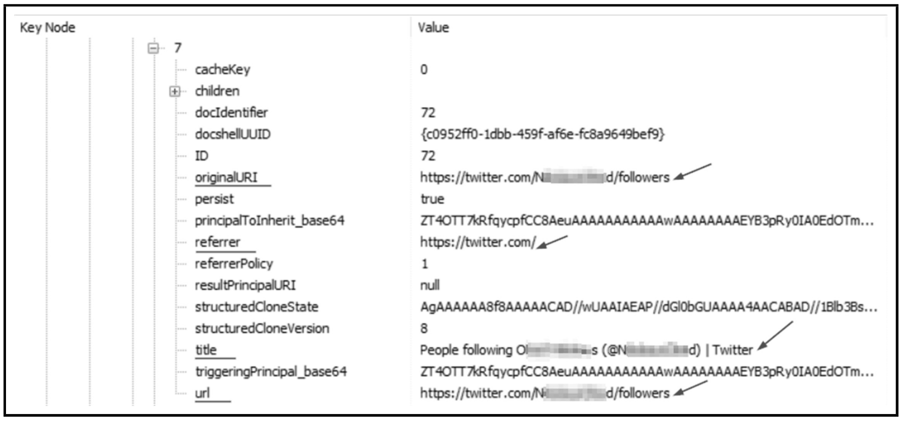


Key Node	Value
1	
cacheKey	0
children	
docIdentifier	17
docshellUUID	{c0952ff0-1dbb-459f-af6e-fc8a9649bef9}
ID	34
persist	true
presState	
0	
principalToInherit_base64	vQZuXxRvRHKDMXv9BbHtkAAAAAAAAAAAwAAAAAAAAEYAAAA4bW96LW5...
resultPrincipalURI	null
structuredCloneState	AgAAAAAA8f8AAAAACAD//wUAAIAEAP//dGl0bGUAAAASAACABAD//1R3aXR...
structuredCloneVersion	8
title	Twitter / Settings
triggeringPrincipal_base64	ZT4OTT7kRfqycpfCC8AeuAAAAAAAAAAAwAAAAAAAAEYB3pRy0IA0EdOTm...
url	https://twitter.com/settings/account

Figure 11. JSON (sessionstore) artefacts showing session where user account settings were accessed

JSON artefacts such as the extracts shown in Figure 12 are useful in identifying the session a URL is part of, and any associated (referrer) URL.

This information can be used when reconstructing how a user arrived at a specific part of the website.



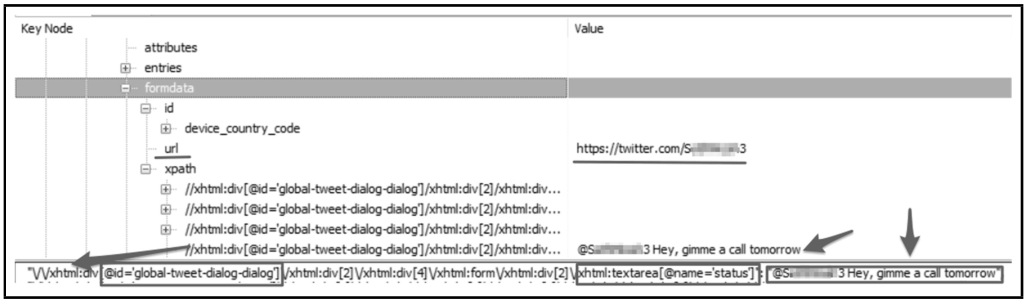
```
Key Node                                Value
        7
          cacheKey                      0
          children
          docIdentifier                 72
          docshellUUID                  {c0952ff0-1dbb-459f-af6e-fc8a9649bef9}
          ID                            72
          originalURI                   https://twitter.com/N[redacted]d/followers
          persist                       true
          principalToInherit_base64     ZT4OTT7kRfqycpfCC8AeuAAAAAAAAAAAwAAAAAAAAEYB3pRy0IA0EdOTm...
          referrer                      https://twitter.com/
          referrerPolicy                1
          resultPrincipalURI            null
          structuredCloneState          AgAAAAAA8f8AAAAACAD//wUAAIAEAP//dGl0bGUAAAA4AACABAD//1Blb3Bs...
          structuredCloneVersion        8
          title                         People following Ol[redacted]s (@N[redacted]d) | Twitter
          triggeringPrincipal_base64    ZT4OTT7kRfqycpfCC8AeuAAAAAAAAAAAwAAAAAAAAEYB3pRy0IA0EdOTm...
          url                           https://twitter.com/N[redacted]d/followers
```

Figure 12. JSON (sessionstore) artefacts showing referrer URL through which the user reached “followers”

As observed during the experiments, when another user’s profile is viewed, the URL displayed in the address bar contains the username or identifier for the visited profile.

Thus, the results can be used to infer that when a recovered URL contains a username or identifier other than that of the logged in user, it indicates a visit to the other user’s profile.

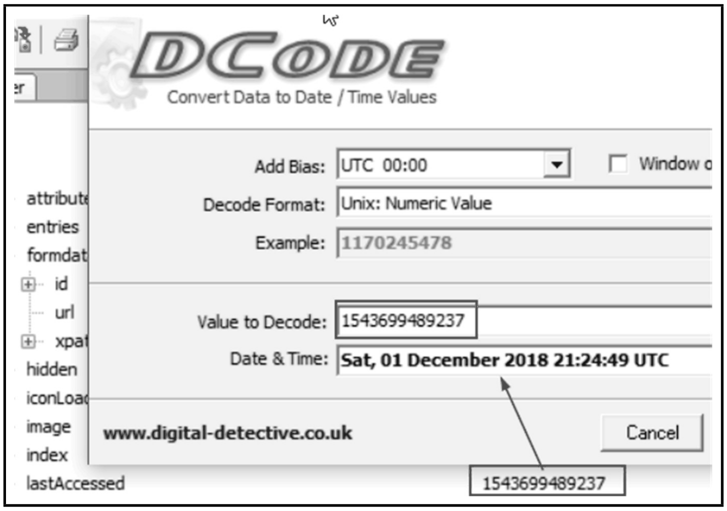
Figure 13 shows a visit to the profile of another user (URL) and the tweet sent by the experimental user.



```
Key Node                                                                   Value
      attributes
      entries
      formdata
        id
          device_country_code
        url                                                                https://twitter.com/S[redacted]3
        xpath
          //xhtml:div[@id='global-tweet-dialog-dialog']/xhtml:div[2]/xhtml:div...
          //xhtml:div[@id='global-tweet-dialog-dialog']/xhtml:div[2]/xhtml:div...
          //xhtml:div[@id='global-tweet-dialog-dialog']/xhtml:div[2]/xhtml:div...
          //xhtml:div[@id='global-tweet-dialog-dialog']/xhtml:div[2]/xhtml:div...   @S[redacted]3 Hey, gimme a call tomorrow
 "\/\/xhtml:div[@id='global-tweet-dialog-dialog']\/xhtml:div[2]\/xhtml:div[4]\/xhtml:form\/xhtml:div[2]\/xhtml:textarea[@name='status']":"@S[redacted]3 Hey, gimme a call tomorrow"
```

Figure 13. JSON (sessionstore) artefacts indicate the user visited another user’s profile and sent a tweet

In order to establish timelines, it is important to review the session within which the activity of interest occurred as shown in Figure 14. This would corroborate visit times indicated in the browser history and would also identify other activities that may have occurred within the timeframe of interest.



Add Bias: UTC 00:00

Decode Format: Unix: Numeric Value

Example: 1170245478

Value to Decode: 1543699489237

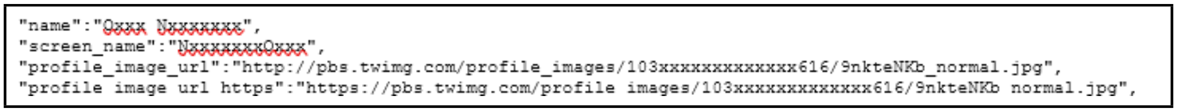
Date & Time: Sat, 01 December 2018 21:24:49 UTC

lastAccessed 1543699489237

Figure 14. JSON (sessionstore) artefacts showing the last accessed date and time for a session. Timestamp decoded by DCode

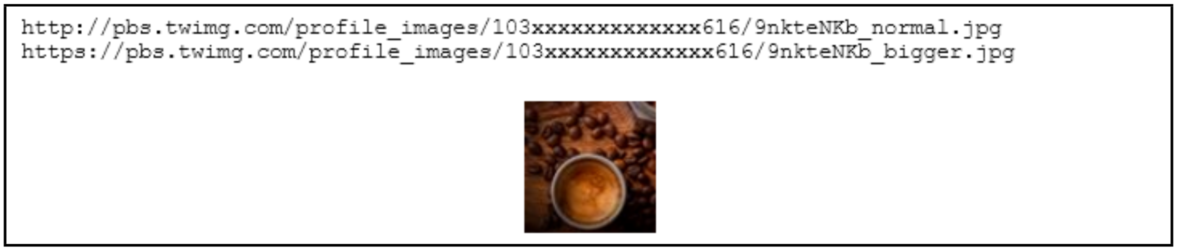
Cache The Firefox cache can be found in: `%SYSTEMROOT%\Users\<username>\AppData\Local\Mozilla\Firefox\Profile`. The profile directory contains a subdirectory with a .default extension and an identical name to that of the history profile folder (see Section 5.1.1).

During the analysis of the cache, artefacts recovered were indicative of a user's direct interaction with other users for example, profile banners and profile photo URLs and images. These URLs validated the contents recovered from webappsstore.sqlite (Figure 10) and sessionstore. Extracts are shown below in Figures 15 and 16:



```
"name":"Oxxx Nxxxxxxx",
"screen_name":"NxxxxxxxOxxx",
"profile_image_url":"http://pbs.twimg.com/profile_images/103xxxxxxxxxxxx616/9nkteNKb_normal.jpg",
"profile image url https":"https://pbs.twimg.com/profile images/103xxxxxxxxxxxx616/9nkteNKb normal.jpg",
```

Figure 15. Extract of user profile data from webappsstore.sqlite



```
http://pbs.twimg.com/profile_images/103xxxxxxxxxxxx616/9nkteNKb_normal.jpg
https://pbs.twimg.com/profile_images/103xxxxxxxxxxxx616/9nkteNKb_bigger.jpg
```

Figure 16. Extract of user profile image from the cache

It was observed that some of the *http://pbs.twimg.com* URLs were for profiles the experimental user did not interact with. This behavior is somewhat expected, depending on the browser in use, as page elements are stored locally pending a refresh when the cached content is out of date as discussed in Section 3.1.

Just as observed with the webappsstore artefacts, the URLs can be grouped into a separate category after the features extracted in Section 5.1.1.1 have been used to filter out the URLs of interest indicating social networking activity.

It is important to note that some of the URLs recovered from the cache were as a result of background processes on Twitter. For example, hashflags and hashtags.

Hashflags can be described as custom emojis that accompany a hashtag (e.g. `#StarWars` ⚔) and are used by Twitter to promote events. For example:

```
https://abs.twimg.com/hashflags/Amazon_Holiday_2018/Amazon_Holiday_2018.png
https://abs.twimg.com/hashflags/WB_LegoMovie_Emmet/WB_LegoMovie_Emmet.png
```

Hashflag URLs may be cached even though they have not been viewed or used by the user. An example of a hashflag URL recovered from the cache is shown below:

```
7109699712   68 74 74 70 73 3A 2F 2F  61 62 73 2E 74 77 69 6D   https://abs.twim
7109699728   67 2E 63 6F 6D 2F 68 61  73 68 66 6C 61 67 73 2F   g.com/hashflags/
7109699744   57 42 5F 4C 65 67 6F 4D  6F 76 69 65 5F 45 6D 6D   WB_LegoMovie_Emm
7109699760   65 74 2F 57 42 5F 4C 65  67 6F 4D 6F 76 69 65 5F   et/WB_LegoMovie_
7109699776   45 6D 6D 65 74 2E 70 6E  67 00 E5 00 00 00 00 00   Emmet.png å
```

Hashtag URLs from the "Trends for you" frame on the user's Twitter homepage were also recovered. The user also did not interact with this part of Twitter or use hashtags during the experiment. Examples of hashtag URLs recovered from the cache are:

```
https://twitter.com/hashtag/Disney?src=hash
https://twitter.com/hashtag/RoaldDahl?src=hash
https://twitter.com/hashtag/StarTrek?src=hash
https://twitter.com/hashtag/WrathOfKhan?src=hash
```

An extract of the hashtag URLs recovered from the cache is shown below:

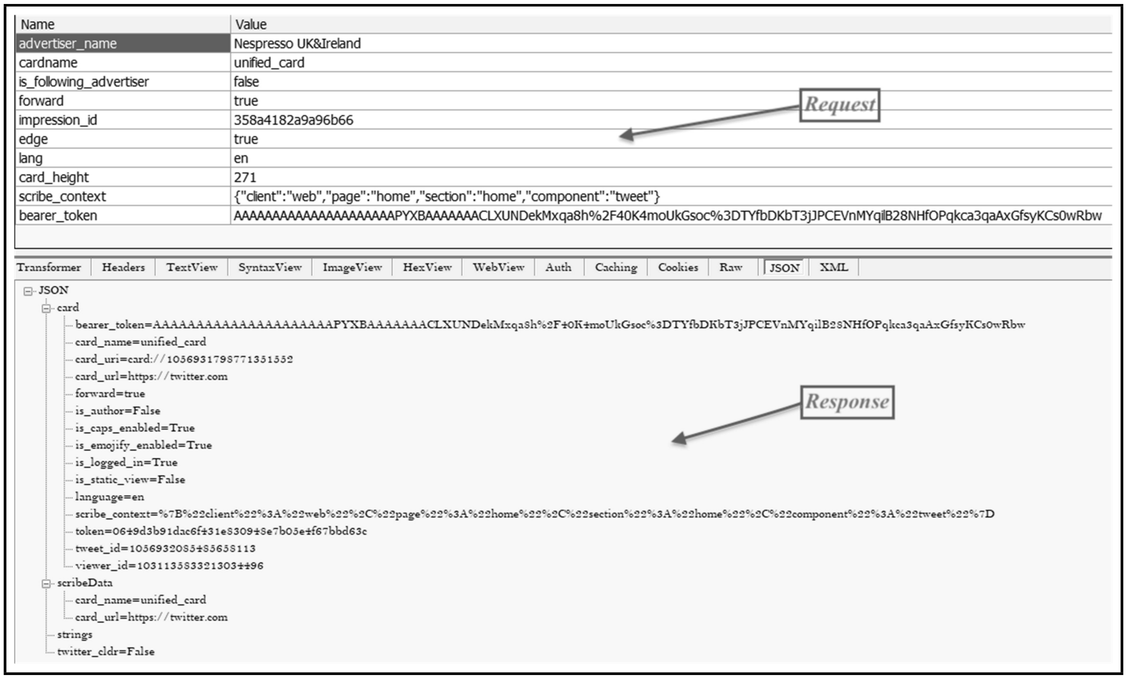
```
7096393600                            68 74 74 70 73 3A 2F 2F           https://
7096393616   74 77 69 74 74 65 72 2E  63 6F 6D 2F 68 61 73 68   twitter.com/hash
7096393632   74 61 67 2F 57 72 61 74  68 4F 66 4B 68 61 6E 3F   tag/WrathOfKhan?
7096393648   73 72 63 3D 68 61 73 68  00                        src=hash
```

Some of the recovered URLs were from suggested/promoted tweets advertised on the user's timeline. It is important to note that these ads and sponsored content can be found on disk even when a user hasn't clicked on them. In the context of this paper, they were a result of continuous scrolling on the user timeline. These tweets can be identified by the example features highlighted below:

```
https://twitter.com/ i / cards /tfw/v1/1056932085485658113?
 advertiser_name=NespressoUK&Ireland &cardname=unified_card&
 is_following_advertiser=false &forward=true&impression_id=358a
4182a9a96b66&edge=true&lang=en&card_height=271& scribe_context=
 "client":"web","page":"home","section":"home","component":"tweet" &
bearer_token=AAAAAAAAAAAAAAAAAAAAPYXBAAAAAAACLXUNDekMxqa8h/4
0K4moUkGsoc=TYfbDKbT3jJPCEVnMYqilB28NHfOPqkca3qaAxGfsyKCs0wR
bw# xdm_e=https://twitter.com &xdm_c=default4701&xdm_p=1
```

xdm is cross-domain messaging and the expected value for *xdm_e* is the base URL of the host. As shown in the URL above, the base URL is *https://twitter.com*. To determine whether a user is following an advertiser, the value for "**is_following_advertiser**" would be "true".

Figure 17 shows an extract of HTTP Request and Response headers captured through Fiddler, which was used to monitor Twitter traffic during the experiments. Fiddler was used to determine what was expected to be written to disk, what eventually was written to disk, and what was not written to disk. The ad URLs captured by Fiddler corroborate the Twitter advertising URL recovered from the user's Firefox cache. They show that the background processes on Twitter can be written to disk.



Name	Value
advertiser_name	Nespresso UK&Ireland
cardname	unified_card
is_following_advertiser	false
forward	true
impression_id	358a4182a9a96b66
edge	true
lang	en
card_height	271
scribe_context	{"client":"web","page":"home","section":"home","component":"tweet"}
bearer_token	AAAAAAAAAAAAAAAAAAAAAPYXBAAAAAAACLXUNDekMxqa8h%2F40K4moUkGsoc%3DTYfbDKbT3jJPCEVnMYqilB28NHfOPqkca3qaAxGfsyKCs0wRbw

Request

```
JSON
  card
    bearer_token=AAAAAAAAAAAAAAAAAAAAAPYXBAAAAAAACLXUNDekMxqa8h%2F40K4moUkGsoc%3DTYfbDKbT3jJPCEVnMYqilB28NHfOPqkca3qaAxGfsyKCs0wRbw
    card_name=unified_card
    card_uri=card://1056931798771351552
    card_url=https://twitter.com
    forward=true
    is_author=False
    is_caps_enabled=True
    is_emojify_enabled=True
    is_logged_in=True
    is_static_view=False
    language=en
    scribe_context=%7B%22client%22%3A%22web%22%2C%22page%22%3A%22home%22%2C%22section%22%3A%22home%22%2C%22component%22%3A%22tweet%22%7D
    token=0649d3b91dac6f431e830948e7b05e4f67bbd63c
    tweet_id=1056932085485658113
    viewer_id=1031135833213034496
  scribeData
    card_name=unified_card
    card_url=https://twitter.com
  strings
  twitter_cldr=False
```

Response

Figure 17. Extracts from Fiddler showing the HTTP Request and Response for the cached sponsored content

Due to the volume of cached data, and although they may not be used for user relationship attribution, it is important to identify and highlight social networking artefacts that are unrelated to direct user interaction. This would help an investigator/analyst focus on artefacts of immediate interest. For example, grouping sponsored content into different categories, separating them from normal user accounts.

Registry and Event Logs The artefacts presented in this section come from a range of sources (system and user generated) as discussed in Section 3.2. They aim to answer questions such as the number of user accounts on the system; the logged in user; dates and times of activity; applications installed or accessed; paths to files and applications etc.

Where there are multiple user accounts on a device, it is important to identify registered account(s) of interest, credentials, permissions and dates and time of access. These can be identified and recovered by analyzing the SAM registry hive and reviewing Windows Event logs.

The hive files (Table 6) were extracted using WinHex and information about the user account such as when the account was created, the username, the account type, application (browser) installation, and access were identified and recovered.

Table 6. Registry hives extracted with WinHex

Hive	Location
SAM SECURITY SOFTWARE SYSTEM	%SYSTEMROOT%\System32\config
NTUSER.DAT	%SYSTEMROOT%\Users\<username>
UsrClass.dat	%SYSTEMROOT%\Users\<username>\AppData\Microsoft\Windows

The SAM hive provided information about the user accounts, date and time created and last login time. The last login date and time can be used to corroborate session information recovered from sessionstore:

```
Username        : Oxxx Nxxxxxxx [1000]
Full Name       :
User Comment    :
Account Type    : Default Admin User
Account Created : Sun Sep 25 14:44:15 2016 Z
Name            :
Last Login Date : Sat Dec  1 18:12:26 2018 Z
Pwd Reset Date  : Sun Sep 25 14:44:15 2016 Z
Pwd Fail Date   : Never
Login Count     : 8
```

Extracts from the SYSTEM hive indicate the path of the Firefox installer in the network shared folder (see Section 4.1), and an indication that Firefox was executed:

```
1533971040|REG|||M... AppCompatCache - Z:\shared_installer_files\Firefox\Setup 61.0.2.exe

1533670994|REG|||M... [Program Execution] AppCompatCache - C:\Program Files\Mozilla Firefox\
firefox.exe [Executed]
```

This type of information recovered from the Registry when cross referenced with

Event logs (see Figure 19) verifies user account information and may be used to connect online identities matching the OS account username e.g. registered social network credentials.

It can also be used to attribute specific browser (social network activity) sessions to a user based on login/logoff times correlated using the *“LogonId”* as seen in Figures 18 and 19.

Other artefacts of interest were recovered from NTUSER.DAT, indicating when the browser (Firefox) was installed, the system default browser as shown in Table 7, where it was launched from and the number of times the browser was launched as shown in Table 8. For example, starting Firefox from the desktop or taskbar shortcut. This artefact could help an investigator/analyst corroborate session and social network activity.

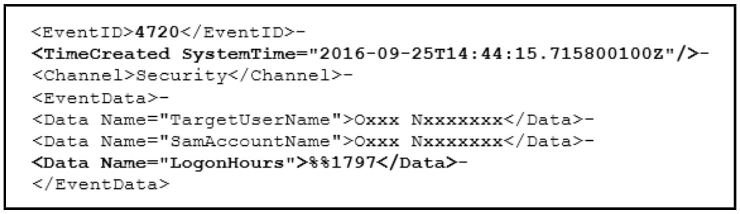
Table 7. URL indicative of account settings modification

Firefox is set as the default browser
StartMenuInternet [Sat Aug 18 18:11:33 2018 (UTC)]
NOTE: default Internet Browser client key
(default) -> Firefox-308046B0AF4A39CB

Table 8. Extract from the Registry showing that Firefox was executed once from Taskbar shortcut

Browser execution and Run count		
Datetime stamp	Path	Run count
Sat Dec 1 2018 18:19:15	9E3995AB-1F9C-4F13-B827-48B24B6C7174\TaskBar\Firefox.lnk	1

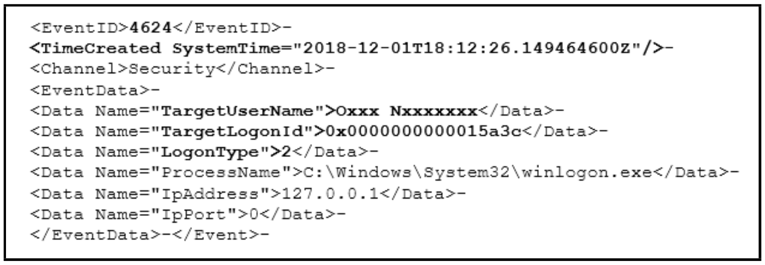
Event IDs 4720, 4624 and 4647 mean ‘A user account was created’, ‘An account was successfully logged on’ and ‘User initiated logoff’ respectively (Ultimate IT Security, 2014). There are other description fields in the Windows event logs that can provide additional information and context (e.g. Account Security ID, Domain etc.) but they are out of scope for this paper. Figures 18, 19 and 20 show extracts from the event logs indicating account creation and activity.



```
<EventID>4720</EventID>-
<TimeCreated SystemTime="2016-09-25T14:44:15.715800100Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="SamAccountName">Oxxx Nxxxxxxx</Data>-
<Data Name="LogonHours">%%1797</Data>-
</EventData>
```

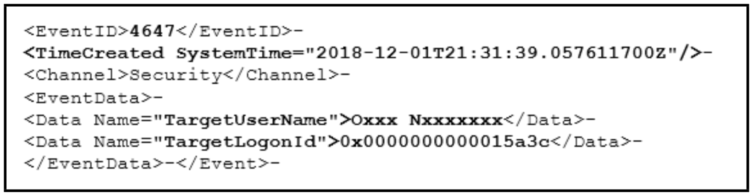
Figure 18. Account creation dates, time, username (extract from Security Event log)

The artefacts presented in this section provide additional means of contextualizing the



```
<EventID>4624</EventID>-
<TimeCreated SystemTime="2018-12-01T18:12:26.149464600Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="TargetLogonId">0x0000000000015a3c</Data>-
<Data Name="LogonType">2</Data>-
<Data Name="ProcessName">C:\Windows\System32\winlogon.exe</Data>-
<Data Name="IpAddress">127.0.0.1</Data>-
<Data Name="IpPort">0</Data>-
</EventData>-</Event>-
```

Figure 19. Login date, time, type and username (extract from Security Event log)



```
<EventID>4647</EventID>-
<TimeCreated SystemTime="2018-12-01T21:31:39.057611700Z"/>-
<Channel>Security</Channel>-
<EventData>-
<Data Name="TargetUserName">Oxxx Nxxxxxxx</Data>-
<Data Name="TargetLogonId">0x0000000000015a3c</Data>-
</EventData>-</Event>-
```

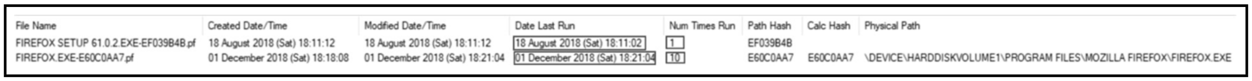
Figure 20. Logoff date, time, and username (extract from Security Event log)

URL artefacts discussed in Section 5.1.1. These artefacts can be used to corroborate the inference that the browser activity of interest as discussed in Section 5.1.2.2, is within the time frame of the user's last logon and logoff on the system.

Prefetch Prefetch is used by Windows for memory management, speeding up the boot process and application start up process. Prefetch files can be found in `%SYSTEMROOT%\Prefetch` and have a .pf extension.

Prefetch artefacts can help an investigator/analyst determine when an application was installed, when it was last run and the number of times the application has been run.

In the context of this research, Prefetch provides information related to Firefox thus, corroborating other artefacts recovered previously (see Tables 7 and 8). Figure 21 is a Prefetch extract showing when Firefox was installed, the last time it was run and the number of times it has been run since installation.



File Name	Created Date/Time	Modified Date/Time	Date Last Run	Num Times Run	Path Hash	Calc Hash	Physical Path
FIREFOX SETUP 61.0.2.EXE-EF039B4B.pf	18 August 2018 (Sat) 18:11:12	18 August 2018 (Sat) 18:11:12	18 August 2018 (Sat) 18:11:02	1	EF039B4B		
FIREFOX.EXE-E60C0AA7.pf	01 December 2018 (Sat) 18:18:08	01 December 2018 (Sat) 18:21:04	01 December 2018 (Sat) 18:21:04	10	E60C0AA7	E60C0AA7	\DEVICE\HARDDISKVOLUME1\PROGRAM FILES\MOZILLA FIREFOX\FIREFOX.EXE

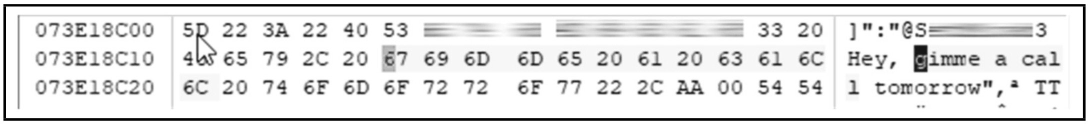
Figure 21. Firefox last run date and run count

Keyword Search As the last task of the second stage of the model, a keyword search was used to recover artefacts resident in other parts of the disk such as unallocated space, slack space and pagefile. The simultaneous search feature in WinHex was used to search across a variety of character encodings. A keyword search is useful in the identification and recovery of outlier artefacts that may be resident in unstructured part of a disk where they are not readily visible or accessible when viewed in a digital forensics tool.

It is important to use search terms or strings that would reduce the number of false positives returned. This may involve using some of the features extracted from the URLs or keys from JSON data discussed in Sections 5.1.1 and 5.1.2.2. Examples of the search terms used include but is not limited to:

- "followed_by":
- "following":
- "follower_requests"
- "profile_image_url"
- "is_following_advertiser"
- "@twitter_handle"

Figure 22 shows an example of the results returned by the keyword search. It contains the tweet sent by the user.



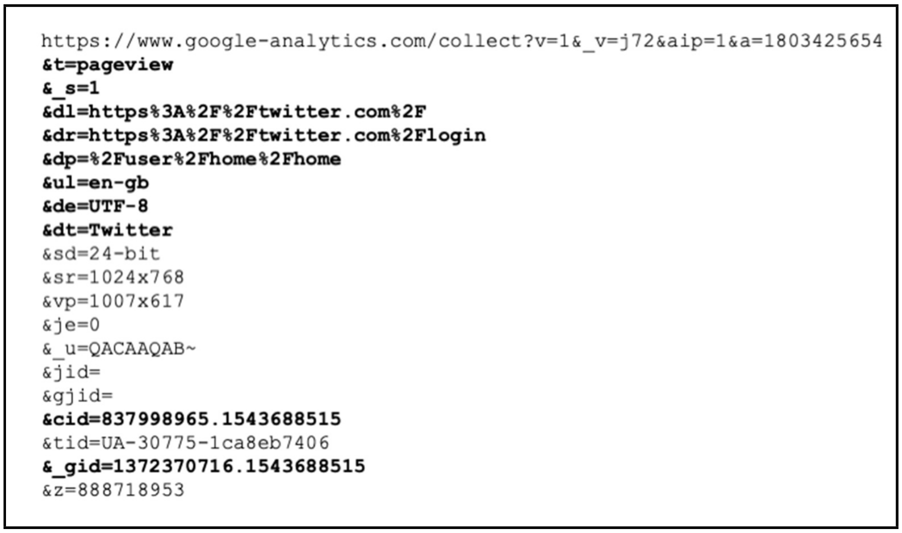
```
073E18C00   5D 22 3A 22 40 53 [redacted] [redacted] 33 20   ]":"@S[redacted]3
073E18C10   48 65 79 2C 20 67 69 6D  6D 65 20 61 20 63 61 6C   Hey, gimme a cal
073E18C20   6C 20 74 6F 6D 6F 72 72  6F 77 22 2C AA 00 54 54   l tomorrow",ª TT
```

Figure 22. Tweet fragment recovered from unallocated space

Keyword searches also highlight artefacts that may be of interest. For instance, a Google Analytics (GA) URL with access to account settings ("settings/safety") inferred in its payload data was recovered.

```
6934584320               68 74 74 70  73 3A 2F 2F 77 77 77 2E      https://www.
6934584336   67 6F 6F 67 6C 65 2D 61  6E 61 6C 79 74 69 63 73   google-analytics
6934584352   2E 63 6F 6D 2F 63 6F 6C  6C 65 63 74 3F 76 3D 31   .com/collect?v=1
6934584368   26 5F 76 3D 6A 37 32 26  61 69 70 3D 31 26 61 3D   &_v=j72&aip=1&a=
6934584384   32 39 34 37 38 31 38 35  33 26 74 3D 70 61 67 65   294781853&t=page
6934584400   76 69 65 77 26 5F 73 3D  32 26 64 6C 3D 68 74 74   view&_s=2&dl=htt
6934584416   70 73 25 33 41 25 32 46  25 32 46 74 77 69 74 74   ps%3A%2F%2Ftwitt
6934584432   65 72 2E 63 6F 6D 25 32  46 73 65 74 74 69 6E 67   er.com%2Fsetting
6934584448   73 25 32 46 73 61 66 65  74 79 26 64 72 3D 68 74   s%2Fsafety&dr=ht
6934584464   74 70 73 25 33 41 25 32  46 25 32 46 74 77 69 74   tps%3A%2F%2Ftwit
6934584480   74 65 72 2E 63 6F 6D 25  32 46 73 65 74 74 69 6E   ter.com%2Fsettin
6934584496   67 73 25 32 46 73 61 66  65 74 79 26 64 70 3D 25   gs%2Fsafety&dp=%
6934584512   32 46 75 73 65 72 25 32  46 68 6F 6D 65 25 32 46   2Fuser%2Fhome%2F
6934584528   68 6F 6D 65 26 75 6C 3D  65 6E 2D 67 62 26 64 65   home&ul=en-gb&de
6934584544   3D 55 54 46 2D 38 26 64  74 3D 54 77 69 74 74 65   =UTF-8&dt=Twitte
6934584560   72 26 73 64 3D 32 34 2D  62 69 74 26 73 72 3D 31   r&sd=24-bit&sr=1
6934584576   30 32 34 78 37 36 38 26  76 70 3D 31 30 30 37 78   024x768&vp=1007x
6934584592   36 35 34 26 6A 65 3D 30  26 5F 75 3D 53 41 43 41   654&je=0&_u=SACA
6934584608   41 51 41 42 7E 26 6A 69  64 3D 26 67 6A 69 64 3D   AQAB~&jid=&gjid=
6934584624   26 63 69 64 3D 38 33 37  39 39 38 39 36 35 2E 31   &cid=837998965.1
6934584640   35 34 33 36 38 38 35 31  35 26 74 69 64 3D 55 41   543688515&tid=UA
6934584656   2D 33 30 37 37 35 2D 36  26 5F 67 69 64 3D 31 33   -30775-6&_gid=13
6934584672   37 32 33 37 30 37 31 36  2E 31 35 34 33 36 38 38   72370716.1543688
6934584688   35 31 35 26 7A 3D 34 37  33 37 33 33 30 30 35 00   515&z=473733005
```

Figure 23 is an example of a GA URL with user login activity in the payload data.



```
https://www.google-analytics.com/collect?v=1&_v=j72&aip=1&a=1803425654
&t=pageview
&_s=1
&dl=https%3A%2F%2Ftwitter.com%2F
&dr=https%3A%2F%2Ftwitter.com%2Flogin
&dp=%2Fuser%2Fhome%2Fhome
&ul=en-gb
&de=UTF-8
&dt=Twitter
&sd=24-bit
&sr=1024x768
&vp=1007x617
&je=0
&_u=QACAAQAB~
&jid=
&gjid=
&cid=837998965.1543688515
&tid=UA-30775-1ca8eb7406
&_gid=1372370716.1543688515
&z=888718953
```

Figure 23. Twitter login activity captured in a Google Analytics URL

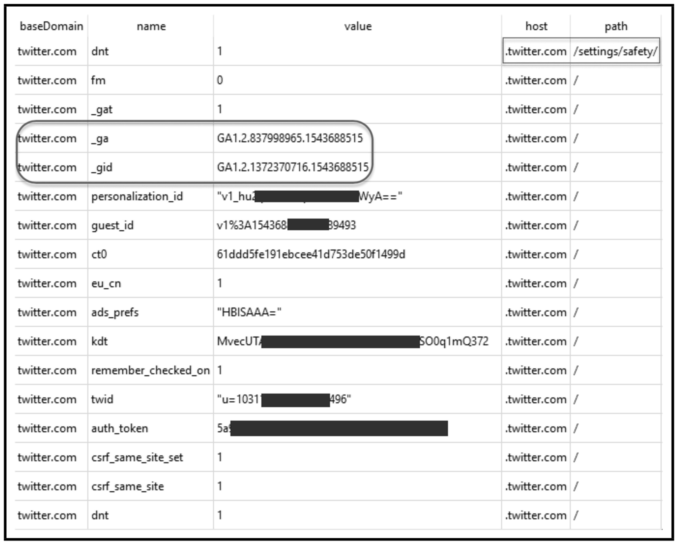
Table 9 breaks down the login URL and describes the parameters.

Other GA URLs results from the keyword search infer user access to “settings/account”, “safety/security” etc. on Twitter. When investigating social network activity, GA URLs if found on disk, may be useful in understanding user activity.

Table 9. Google Analytics URL parameter breakdown (source: Google Developers (2018))

Parameter	Value	Description
t	pageview	This is the ‘Hit’ type. Permitted values for this parameter are ‘pageview’, ‘screenview’, ‘event’, ‘transaction’, ‘item’, ‘social’, ‘exception’, ‘timing’.
_s	1	Hit sequence. The value increments by 1 with each pageview hit.
dl	https://twitter.com/	Document location URL: This parameter sends a resource (or document) location.
dr	https://twitter.com/login	Document referrer: the format for the value for this parameter is a URL (and specifies the referral source of traffic).
dp	/user/home/home	Document path (i.e. resource path) specifies the ‘path’ portion of the URL.
ul	en-gb	User language.
de	UTF-8	This specifies the character set used in encoding the page / resource (Twitter).
dt	Twitter	Document title. In this instance, “Twitter” is the page title.

Figure 24 shows the parameters from the GA URL that allude to the cookies set during the session.



baseDomain	name	value	host	path
twitter.com	dnt	1	.twitter.com	/settings/safety/
twitter.com	fm	0	.twitter.com	/
twitter.com	_gat	1	.twitter.com	/
twitter.com	_ga	GA1.2.837998965.1543688515	.twitter.com	/
twitter.com	_gid	GA1.2.1372370716.1543688515	.twitter.com	/
twitter.com	personalization_id	"v1_hu[redacted]WyA=="	.twitter.com	/
twitter.com	guest_id	v1%3A154368[redacted]9493	.twitter.com	/
twitter.com	ct0	61ddd5fe191ebcee41d753de50f1499d	.twitter.com	/
twitter.com	eu_cn	1	.twitter.com	/
twitter.com	ads_prefs	"HBISAAA="	.twitter.com	/
twitter.com	kdt	MvecUT[redacted]SO0q1mQ372	.twitter.com	/
twitter.com	remember_checked_on	1	.twitter.com	/
twitter.com	twid	"u=1031[redacted]496"	.twitter.com	/
twitter.com	auth_token	5a[redacted]	.twitter.com	/
twitter.com	csrf_same_site_set	1	.twitter.com	/
twitter.com	csrf_same_site	1	.twitter.com	/
twitter.com	dnt	1	.twitter.com	/

Figure 24. Cookies.sqlite shows cookie info found in a Google Analytics URL

Recovering and correctly interpreting artefacts such as the ones discussed in this section will enable the investigator/analyst to explain the important aspects of the recovered URL features in the context of a user's social relationships and activity. For example, explaining:

i a user's connection to a social networking site e.g. account set up and credentials

ii a user's social relationships e.g. if the user is following or being followed by another user

iii whether Twitter IDs found are as a result of direct contact, sponsored content or background processes

5.2 Section Summary

Section 5.1 discussed the experimental results, and how the proposed two-stage model can be implemented. This section addresses the relevance of artefacts recovered and the attribution of artefacts to a user.

5.2.1 Relevance of Recovered Artefacts

There are a variety of crimes that can be committed on social media (Osborne, 2010; BBC News, 2012; McGuire, 2019a, 2019b) these include activities such as, but not limited to cyberstalking, cyberbullying, inappropriate contact with another individual (including minors), fraud (digital currency or crypto-scams, identity theft), impersonation, social engineering and, dissemination of malware.

There are also other instances where social media is used to facilitate physical crime (BBC, 2010; Bowcott et al., 2011; Press Association, 2014; McGuire, 2019b). For example, burglaries, kidnapping, murder, criminal damage, drug, human or firearms trafficking.

The artefacts discussed in this paper can provide an investigator with information and context when making a determination about possible criminal activity. For example, an investigation into crimes committed on social media would seek to verify that the suspect had initiated contact with this victim. Artefacts inferring profile visits as discussed in Section 5.1.1 can add value to an investigation as it can be used to attribute a visit to the victim's profile to the suspect, and corroborating artefacts (Section 5.1.2) can be used to contextualize the suspect's activities.

Different artefacts provide insight into the possible actions of a user that created them. When taken in isolation, they do not provide the full picture of events however, corroborating each artefact in context, will give the investigator an idea of the sequence of events and the ability to exclude artefacts that are not relevant to the case. Social media artefacts as discussed in this paper can thus be used to reconstruct events, show the intent of the user, and determine user associations/relationships during an investigation.

5.2.2 Attributing Artefacts to a User

When considering the attribution of artefacts to a user, it is important to establish that:

i a link exists between a user (suspect) and a social media account

ii links exist between the user (suspect) and the activity performed by the social media account.

To determine the connection between a user and a social media account, artefacts from formhistory (Section 5.1.2.1) may be used to determine if there are login credentials for the social media account. Also, browser history and session information artefacts may be used to establish a pattern of logins and the home page for the account of interest.

Once a determination about account ownership has been made, the activity of interest associated with the social media account can be linked to the user. This can be achieved by viewing the activity in context, using the URL and corroborating artefacts as well as other types of evidence (alibi, statements) and then determining whether the user was at the keyboard at the time of an incident.

Depending on the type of activity being investigated, it is possible to use the data obtained through the two-stage model to attribute possible actions to a suspect. An example would be using images recovered from the browser cache, in conjunction with other supporting artefacts such as session information and browser history to attribute drug trafficking activity to a social media account. Using additional artefacts such as the account login credentials recovered from formhistory, strings of text advertising drugs obtained through keyword search etc., a determination can be made linking the suspect to the account and thus to the activity.

6. CONCLUSION AND FUTURE WORK

This paper has proposed a two-staged model for investigating social network activity. It has shown that a user's social networking activity can be inferred based on a range of artefacts and that it is possible based on these artefacts recovered, to identify and prioritize evidence that is pertinent to a case.

Although the syntax (see Section 2.3.1) is constant, some parts of URLs may be subject to change in structure due to improvements or changes implemented by the service provider. Such changes may include the implementation of shortened URLs or changing the location of resources on the website e.g. moving user photos from /home to /home/profile.

This model is currently focused on manual analysis with the help of digital forensics tools. Further work is required to improve this model by automating each stage of the evidence recovery process and testing its generalizability and applicability across various browsers and operating systems.

An automated framework stemming from this research is currently under development and will be used as a proof of concept to test the recoverability of artefacts and user relationship attribution.

REFERENCES

ACPO. (2012). *Good Practice Guide for Digital Evidence.* Retrieved from `http://library.college.police.uk/docs/acpo/digital-evidence-2012.pdf` (Version: 5.0)

Agency. (2015). *Five internet trolls a day convicted in UK as figures show ten-fold increase.* Retrieved from `https://www.telegraph.co.uk/news/uknews/law-and-order/11627180/Five-internet-trolls-a-day-convicted-in-UK-as-figures-show-ten-fold-increase.html`

Arshad, H., Jantan, A., & Omolara, E. (2019). Evidence collection and forensics on social networks: Research challenges and directions. *Digital Investigation*, *28*, 126–138.

BBC. (2010). *BBC News - Facebook murderer to serve at least 35 years* (No. 6/27/2010). Retrieved from `http://news.bbc.co.uk/1/hi/england/wear/8555221.stm`

BBC News. (2012). *Huge rise in social media 'crimes'.* Retrieved from `https://www.bbc.co.uk/news/uk-20851797`

Bello, M., & DiBlasio, N. (2013). *Twitter: The new face of crime.* USA Today. Retrieved from `http://www.usatoday.com/story/news/nation/2013/09/29/twitter-crime-dark-side/2875745/`

Berners-Lee, T., Masinter, L., & McCahill, M. (1994). *Uniform Resource Locators (URL) - RFC 1738.* Retrieved from `http://www.ietf.org/rfc/rfc1738.txt`

Bowcott, O., Carter, H., & Clifton, H. (2011). *Facebook riot calls earn men four-year jail terms amid sentencing outcry.* Retrieved from `https://www.theguardian.com/uk/2011/aug/16/facebook-riot-calls-men-jailed`

Cambridge University Press. (2019). *Cambridge Dictionary [Online].* Retrieved from `http://dictionary.cambridge.org`

Carvey, H. (2018). *RegRipper.* Retrieved from `https://github.com/keydet89/RegRipper2.8`

Case, A., & Marziale, L. (n.d.). *RegistryDecoder.* Retrieved from `http://www.infosecisland.com/blogview/17867-Open-Source-Registry-Decoder-11-Tool-Released.html`

Casey, E. (2002). Error, Uncertainty, and Loss in Digital Evidence. *International Journal of Digital Evidence*, *1*(2).

Casey, E. (2005). *Computer Crime and Digital Evidence: Forensic Science, Computers and the Internet. In Encyclopedia of Forensic and Legal Medicine.* Oxford: Elsevier. doi: doi:10.1016/B0-12-369399-3/00062-8

Casey, E. (2011). Digital Evidence and Computer Crime, Forensic Science, Computers and the Internet. In (Third Edition ed., chap. 1: Foundations of Digital Forensics). Elsevier Inc.

Chisum, W. J., & Turvey, B. E. (2000).

Evidence Dynamics: Locard's Exchange Principle & Crime Reconstruction. *Journal of Behavioural Profiling*, *1*(1).

Chisum, W. J., & Turvey, B. E. (2007). *A History of Crime Reconstruction. In Crime Reconstruction.* Elsevier.

Crown Prosecution Service (CPS). (2018). *Guidelines on prosecuting cases involving communications sent via social media.* Retrieved from `https://www.cps.gov.uk/legal-guidance/social-media-guidelines-prosecuting-cases-involving-communications-sent-social-media`

Cusack, B., & Son, J. (2012). Evidence Examination Tools for Social Networks. In *10th australian digital forensics conference* (pp. 33–40). SRI Security Research Institute, Edith Cowan University, Perth, Western Australia. doi: doi:10.4225/75/57b3afc1fb861

DB4S Project. (n.d.). *DB Browser for SQLite.* Retrieved 2019-02-24, from `http://sqlitebrowser.org`

Garfinkel, S. L. (2006). Forensic Feature Extraction and Cross-Drive Analysis. *Digital Investigation*, *3S*, 71–81. Retrieved from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.581.9553&rep=rep1&type=pdf`

Garfinkel, S. L. (2013). Digital media triage with bulk data analysis and bulk_extractor. *Computers & Security*, *32*, 56–72. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167404812001472` doi: doi:10.1016/J.COSE.2012.09.011

Google Developers. (2018). *Measurement Protocol Parameter Reference — Analytics Measurement Protocol — Google Developers.* Retrieved 2019-03-04, from `https://developers.google.com/analytics/devguides/collection/protocol/v1/parameters`

Haroon, S., & Carter, H. (2010). *Facebook security measures criticised after Ashleigh Hall murder.* The Guardian. Retrieved from `http://www.theguardian.com/uk/2010/mar/09/ukcrime-facebook`

Huber, M., Mulazzani, M., Leithner, M., Schrittwieser, S., Wondracek, G., & Weippl, E. (2011). Social Snapshots: Digital Forensics for Online Social Networks. In *Proceedings of the 27th annual computer security applications conference* (pp. 113–122). New York, NY, USA: ACM. doi: doi:10.1145/2076732.2076748

Jang, Y. J., & Kwak, J. (2015). Digital forensics investigation methodology applicable for social network services. *Springer Series in Multimedia Tools Appl*, *74*, 5029–5040. Retrieved from `https://link.springer.com/content/pdf/10.1007%2Fs11042-014-2061-8.pdf` doi: doi:10.1007/s11042-014-2061-8

Jonsson, P. (2011). *'Flash robs': How Twitter is being twisted for criminal gain [VIDEO].* The Christian Science Monitor. Retrieved from `http://www.csmonitor.com/USA/2011/0803/Flash-robs-How-Twitter-is-being-twisted-for-criminal-gain-VIDEO`

Keyvanpour, M., Moradi, M., & Hasanzadeh, F. (2014, 01). Digital forensics 2.0: A review on social networks forensics. *Studies in Computational Intelligence*, *555*, 17–46. doi: doi:10.1007/978-3-319-05885-6-2

Mabuto, E. K., & Venter, H. S. (2012).

User-generated digital forensic evidence in graphic design applications. In *Proceedings title: 2012 international conference on cyber security, cyber warfare and digital forensic (cybersec)* (pp. 195–200). IEEE. Retrieved from http://ieeexplore.ieee.org/document/6246107/ doi: doi:10.1109/CyberSec.2012.6246107

McGuire, M. (2019a). *Into The Web of Profit: Social Media Platforms and the Cybercrime Economy.* Bromium. Retrieved from https://www.bromium.com/wp-content/uploads/2019/02/Bromium-Web-of-Profit-Social-Platforms-Infographic.pdf

McGuire, M. (2019b). *Social Media Platforms and The Cybercrime Economy: The next chapter of Into The Web of Profit.* Bromium.

McKemmish, R. (2008). When is Digital Evidence Forensically Sound? *Advances in Digital Forensics*, *IV*, 3–15.

Moore, K. (2014). *Social media 'at least half' of calls passed to front-line police.* BBC News. Retrieved from https://www.bbc.co.uk/news/uk-27949674

Murr, M. (2007). *The admissibility vs. weight of digital evidence — Forensic Computing.* Retrieved 2019-04-14, from https://forensicblog.org/the-admissibility-vs-weight-of-digital-evidence/

NirSoft. (2018a). *FullEventLogView.* Retrieved from https://www.nirsoft.net/utils/full_event_log_view.html

NirSoft. (2018b). *MZCacheView.* Retrieved from https://www.nirsoft.net/utils/mozilla_cache_viewer.html

Oh, J., Lee, S., & Lee, S. (2011). Advanced evidence collection and analysis of web browser activity. *Digital Investigation*, *8, Supplem*(0), S62 – S70. Retrieved from http://www.sciencedirect.com/science/article/pii/S1742287611000326 doi: doi:10.1016/j.diin.2011.05.008

Osborne, B. (2010). *Twitter sees more active users, but also attracts more criminal activity.* Geek Website. Retrieved from http://www.geek.com/news/twitter-sees-more-active-users-but-also-attracts-more-criminal-activity-1130461/

Powell, A., & Haynes, C. (2019). Social Media Data in Digital Forensics Investigations. *Digital Forensic Education*, 281–303.

Press Association. (2014). *Peter Nunn jailed for abusive tweets to MP Stella Creasy.* Retrieved from https://www.theguardian.com/uk-news/2014/sep/29/peter-nunn-jailed-abusive-tweets-mp-stella-creasy

Rankin, B. (2010). *Send in the 'Twitter squad': Police forces may need dedicated to cope with rising social media crime.* Mirror News. Retrieved from http://www.mirror.co.uk/news/technology-science/technology/rocketing-crime-complaints-involving-social-1507527

Richards, J. (2007). *Sex offenders can use social sites, say police - Times Online* (Vol. 2010).

Select Committee on Communications. (2014). *CHAPTER 2: SOCIAL MEDIA AND THE LAW* . Retrieved from https://publications.parliament.uk/pa/ld201415/ldselect/ldcomuni/37/3702.htm

Shaw, U., Das, D., & Mehdi, S. P. (2016).

Social Network Forensics: Survey and Challenges. *International Journal of Computer Science and Information Security (IJCSIS)*, *14*(11), 310–316.

Sommer, P. (1999). Intrusion Detection Systems as Evidence. *Computer Networks*, *31*(23–24), 2477–2487.

Taylor, M., Haggerty, J., Gresty, D., Almond, P., & Berry, T. (2014). Forensic investigation of social networking applications. *Digital Investigation*, *11*, 9–16.

Telerik. (2018). *Fiddler - Free Web Debugging Proxy - Telerik.* Retrieved from `https://www.telerik.com/fiddler`

Ultimate IT Security. (2014). *Windows Security Log Encyclopedia.* Retrieved 2019-01-18, from `https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx`

Woan, M. (2013). *PrefetchForensics.* GitHub. Retrieved from `https://github.com/woanware/woanware.github.io/blob/master/downloads/PrefetchForensics.v.1.0.4.zip`

Wood, C. (2018). *WhatsApp photo drug dealer caught by 'groundbreaking' work.* BBC News. Retrieved from `https://www.bbc.co.uk/news/uk-wales-43711477`

X-Ways Software Technology, AG. (2018). *WinHex.* WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor. Retrieved from `https://www.x-ways.net/winhex/`

Zainudin, N. M., Merabti, M., & Llewellyn-Jones, D. (2011). Online social networks as supporting evidence: A digital forensic investigation model and its application design. In *2011 international conference on research and innovation in information systems* (pp. 1–6). doi: doi:10.1109/ICRIIS.2011.6125728