Social Snapshots: Digital Forensics for Online Social Networks

Markus Huber*† Martin Mulazzani* Manuel Leithner*

Sebastian Schrittwieser* Gilbert Wondracek‡ Edgar Weippl*

SBA Research*
{mhuber, mmulazzani, mleithner, sschrittwieser, eweippl}@sba-research.org

Vienna PhD school of informatics† Vienna University of Technology‡

ABSTRACT

Recently, academia and law enforcement alike have shown a strong demand for data that is collected from online social networks. In this work, we present a novel method for harvesting such data from social networking websites. Our approach uses a hybrid system that is based on a custom add-on for social networks in combination with a web crawling component. The datasets that our tool collects contain profile information (user data, private messages, photos, etc.) and associated meta-data (internal timestamps and unique identifiers). These *social snapshots* are significant for security research and in the field of digital forensics. We implemented a prototype for Facebook and evaluated our system on a number of human volunteers. We show the feasibility and efficiency of our approach and its advantages in contrast to traditional techniques that rely on application-specific web crawling and parsing. Furthermore, we investigate different use-cases of our tool that include consensual application and the use of sniffed authentication cookies. Finally, we contribute to the research community by publishing our implementation as an open-source project.

Keywords

online social networks, forensics, security

1. INTRODUCTION

Over the past years, Online Social Networks (OSNs) have become the largest and fastest growing websites on the Internet. OSNs, such as Facebook or LinkedIn, contain sensitive and personal data of hundreds of millions of people, and are integrated into millions of other websites [11]. Research has acknowledged the importance of these websites and recently, a number of publications have focused on security issues that are associated with OSNs. In particular, a number of empirical studies on online social networks [1, 15, 18, 17, 29] highlight challenges to the security and privacy of social network users and their data.

We found that these, and similar studies, heavily depend on datasets that are collected from the social networking websites themselves, often involving data that is harvested from user profiles. Furthermore, as social networks continue to replace traditional means of digital storage, sharing, and communication, collecting this type of data is also fundamental to the area of digital forensics. For example, data from OSNs have been used successfully by criminal investigators to find criminals and even confirm alibis in criminal cases [7, 27].

While traditional digital forensics is based on the analysis of file systems, captured network traffic or log files, new approaches for extracting data from social networks or cloud services are needed. Interestingly and contrary to our intuition, we found little academic research that aims at developing and enhancing techniques for collecting this type of data efficiently. Despite the growing importance of data from OSNs for research, current state of the art methods for data extraction seem to be mainly based on custom web crawlers. However, we found this naïve approach to have a number of shortcomings:

- High network traffic: The extraction of profile data via traditional web crawling can be regarded as costly with regard to the required network resources, as it typically incurs a large amount of HTTP traffic and causes a high number of individual network connections. Apart from inherent disadvantages, social networking websites may also choose to block network access for clients that cause high levels of traffic, thus preventing them from harvesting additional data.

- Additional or hidden data: Per definition, web crawlers can only collect data that is accessible on the target website. However, we found that social networks often publish interesting meta-information (e.g. content creation timestamps or numeric identifiers) in other data sources, for example via developer APIs.

- Maintainability: The structure and layout of websites tend to change unpredictably over time. Additionally,

ACSAC '11 Dec. 5-9, 2011, Orlando, Florida USA

the increasing use of dynamic or interpreted content (for example, JavaScript) leads to high maintenance requirements for custom web crawlers.

In this work, we introduce a novel method for data collection from social networks that aims to overcome these problems. Our approach is based on a hybrid system that uses an automated web browser in combination with an OSN third-party application. We show that our system can be used efficiently to gather "*social snapshots*", datasets which include user data and related information from the social network.

The main contributions of our work include:

- We introduce novel techniques to efficiently gather data from online social networks that may be used as criminal evidence. Our tool gathers more data than possible with today's approaches and it makes it feasible to link "online evidence" to traditional forensic artifacts found on computers using state-of-the-art tools (e.g. Encase).
- We implemented a prototype application that is aimed at Facebook, and released it under an open-source license.
- We performed an experimental evaluation involving a real-world test with volunteers and show results.

The rest of the paper is organized as follows: Section 2 introduces digital forensics followed by Section 3 which describes the design of our social snapshot framework. We evaluate the feasibility of social snapshots in Section 4. Section 5 discusses our results, Section 6 surveys related work and Section 7 concludes.

2. BACKGROUND

Digital forensics has received increasing attention in recent years as more and more crimes are commited exclusively or with the involvement of computers. Digital traces help courts and law enforcement agencies to capture valuable evidence for investigations. Existing research as well as applications in the area of digital forensics focus on filesystems [5], volatile memory [6, 16], databases [13], network traffic [8] and of course logfiles. The emergence of new online services replaces the traditional means of digital storage, sharing, and communication [4]. While traditional forensic approaches rely on the seizure of the suspect's hardware (computer, smartphone, storage media) for analysis, the emergence of online services, social networks and novel online communication methods can render this approach useless: A techno-savvy adversary might use a computer without hard disk, communicating securely with the use of encryption and storing files distributed all over the world. This would leave no traces locally for the forensic examiners to work with as soon as the computer is shut down. Another problem is the worldwide distribution of the Internet with its multitude of jurisdictions: while a court might order a company that is located within the same country to reveal information about a suspect, across borders this request may not stand.

With hundreds of millions of people sharing and communicating on social networks, they become more and more important for crime scene investigations. Traditional approaches to forensics on cloud computing and social network forensics are insufficient from an organizational as well as a technical point of view [2, 26]: the physical location of server systems is only known to the company, making seizure of hardware for examination in a forensic lab infeasible.

Often, the social network operator in question cooperates with law enforcement but in an equal number of cases they do not. Delicts that might happen solely within social networks such as cyber- stalking, mobbing or grooming, in combination with cross-border jurisdictions make it very hard to gather evidence in a forensically sound manner. A sample of social network related crimes can be found in [28]. With the increasing number of users we expect the number of social network related investigations to increase heavily in the near future. The Electronic Frontier Foundation (EFF) compiled a report [9] on U.S. law-enforcement agencies' access to social networking data. Most social networking providers have dedicated services to cater for law-enforcement requests. For example, Facebook offers two types of data: basic subscriber information ("Neoselect") and extended subscriber information ("Neoprint"). Our social snapshot application resembles a Neoprint whereas the entire subscriber content is fetched. Our social snapshot application thus offers an alternative for evidence collection, especially for non U.S. law-enforcement agencies.

3. DESIGN

Our digital forensics application enables an investigator to snapshot a given online social network account including meta-information, a method we termed "social snapshot". Meta-information such as exact timestamps are not available to the user via the user interface of the web application. A social snapshot represents the online social networking activity of a specific user such as circle of friends, exchanged messages, posted pictures etc. Due to the diversity of information available via OSNs we propose a twofold approach: an automated web-browser in combination with a custom third-party application. The social snapshot application is initialized with a user's credentials or authentication cookie. In the following, a custom third-party application is temporarily added to the target account. This application fetches the user's data, pictures, friend list, communication, and more. Information that is unavailable through the third-party application is finally gathered using traditional web-crawling techniques. By automating a standard web-browser and avoiding aggressive web-crawling we simulate the behavior of a human OSN user, thus minimizing the risk of being blocked by the social networking site. In this section, we describe the design of our approach as well as the individual components of our digital forensic framework.

3.1 Social Snapshot Framework

Figure 1 shows the core framework of our social snapshot application. **(1)** The social snapshot client is initialized by providing the target user's credentials or cookie. Our tool then starts the automated browser with the given authentication mechanism. **(2)** The automated browser adds our social snapshot application to the target user's profile and sends the shared API secret to our application server. **(3)** The social snapshot application responds with the target's contact list. **(4)** The automated web browser requests specific web pages of the user's profile and her contact list. **(5)** The received crawler data is parsed and stored. **(6)** While the automated browser requests specific web pages our so-

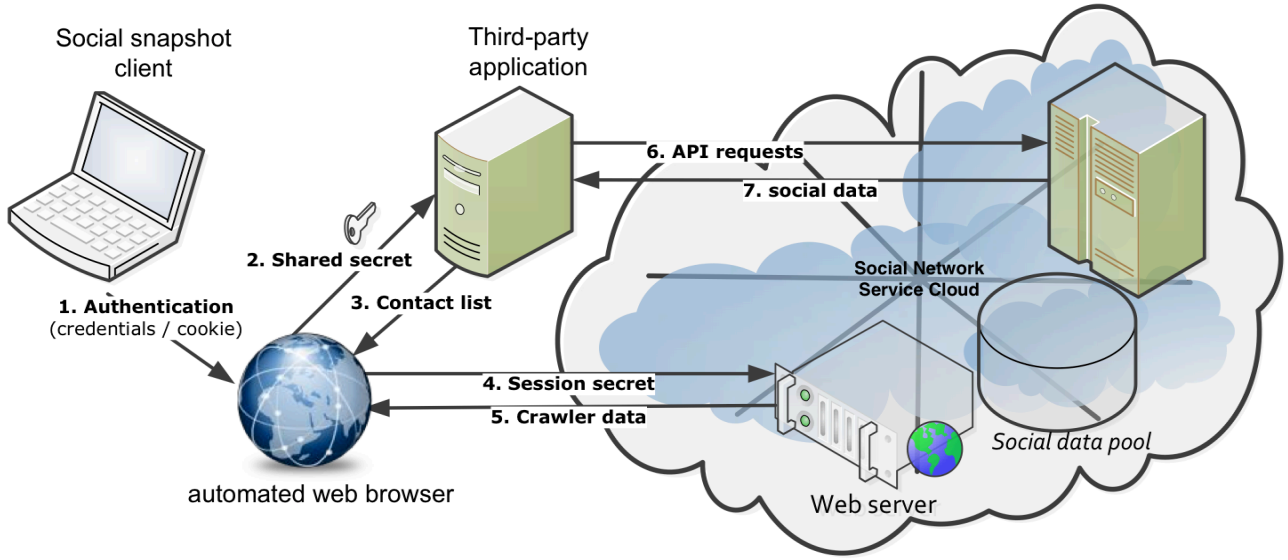

Figure 1: Collection of digital evidence through our social snapshot application.

cial snapshot application gathers personal information via the OSN API. **(7)** Finally the social data collected via the third-party application is stored on the social snapshot application server.

3.2 Authentication

In order to get access to the complete content of a target's social network account, social snapshots depend on gathering the initial authentication token. In the following, we outline three digital forensic scenarios that explain how this initial gathering of the authentication token works and that are representative for real-world use cases.

Consent. This naïve approach requires consent from the person whose social networking profiles are analyzed. A person would provide the forensic investigator temporary access to her social networking account in order to create a snapshot. This would also be the preferred method for academic studies to conduct this research in an ethically correct way and to comply with data privacy laws. We used this method for the evaluation of our proposed application as further described in Section 4.

Hijack social networking sessions. Our social snapshot application provides a module to hijack established social networking sessions. An investigator would monitor the target's network connection for valid authentication tokens, for example unencrypted WiFi connections or LANs. Once the hijack module finds a valid authentication token, the social snapshot application spawns a separate session to snapshot the target user's account.

Extraction from forensic image. Finally, physical access to the target's personal computer could be used to extract valid authentication cookies from web-browsers. Stored authentication cookies can be automatically found searching a gathered hard drive image or live analysis techniques such as Forenscope [6].

3.3 Depth of Information Collection

Starting from a target profile, a number of subsequent

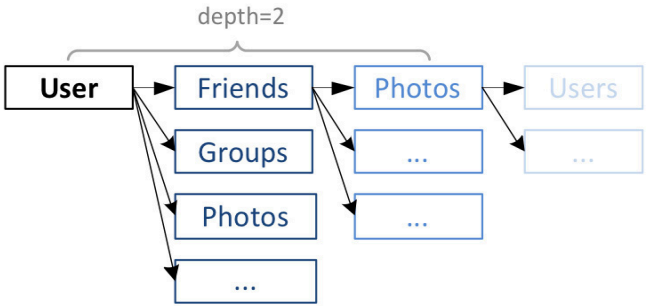

Figure 2: Example for elements fetched with social snapshot of depth=2

elements become available for crawling such as the user's friends, uploaded photos and joined groups. With these elements, again, a number of subsequent elements can be accessed. For example, the single-view page of a photo can contain comments and likes of other users, who do not necessarily have to be direct friends of the owner of the photo. Additionally, users can be tagged in photos. These are all starting points for further crawling. The same applies for groups; A group gives access to the profiles of all group members, photos with users tagged, who are potentially not members of the group, and so forth. Consequently, a social snapshot of a single user does not only obtain the user's data and data of her friends, but its depth can reach a high value. Thus, the depth of the social snapshot is an essential configuration option which controls the social snapshot's extent. Figure 2 shows an example of a social snapshot with $depth = 2$. For a given user all of her friends are first fetched, followed by the friend's photos. The single path for photos of the friend's user illustrates the magnitude of available paths and thus data. Defining a specific social snapshot depth enables us to limit the amount of fetched data. The amount of data grows exponentially with social snapshot depth.

It is important to note that the relevance of data is not the same for different elements. For example, tagged users in a photo are most likely in a closer relationship to the owner of the photo than two users that joined the same group, just because of similar interests. Therefore, the social snapshot tool prioritizes element types that suggest higher data relevance and uses them as a starting point of each iteration. The prioritization is performed on the basis of predefined priority flags in the third-party application.

3.4 Modules

Our social snapshot application consists of a number of modules, which we describe in the following. The core modules are the automated web browser and our custom third-party application as outlined in Figure 1.

Social snapshot client. The social snapshot client module initializes the data gathering process with a given user's credentials or cookies. Once started, the client first authenticates itself against the target online social network. In the following, the client automatically adds our custom third-party application with the highest possible permissions to the target's account. Information that cannot be retrieved through our third-party application is crawled and parsed by the client. Once all information has been retrieved, the client removes the third-party application and logs out of the given social networking account. The interaction with the social network as well as web-crawling is performed by the Selenium framework [22], which we describe in the following. We implemented the social snapshot client in Java and the module offers a command line interface.

Automated web browser. The browser module is responsible for the basic interaction with the target online social network. We used the Selenium testing framework [22] to automate the Mozilla Firefox browser. Selenium comes with a command line server that receives Selenium commands. Therefore, we can use the framework to script the behavior of an average user using her Firefox web-browser to surf a social networking website. We had to overcome one initial obstacle though: cookie authentication with Selenium which was not supported out-of-the-box. We finally patched the original Java source code of the command line server to be able to correctly set HTTP cookies for the cookie authentication mode.

Third-party social snapshot application. Our OSN social snapshot application is a third-party application, which sole purpose consists of gathering all possible account data through the target OSN's API. The main design goal of our third-party OSN application is performance, thus multiple program threads are used to gather information as quickly as possible. The third-party application can be configured to prioritize specific account data and to download only a predefined set of account artifacts (social snapshot depth).

Hijack. The hijack module is a network sniffer module that collects valid OSN HTTP authentication cookies from sources such as LAN or WiFi connections. We built our hijack module on the basis of Mike Perry's modified libpkt library[23], which works out of the box with LAN, unencrypted WiFi, and WEP encrypted WiFi connections. The hijack module offers a command line interface and is implemented in Python.

Digital image forensics. The digital image forensics module matches image files gathered from online social networks with their original source. The goal is to find the pristine image of a compressed picture extracted through our social snapshot application. All images are initially clustered according to their color histograms, rescaled and compressed to the target picture size, and finally matched with pattern recognition techniques. As social networks typically remove meta (EXIF) information of uploaded images this module is helpful in finding the source of collected pictures from OSNs and thus restore information such as the original image creation time, camera model etc.

Analysis. The analysis module is a parser for the results gathered with the data collection modules of our application. It parses the crawled data as well as the information collected through the OSN's API. Furthermore, the analysis module fetches additional content such as photos that are openly available by knowing the URI from online social networks. Finally, it generates a report on the social snapshot data. The analysis module can be used to generate exact timelines of communication, metadata summaries, e.g. of pictures, a weighted graph from the network of friends, or their online communication.

4. RESULTS AND EVALUATION

In this section, we describe the evaluation of our social snapshot application. Our generic social snapshot approach is applicable to the majority of today's social networking services. The sole requirement for target social networks is the availability of a developer API or the adaption of our automated browser.

For a forensic tool there are some special requirements:

- Ability to *reproduce* results,
- Create a *complete* snapshot of the account.

To make digital evidence sufficiently reliable for court it is helpful if the process of gathering the evidence can be reproduced with identical results. In dynamic Web-based applications this is not possible because data is continuously added (eg. posts by friends) or removed (eg. friends-of-friends deciding to unshare data by modifying their privacy settings). It is, however, possible to have two or more independent investigators make snapshots at a similar time. While not all artifacts will be identical one can easily compare the sets of artifacts retrieved by our tool.

It is important that all artifacts used in the case are contained in both sets and that the sets do not contain too many unique artifacts because this would suggest that the snapshots are not reliable. Similar to information retrieval research we can thus adapt the metrics of precision and recall. n independent investigators gather each a set of artifacts A_i. $\text{Precision}_j = \frac{|\bigcup_{i=0}^{n} A_i \bigcap A_j|}{|A_j|}$ $\text{Recall}_j = \frac{(|\bigcup_{i=0}^{n} A_i) \bigcap A_j|}{|\bigcup_{i=0}^{n} A_i|}$. Both can be combined to the F score.

$$F = 2 \cdot \frac{\frac{|\bigcup_{i=0}^{n} A_i \bigcap A_j|}{|A_j|} \cdot \frac{(|\bigcup_{i=0}^{n} A_i) \bigcap A_j|}{|\bigcup_{i=0}^{n} A_i|}}{\frac{|\bigcup_{i=0}^{n} A_i \bigcap A_j|}{|A_j|} + \frac{(|\bigcup_{i=0}^{n} A_i) \bigcap A_j|}{|\bigcup_{i=0}^{n} A_i|}} \quad (1)$$

4.1 Social Snapshots on Facebook

At the time of writing Facebook is the most popular online social network with a claimed user base of over 600 millions of users. Furthermore, Facebook supports third-party applications and user profiles contain a plethora of information. We thus decided to evaluate our social snapshot

tool on Facebook. Third-party applications on Facebook have access to account data via the Graph API[10]. Almost the entire account data of Facebook users and their contacts are made available through their API. Facebook solely makes sensitive contact information such as phone numbers and e-mail addresses inaccessible to third-party applications. Hence our social snapshot client crawls the contact information of Facebook profiles, while all remaining social data is fetched through a custom third-party application. In October 2010, Facebook introduced a download option[12] that enables users to export their account data. Table 1 outlines the different profile content elements gathered through our social snapshot application as compared with Facebook's download functionality. As shown in Table 1, the download functionality only offers a very limited representation of a user's online activity. For example, for a given user's friends, only their ambiguous names are made available and no information on the activity of a given user's friends is included.

Element	Download	social snapshot
Contact details	–	✓Crawler
News feed	–	✓Graph API
Checkins	–	✓Graph API
Photo Tags	–	✓Graph API
Video Tags	–	✓Graph API
Friends	name only[a]	✓Graph API
Likes	name only[a]	✓Graph API
Movies	name only[a]	✓Graph API
Music	name only[a]	✓Graph API
Books	name only[a]	✓Graph API
Groups	name only[a]	✓Graph API
Profile feed (Wall)	limited[b]	✓Graph API
Photo Albums	limited[b]	✓Graph API
Video Uploads	limited[b]	✓Graph API
Messages	limited[b]	✓Graph API

[a] No additional information available.
[b] Missing meta-information such as UIDs.

Table 1: Account information available through social snapshots compared with Facebook's download functionality.

4.2 Hardware and Software Setup

To test the functionality of our social snapshot application, we developed a third-party application for Facebook based on their PHP Graph SDK. One of the main modifications we performed on their original library was the support for multi-threaded API requests. Our third-party social snapshot application for Facebook is thus able to handle a number of predefined API requests simultaneously. The single requests are hereby pushed on a request queue with a specific priority. Hence our third-party application can be configured to, for example, fetch private messages before user comments of a Facebook group. The extent/depth of social snapshots can be further configured as a parameter for our third-party application. We deployed it on a Linux server in our university network.

Our third-party application fetches Facebook elements of a given account and stores them as separate JSON files. The separate JSON files correspond to specific requests, whereas the files are named as follows. The first part of the JSON file name is the ID of an API object while the second part specifies the requested connection detail. For instance, "123456789∼friends.request" contains all friends of the object with ID 123456789 formatted as a JSON object. In order to improve the performance of our application, we configured it not to download any videos or photos through the Graph API directly. As the third-party application collects direct links to photos, the digital image forensics module was configured to download photos during the analysis phase. Once the third-party application is finished fetching account data, it creates a tarball containing the social snapshot data.

The social snapshot client was adapted to fetch contact details of given user profiles and automatically add our third-party application to a target account. One particular challenge we had to overcome was to reliably obtain the list of friends of a given target account. Obstacles we had to cope with were the changing layout of the friend lists as well as Facebook only displaying a random subset of friends at a given time. We overcame the obstacles of creating the list of friends to be crawled, by fetching it through our third-party application and sending the profile links back to the client. Our client generates requests for every friend of the target user and sends them to the Selenium server that automates a Mozilla Firefox browser. The responses from the automated web browser module are parsed by the client and the contact information is extracted with a set of XPath queries. The client finally creates a CSV-file containing the contact information of all users. We deployed our client application in a virtual machine with a standard Ubuntu Desktop that runs our patched Selenium server. Our social snapshot analysis module implements both a parser for the fetched JSON Graph API requests as well as for fetched CSV contact details. The analysis module merges the results from the social snapshot client and the third-party application into a single database. We implemented the analysis module in Java.

We furthermore extended our digital image forensics module to automatically search a social snapshot for photo links, which it automatically downloads from the Facebook content distribution network. The hijack module did not require any Facebook specific modifications as it simply strips cookies of a given domain from a monitored network connection.

4.3 Test Subjects and Setting

We recruited human volunteers via e-mail, describing our experiment setting. The e-mail contained the experiment instructions and a briefing on how their personal information is going to be stored and analyzed. Furthermore, we briefed volunteers on the ethics of our experiment: no Facebook account data is modified, the social snapshots are stored in an encrypted filecontainer, no personal information is given to third-parties nor published. The invitation to support this first social snapshot evaluation was sent to researchers and students in computer science. Finally 25 people gave their consent to temporarily provide us access to their Facebook accounts. Volunteers temporarily reset their Facebook account credentials, which we used to create a social snapshot of their accounts. Once a social snapshot had been created, we informed our test group to reset their account password.

We configured our third-party social snapshot application for fetching an extensive account snapshot. We found that 350 simultaneous API requests lead to the best performance results in a series of indicative experiments we conducted beforehand. Our third-party application was configured to fetch the following elements recursively:

- Highest priority ($priority = 3$)
  inbox, outbox, friends, home, feed, photos, albums, statuses
- Medium priority ($priority = 2$)
  tagged, notes, posts, links, groups, videos, events
- Lowest priority ($priority = 1$)
  activities, interests, music, books, movies, television, likes

Our priority settings ensure that important information is fetched first. Account elements with highest and medium priority are fetched with $depth = 2$ while elements with the lowest priority are gathered with $depth = 1$. Thus a social snapshot of a given user includes for example, her friend's groups, tagged pictures, links etc. but no pictures, comments, etc. are downloaded from her favorite television series. These social snapshot settings imply that not only the target's account is completely fetched but also social data on the targets' friends is collected.

4.4 Results on Social Snapshot Performance

Figure 3 illustrates the time required by our third-party social snapshot application to snapshot the test accounts through the Graph API. Our third-party application required on average 12.79 minutes. Account elements of our test accounts were on average fetched with $93.1kB$ per second.

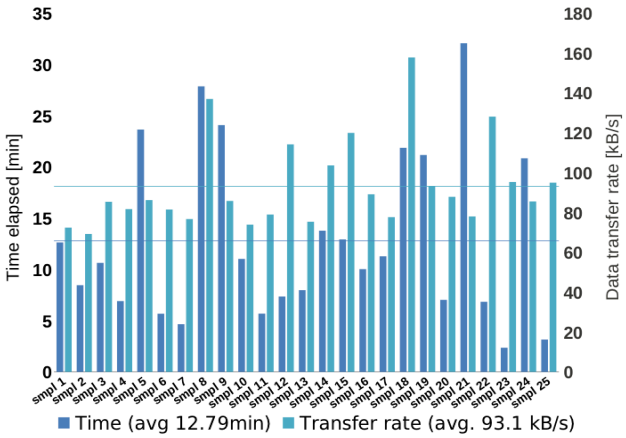

Figure 3: Required time and transfer rate of our social snapshot third-party application.

The time required for crawling contact details with our automated web browser is outlined in Figure 4. Test accounts have been crawled within 14 minutes on average. The average elapsed time per account corresponds to 3.4 seconds per user profile page.

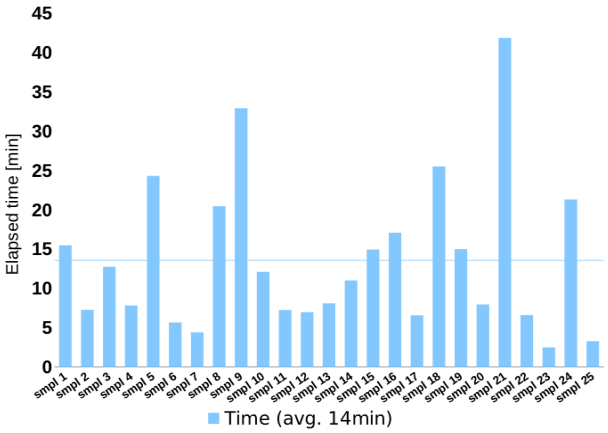

Figure 4: Time required for crawling contact details with social snapshot client and automated web browser.

4.5 Results on Social Snapshot Completeness

As illustrated in Figure 5, our third-party application found and fetched on average 9,802 Facebook account elements per test subject. The storage size of the fetched JSON files accounted to $72.29MB$ on average. Listing 1 shows an anonymized example from the fetched Facebook account elements. The example represents the basic information fetched of the user "John Doe" formatted as a JSON object. This example request also highlights that account data fetched through the Graph API provides a richer information set for further investigations. The standard web interface does not provide information if a user's account is verified nor an update time that is accurate to the nearest second with information on the used time zone.

Listing 1: Example of collected JSON element

```
{"id":"12345678","name":"John Doe",
"first_name":"John","last_name":"Doe",
"link":"http:\/\/www.facebook.com\/johndoe",
"username":"johndoe","birthday":"04\/01\/1975",
"hometown":{"id":"","name":null},
"quotes":"social snapshot your account!.\n",
"gender":"male","email":"johndoe@example.com",
"timezone":2,"locale":"en_US","verified":true,
"updated_time":"2011-05-15T13:05:19+0000"}
```

Compared to data collected via the standard web interface, our social snapshot contains a number of additional information tokens. Most notably for forensic investigation is the availability of exact creation timestamps through the Graph API. We used our image forensic module to download all unique photos in the highest available resolution from the gathered social snapshots. The downloaded photos corresponded to 3,250 files or $225.28MB$ on average per test account.

Figure 6 shows the additional contact details crawled with our social snapshot client. On average, our social snapshot client had to crawl 238 profile sites per test account. For all crawled profile pages our crawler found 22 phone numbers, 65 instant messaging accounts, as well as 162 e-mail addresses on average. We noticed that after a number of subsequent requests to user profiles of a given account, Facebook

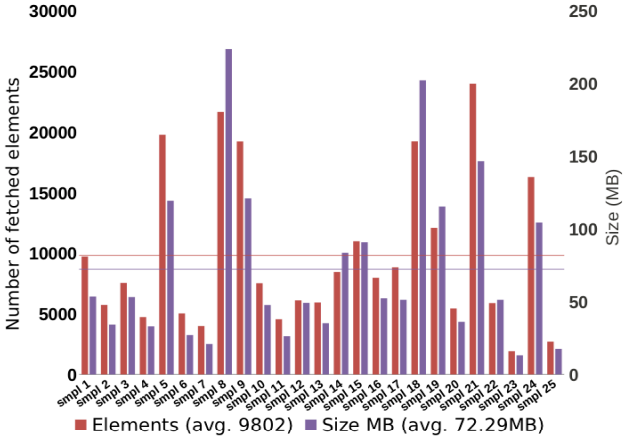

Figure 5: Account elements fetched through social snapshot third-party application.

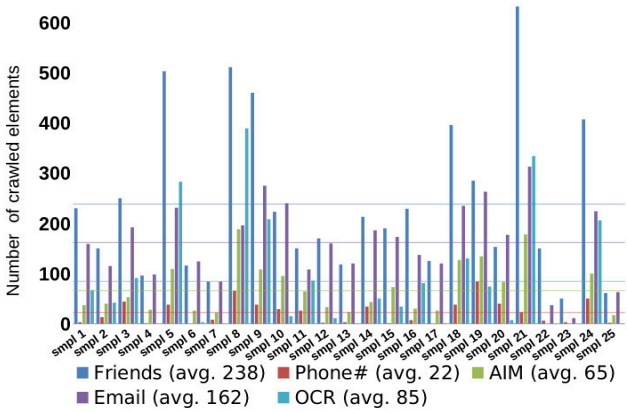

Figure 6: Contact details crawled with social snapshot client and automated web browser.

replaces textual e-mail addresses with images. This behavior was noticeable with our social snapshot client, whereas on average we fetched 85 e-mail addresses in image form (OCR in Figure 6). Due to the fact that Facebook uses e-mail addresses in image form as a web crawler protection method, we could not directly parse the fetched images.

Finally we used our analysis module to verify the integrity of the collected snapshots. We successfully verified that every entry in our fetched contact details CSV files had correspondent entries within the retrieve JSON files, as well as that no invalid responses where received through the Graph API. We furthermore implemented a mechanism for the analysis module to overcome the obstacle of parsing image e-mail addresses. By providing Facebook's e-mail image creation script the maximum possible font size of 35 instead the default of 8.7, we fetched higher resolution versions of the e-mail address pictures. We could thus rely on GNU Ocrad[14] to resolve these high resolution images into their textual representation. The idea of replacing the default font size with a larger one was first described in [25] and we could successfully verify that the described method still applies.

4.6 Indicative Cookie Authentication Experiments

We performed a number of indicative experiments to verify our cookie authentication method on Facebook. Both non-persistent as well as persistent cookie authentication is available. Persistent cookies are valid for 30 days in the case of Facebook. We successfully tested our social snapshot tool with the hijack module on a number of non-persistent users over an unencrypted test WiFi network. Furthermore, we successfully validated our social snapshot application with persistent cookies extracted from web browser profile files. In the case of one particular test setting, namely our university campus WiFi, we could observe as many as 50 valid social networking sessions within one hour.

4.7 Forensic Analysis of Social Snapshots

Collected social snapshots enable the forensic analysis of social network activity of specific users and their online peers. Since the entire content of a users' social networking account with exact timestamps is collected, timelines can be easily generated. Moreover, social snapshots offer a valuable source for further investigations. The collected e-mail addresses could for example be used to identify users on other online platforms such as photo and file storage services, while collected media data could be matched with evidence collected through traditional forensic images. Figure 7 shows an example of a generated timeline for a fictitious forensic investigation on the "Dalton Gang". The Dalton gang is suspected of having committed an aggravated bank robbery between 8:00am and 8:30am on the 13th of January 2011. All four gang members have an alibi for the specific time and said they were all on a joint getaway together with their families. Bob Dalton, the head of the gang, presents a group photo he posted on Facebook that very day. In order to validate the posting, five close friends of Bob give their consent to *social snapshot* their social networking accounts. While the posted group photo correctly shows up with the specified date in all five social snapshots, an interesting posting from Bob Dalton's wife is collected in two of the social snapshots. The posting dated one week before the robbery, timestamped with 01/06/2011 07:32:12 AM reads "Off to the beach, for our family group picture. Hehe". The investigators at this point start to suspect that the alibi picture had been taken a week beforehand to fabricate an alibi. Unaware to Bob's brother Grat Dalton, investigators social snapshot his account using the *hijack module* during his daily Internet browsing, exploiting a coffeeshop's insecure WiFi connection. Analyzing Grat's social snapshot the investigator noticed that Grat exchanged private messages with his brother Bob on the day of the robbery. The first messages with ID 00000000 sent at 3:20:32 PM reads "Grat, That was almost too easy today ... we should start thinking on how to spend all the Benjamins:-). greetings Bob". In the second message Grat replied to Bob at 6:27:12 PM: "Yeah almost too easy:-) Great idea with the group picture at the beach btw, that will cause them some serious teeth gnashing." With this further evidence on a possible false alibi, the investigators perform a house search on Bob Dalton's home. While the search does not reveal any of the stolen money,

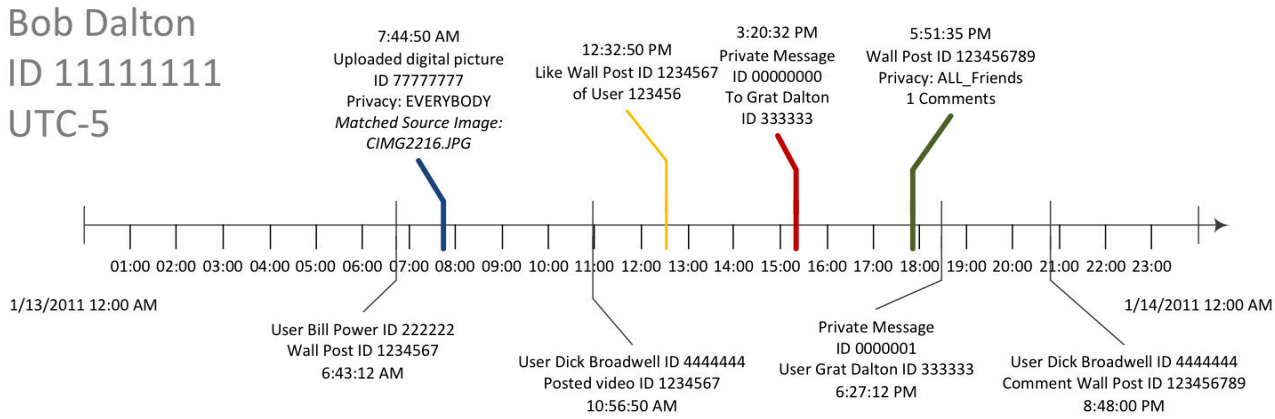

Figure 7: Example timeline created from collected social snapshot.

the personal computer of Bob Dalton is seized during the house search. Amongst digital documents and images the investigators find a valid Facebook *authentication cookie* on Bob's forensic image. The investigator creates a social snapshot of Bob's social networking account using the extracted authentication cookie. Comparing Bob's and Grat's social networking activity on the day of the robbery they find that the social snapshots accurately correlate with a F1-score of 0.84, and both accounts hold the treasonous private messages. The timeline generated from the social snapshot and outlined in Figure 7 shows Bob's online activity on the day of the bank robbery. Curious as to whether the pristine digital image of Bob's posting can be recovered the investigator runs the *digital image forensic module* to match digital images from the forensic image with the image collected through the seven independent social snapshots. The digital image forensic module reports a positive match on a digital image named "CIMG2216.JPG". The original EXIF information of image "CIMG2216.JPG" reveals that their alibi group picture had indeed been taken a week before the robbery.

4.8 Social Snapshot Open-Source Release

We release the social snapshot core framework for Facebook under a GPL v3 open source license[1]. The source code contains the social snapshot client, our third-party application, as well as the patched Selenium server. Not included in the open source release are the analysis and photo forensics modules. We furthermore decided not to release the hijack module, which could be potentially misused for malicious attacks.

5. DISCUSSION

Our evaluation required on average 9,802 API and 238 HTTP requests to successfully snapshot an entire social networking account in less than 15 minutes. In order to collect forensic evidence with traditional web-crawling more than 10,000 HTTP requests are necessary to snapshot a single test account. The generated network traffic of traditional web-crawling would have been likely detected and blocked by social networking providers. Moreover, our evaluated approach retrieved the great majority of social networking account data without the requirement of additional parsing and with exact timestamps. During the implementation of our social snapshot techniques, Facebook's web-site layout changed a number of times. Since only contact details were crawled, we could promptly adapt the parser of our client, while our third-party application did not require any changes at all. As Facebook has no review process for third-party applications we could also make our third-party application available straightforward. Third-party applications on Facebook do not even have to appear in their application directory in order to be usable.

Apart from digital forensics, social snapshots could also be used to raise user awareness. Users would run our social snapshot tool and get a report on their account data. Thus, social networking users could sight the magnitude of information that is stored with their social networking providers. We hope that this would help the average social networking user to make better informed decisions on which information they post.

Unencrypted social networking sessions enable the gathering of social snapshots for digital forensics but also pose a serious security threat. Since HTTPS is not enabled by default on today's social networking services, user sessions can easily be hijacked. Two proof-of-concept tools have been released that make session hijacking of social networking sessions available to the average user. *Firesheep* [3] has been released in October 2010 as a browser extension and at the time of writing is not functioning anymore. *Faceniff* [24] offers a point-to-click interface and supports a number of wireless network protocols. It is an Android application for hijacking social networking sessions released in June 2011. Both hijacking applications were released in order to create awareness for the problem of insecure social networking sessions. It is trivial however to couple such simple hijacking applications with our social snapshot tool. Thus, attackers could harvest complete account snapshots in an automated fashion. It has been shown [17] that the large amount of sensitive data stored in social networks could be used for large-scale spam attacks via session hijacking.

6. RELATED WORK

Numerous forensic frameworks have been proposed in re-

[1] `https://github.com/mleithner/SocialSnapshot`

cent years. However, none of them were designed specifically to extract information from social networks. To the best of our knowledge, no other publication has examined the impact of a hybrid API and crawler based approach to digital forensics in social networks.

Even though social networks are not per-se part of the cloud computing paradigm, the area of cloud forensics poses some related challenges as these service operators rely on private clouds for their infrastructure. Specifically the unknown location of data centers [26] and the difficulty to obtain access to forensic data sources without trusting a third party [2] as well as data provenance [20]. Pyflag [8], on the other hand, is a modular network forensic framework built to analyze network dumps. Among other features it is able to rebuild HTML pages from packets, allowing the examiner to view the webpages the suspect has seen even if it used AJAX or other dynamic techniques for representation. Xplico [30] is an Internet traffic decoder which can retrieve Facebook chat conversations from network dumps.

In relation to our digital image forensics module a recent approach is PhotoDNA [21], which is a program to detect known and explicitly illegal pictures based on calculated signatures. It is only available to law enforcement agencies. Similar to signature-based antivirus software, a trusted party calculates the signatures for illicit pictures such as child pornography which in turn is then compared with the signatures of pictures in webpages, data archives or pictures from forensic hard drive examinations. In [19] characteristics of embedded thumbnails are used to authenticate the source of a picture. While both approaches work similar to our module, they have not been designed or employed to compare digital images from social networks with pictures from a suspect's hard drive.

7. CONCLUDING REMARKS

Social snapshots explore novel techniques for automated collection of digital evidence from social networking services. Compared with state-of-the-art web crawling techniques our approach significantly reduces network traffic, is easier to maintain, and has access to additional and hidden information. Extensive evaluation of our techniques have shown that they are practical and effective to collect the complete information of a given social networking account reasonably fast and without detection from social networking providers. We believe that our techniques can be used in cases where no legal cooperation with social networking providers exists. In order to provide a digital evidence collection tool for modern forensic investigations of social networking activities, we release our core social snapshot framework as open source software. We will continue to extend the analysis capabilities of our forensic software and cooperate with partners on the evaluation of real-world cases.

7.1 Acknowledgments

The research was funded by COMET K1, FFG - Austrian Research Promotion Agency, by the Austrian Research Promotion Agency under grants: 820854, 824709, 825747, and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257007. Recruiting test subjects in a security lab and computer science environment was especially challenging as most people misconceived our temporary Facebook access request for a clumsy social engineering attack. Therefore, we would like to especially thank the human volunteers who supported our experiments by providing their Facebook data. The authors would also like to thank Barbara Weber and Robert Sablatnig for their advice and feedback in the creation of this paper.

8. REFERENCES

[1] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *Proceedings of the 18th international conference on World wide web*, pages 551–560. ACM, 2009.

[2] D. Birk and C. Wegener. Technical issues of forensic investigatinos in cloud computing environments. In *Systematic Approaches to Digital Forensic Engineering, 2011. SADFE 2011. Sixth International Workshop on*. IEEE.

[3] E. Butler. Firesheep. Online at `http://codebutler.com/firesheep`, oct 2010.

[4] M. Caloyannides, N. Memon, and W. Venema. Digital forensics. *Security & Privacy, IEEE*, 7(2):16–17, 2009.

[5] B. Carrier. *File system forensic analysis*. Addison-Wesley Professional, 2005.

[6] E. Chan, S. Venkataraman, F. David, A. Chaugule, and R. Campbell. Forenscope: A framework for live forensics. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 307–316. ACM, 2010.

[7] CNN. Facebook status update provides alibi. Online at `http://cnn.com/2009/CRIME/11/12/facebook.alibi/index.html`, nov 2009.

[8] M. Cohen. PyFlag-An advanced network forensic framework. *digital investigation*, 5:S112–S120, 2008.

[9] EFF. Social Media and Law Enforcement: Who Gets What Data and When? Online at https://www.eff.org/deeplinks/2011/01/social-media-and-law-enforcement-who-gets-what.

[10] Facebook. Graph API. Online at `https://developers.facebook.com/docs/reference/api/`.

[11] Facebook. Statistics of Facebook. Online at `http://www.facebook.com/press/info.php?statistics`. Accessed April 20th, 2011.

[12] Facebook. The Facebook Blog: Giving You More Control. Online at `https://blog.facebook.com/blog.php?post=434691727130`, oct 2010.

[13] K. Fowler. *SQL Server forensic analysis*. Addison-Wesley Professional, 2008.

[14] FSF. Ocrad - The GNU OCR. Online at `http://www.gnu.org/software/ocrad/`.

[15] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th annual conference on Internet measurement*, pages 35–47. ACM, 2010.

[16] B. Hay, K. Nance, and M. Bishop. Live analysis: Progress and challenges. *Security & Privacy, IEEE*, 7(2):30–37, 2009.

[17] M. Huber, M. Mulazzani, E. Weippl, G. Kitzler, and S. Goluch. Friend-in-the-middle attacks: Exploiting social networking sites for spam. *Internet Computing*, 2011.

[18] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[19] E. Kee and H. Farid. Digital image authentication from thumbnails. *Proceedings of the SPIE, Electronic Imaging, Media Forensics and Security XII*, 2010.

[20] R. Lu, X. Lin, X. Liang, and X. Shen. Secure provenance: the essential of bread and butter of data forensics in cloud computing. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 282–292. ACM, 2010.

[21] Microsoft. PhotoDNA. Online at `http://www.microsoftphotodna.com/`.

[22] OpenQA. Selenium wep application testing system. Online at `http://seleniumhq.org/`.

[23] M. Perry. CookieMonster: Cookie Hijacking. Online at `http://fscked.org/projects/cookiemonster`, aug 2008.

[24] B. Ponurkiewicz. Faceniff. Online at `http://faceniff.ponury.net/`, jun 2011.

[25] N. A. Rahman. Scraping facebook email addresses. Online at `http://www.kudanai.com/2008/10/scraping-facebook-email-addresses.html`, aug 2008.

[26] M. Taylor, J. Haggerty, D. Gresty, and D. Lamb. Forensic investigation of cloud computing systems. *Network Security*, 2011(3):4–10, 2011.

[27] The New York Criminal Law Blog. Criminal found via Facebook. Online at `http://newyorkcriminallawyersblog.com/2010/03/assault-criminal-who-was-found-via-facebook-is-back-in-ny.html`, mar 2009.

[28] The Washington Post. Facebook: a place to meet, gossip, share photos of stolen goods. Online at `http://www.washingtonpost.com/wp-dyn/content/article/2010/12/14/AR2010121407423_pf.html`, dec 2010.

[29] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A Practical Attack to De-Anonymize Social Network Users. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2010.

[30] Xplico. Xplico - Network Forensic Analysis Tool. Online at `http://www.xplico.org/`.