A Facebook Event Collector Framework for Profile Monitoring Purposes

Hugo Fonseca, Eduardo Rocha
Instituto de Telecomunicações
Campus de Santiago
3810-193 Aveiro, Portugal
Email: {diogolopes,erocha}@av.it.pt

Paulo Salvador, António Nogueira, Diogo Gomes
DETI, University of Aveiro
Instituto de Telecomunicações
Campus de Santiago
3810-193 Aveiro, Portugal
Email: {salvador,nogueira,dgomes}@ua.pt

***Abstract*—Social networks have recently emerged to become vital tools for information and content dissemination among connections. Indeed, the immense increase of the number of users of Facebook made it rise to become the largest existing social network with more than 1.2 billion active users. However, these numbers also rose the attention of hackers and attackers who aim at propagating *malware* and viruses for obtaining confidential information regarding social network users. In this manner, it is crucial that each Facebook user is able to easily access, control and analyse the information shared on the corresponding profile so that profile usage deviations can be more efficiently detected. However, despite the fact that Facebook allows an analysis of all user actions through the *Timeline Review*, this information is not comprehensively organized and there is no statistical analysis of the user generated data. In this paper, we propose a novel framework comprising a Facebook event collector, which by being provided with an authentication token for a user profile obtained through a Facebook application developed for this purpose, collects all the corresponding posted information and stores it in a relational database for *a posteriori* analysis. Through the graphical interface of the developed application, users can access all stored information in a comprehensible manner, according to the type of event, thus facilitating the analysis of the user's behaviour. By storing each event with the corresponding *timestamp*, we are able to perform an efficient and comprehensive analysis of all posted contents and compute statistical models over the obtained data. In this manner, we can create a notion of normal usage profile and detect possible deviations which may be indicative of a compromised user account.**

***Index Terms*—Social networks; Monitoring Framework; User Behaviour Modelling; Profile Hijack.**

I. Introduction

Facebook is, nowadays, considered the largest social network with more than 1.2 billion users actively disseminating different types of information and contents throughout their connections [1]. This rapid increase of the number of social network users together with the diversity of information and contents that can be posted through these networks, led to an immense increase of the number of attacks to users' profiles as well as to an enormous increase of the dissemination of different types of suspicious contents. Indeed, harvesting information on-line for creating targeted attacks or exploiting the trust relationships users establish in social networks, have become priority targets of attackers. Therefore, users are required to constantly update their profile security definitions to increase the protection of their profiles as well as of their confidential information. Moreover, the ability to detect when illicit information is posted on behalf of a user is critical for the efficient detection of possible profile hijacks. Indeed, many existing *bots* and *worms*, through an exploitation of the credentials of a user, can disseminate illicit or compromised contents through the connections of the exploited account without the user's consent. Consequently, a complete and thorough analysis of contents posted by a target profile can allow the identification of situations in which accounts become compromised and under the control of attackers without the users' knowledge. This is a critical issue since by exploiting the trust relationships a user establishes with his/hers connections, compromised contents containing, for instance, links for *phishing* attacks, can be more efficiently disseminated across the social network without raising any suspicion from the targeted users.

In this paper, we present a novel Facebook framework comprising an event collector, which by obtaining an authentication token from a specific profile through a developed Facebook application, can periodically collect and store all the information posted in the corresponding profile as well as all profile related activities. Such events include the creation and/or deletion of new connections, status and photos updates among others. On top of this event collector, we also propose a simple profile modelling module which, by interacting with a database containing all collected information for a specific profile, can present to the user several statistical informations related to the usage of the analysed profile. In this manner, the framework enables the creation and analysis of statistical models for the user behaviour on the social network. In this work, we exemplify this by computing the average occurrence of certain profile related events, over a user-defined period of time, and by creating a definition of normal usage profile. Subsequently, by performing a comparison of these values with the most recent profile usage statistics, deviations can be efficiently detected. Such deviations can be indicative of an illicit usage of the analysed profile and, consequently, allow an efficient detection of possible account hijacks. Moreover, we also present and discuss some implementation issues that were faced during this work so that the remaining research community becomes aware of these when implementing simi-

lar platforms. Finally, although, the presented event collector, platform and Facebook application are still in a testing and evaluation phase, we intend to make it available so that the remaining community on Facebook can analyse their activities and usage profile on the social network.

The remaining part of this paper is organized as follows: Section II presents some of the most relevant related work on social network data analysis; Section III presents the details of the proposed framework and its several components; Section IV presents the obtained results alongside with some development issues imposed by Facebook and, finally, Section V presents some brief conclusions about the conducted work.

II. Related Work

In [2], the authors proposed a dynamic behavioural framework for the identification of suspicious profiles in social networks. The proposed approach is based on three main indicators which comprise the balance, energy and anomaly which are all synthesized from daily user data. Balance refers to the visibility contained with a number of posted messages. The second indicator defines the energy that is consumed by a profile for increasing its visibility. Finally, the third indicator indicates the anomaly score of an observed activity-visibility pair. The authors argue that suspicious users will have unusual visibility and activity pairs which is then reflected on the anomaly score. This analysis is performed throughout a time period in which each indicator is computed and a score is then associated to each analysed profile indicating how suspicious it is. The proposed approach was applied to a set of 2000 Twitter profiles and used during a period of 30 days. By analysing the obtained results, it was concluded that suspicious profiles have a more heterogeneous behaviour than a normal one. Moreover, the authors also stated that suspicious profiles present more extreme values of balance and that these are also more likely to spend more energy than normal profiles. Finally, suspicious profiles are also likely to have an unusual activity-visibility pairing.

A recent work [3] analysed to which extent spam has invaded social networks and how do spammers operate. A large set of "honey-profiles" was created on three large social networking sites in order to collect data about spamming. The contacts and messages received by the created profiles were then logged and analysed. Anomalous behaviours of users were identified and, based on this identification, the authors developed techniques for an automatic detection of spammers. The authors proposed features such as the FF Ratio, which compares the number of friend requests with the number of friends, the URL Ratio, which computes the number of URLs on received messages, the Message Similarity, which leverages the similarity between sent messages, among many others. Based on the chosen features set, two systems for detecting spam bots were built in Facebook and Twitter which used the Random Forest algorithm, from the WEKA framework [4], for the classification. Information obtained from 1000 profiles and from 173 spam bots was used for training the classifier. The classification results presented a false positive ratio of 2% and a false negative ratio of 1% for Facebook.

In [5], a SOcial network Aided Personalized and effective spam filter (SOAP) was proposed. In this approach, each node connects to its social friends and forms a distributed overlay by using social links. Each node then uses SOAP to collect information and check spam in an autonomous manner. The authors argue that one of the novelties of the approach is the fact that it exploits social relationships and their interests to detect spam dynamically. Trace data sets obtained from Facebook were then used for evaluating the accuracy of the proposed approach and the obtained results showed that SOAP in fact improves the accuracy of Bayesian spam filters.

A framework for reducing the spread of threats between users of a social network was proposed in [6]. It comprises a Distributed Network Intrusion Detection System (DNIDS) for monitoring the propagation of threats and viruses over the monitored social network. This network was inferred from e-mail addresses obtained from the logs of e-mail servers from the Ben-Gurion University. The authors were able to slow down and to prevent the propagation of threats by cleaning the traffic from central users of the network.

A work published in 2012 outlined several new privacy leaks both in Facebook and Twitter which allow attackers to collect information for launching targeted attacks such as spam and phishing contents [7]. Moreover, the authors analysed the issues introduced by Facebook's Timeline and by several social *plugins* which users can install associate to their profiles. Finally, the authors introduced a new type of attack over Facebook known as *social network relays attack* and how can an attacker control permanently a compromised account after the first take over. Finally, solutions for all the presented security breaches are proposed but the efficiency of the proposed mechanisms is not evaluated.

A tool named *SafeGo* [8], recently presented by BitDefender. aims at bringing Internet security features to social networking. It does so by protecting users from malware threats that attempt to exploit the trust a user has with his/hers connections. Several features include *Friendly Advice*, which enables users to warn their connections of illicit content posted to their newsfeed, and *On-Post Scanning* which scans status updates in order to detect posted content with underlying threats. The developed application uses the BitDefender anti-malware and antiphishing engines for scanning URLs through an in-cloud approach which is based on a blacklist of untrusted URLs. However, unknown threats cannot be detected using this application.

The detection of Sybil attacks in large social networks was addressed in [9]. In this type of attack, an adversary creates multiple bogus identities to compromise the normal running of the targeted system. In this paper, the authors outline a defence mechanism, named SybilDefender, which leverages the network topologies to defend against these attacks. The authors state that the proposed solution is scalable and efficient to be deployed in large social networks. The proposed approach is able to detect the sybil nodes and the underlying

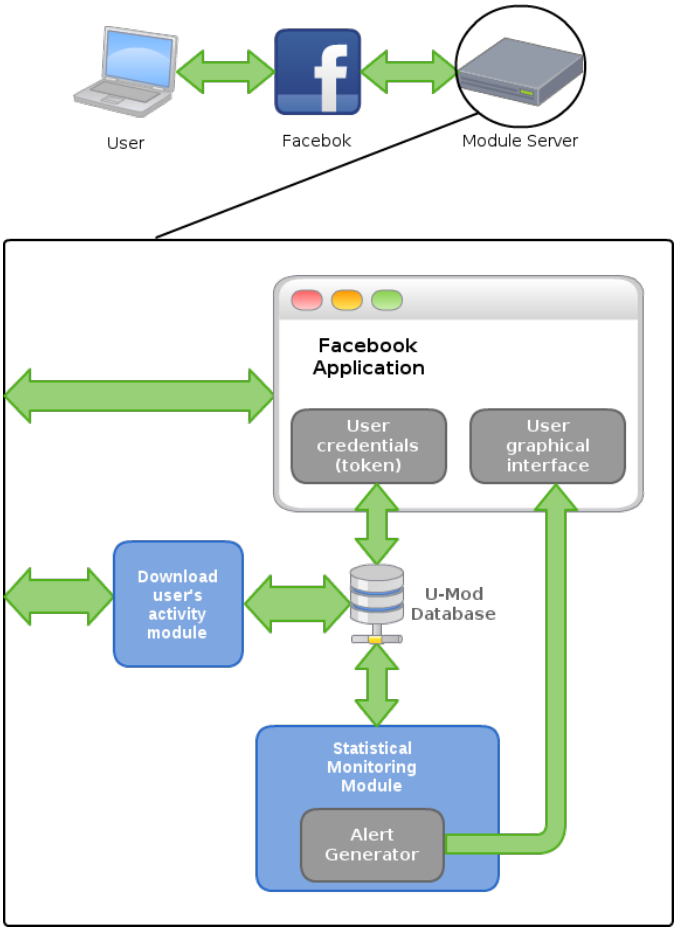

Figure 1: Framework Architecture.

community. To sustain, their claim, several experiments were run on two sets of 3 million nodes real-world social topology which showed that SybilDefender outperformed the state-of-the-art by one to two orders of magnitude in terms of both accuracy and detection speed. SybilGuard [10] and SybilLimit [11] are two decentralized algorithms that were proposed for the identification of Sybil attacks in social network topologies. However, the first mentioned work suffers from high false-negatives while the second needs to test node by node to detect if it is a sybil node or not.

III. Developed Framework

The proposed framework, depicted in figure 1, comprises a relational database, a Facebook application and different modules which download and analyse the collected data and interact with a *User Graphical Interface*. The developed database stores meta-data obtained from user's activities on Facebook. A Facebook application, based on the Graph API [12], was developed to allow the proposed framework to access the required information of a profile and store it the mentioned database. The framework then, by processing the collected data application, allows users to visualize the retrieved data and the computed models through its graphical interface. When a user logs in for the first time and accepts the required permissions, all credentials referring to the user are stored in the database, allowing the framework to access the user's activity without the need of on-line presence of the user when retrieving data. These credentials are the $user\_id$, for identifying the user of the application, and an authentication token with a valid lifetime of 2 months which implies a renewal of such token after the mentioned period of time. This renewal can be perform by a simple log-in to the developed application to enable a continuous monitoring of the profile. If such renewal does not occur and the token expires, then the developed framework is no longer able of accessing the profile information. Consequently, the information related to that user will no longer be updated. However, if a renewal occurs after the expiring of the authentication token, the system is able of collecting all data since the expiration date of the token

The data collected by the *Download User's Information Module* consists of three different information fields related to each *Object*. The definition of an *Object*, in this context, consists of an event on the analysed profile, *i.e.*, a status update on the user's wall, a like to another Facebook page among others. To each object, the application collects the corresponding *timestamp*, $user\_id$ as well as an identifier of the corresponding object on Facebook. The user's privacy is guaranteed since this identifier is used solely for checking if the related content is still available on Facebook.

Several statistical modules can be added and deployed in the proposed framework in order to assess the collected data and compute different user models. Moreover, the framework, its databases and statistical monitoring modules do not require connection to the profile in order to process data stored in the database and generate statistics or alerts since this a local procedure. Although every time the *Download User's Information Module* collects data from Facebook, or when a user accesses the application, a connection between the system and the social network needs to occur.

A. DataBase Specification

A database was developed for storing all the information obtained from Facebook so that it is accessible for future analysis either by the *Statistical Monitoring Module* or by any other module that can be deployed in the framework. The database, named *U-MODdb* (User MODelling database) was developed using MySQL [13] and PHPMyAdmin [14]. Its structure, depicted in figure 2, is composed of several tables all related to the main entry *Users* which stores several information that identify the user of the application. This includes the $user\_id$, its authentication token, the city and country of birth and of current location, the last time the user logged-in the app and the last time the framework collected information from the profile. From this entry, information related to the several components of a profile is stored into different tables. These store the *timestamps* and identifiers of the several user activities which include the groups that user belongs to (*umoddb.usGro*), the list of friends (*umoddb.usLisFri*), posted photos (*umoddb.photos*), status up-

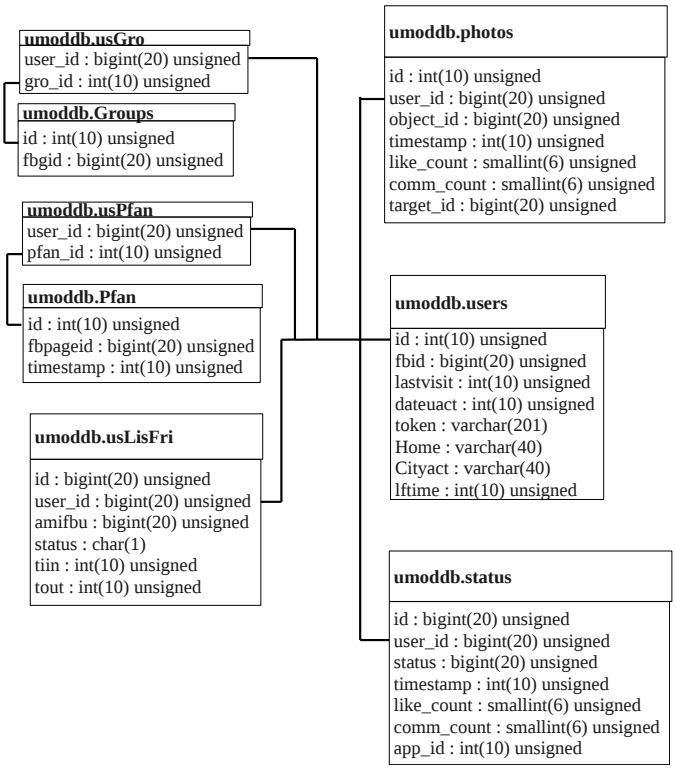

Figure 2: Framework database structure.

loads (*umoddb.status*) and the pages that a user checked with like button (*umoddb.usPFan*). For each entry, the Facebook identifier is stored as well as an unique identifier which allows the unequivocal identification of the object within the created database. In this manner, the consistency of the stored data is guaranteed.

To be able to store information related to users that subscribed the developed application and of the corresponding friends who can, at any time, also subscribe the application and for avoiding data replication and deletion, a simple user identifier is added to the database. Therefore, if a friend from a subscribing user installs the application, he already has an ID as an user within the database and does not require an additional identifier to be in a friend relation.

User authentication is performed using the Facebook log-in engine. However, if user advanced options are required for concrete types of users, such as administration, these can be defined manually but an option is inserted in the graphical interface for smoothing the process

B. Detecting Profile Usages Deviations

As can be seen in figure 1, the proposed framework includes a *Statistical Monitoring Module* which is able to analyse and process the information stored per user. In this manner, by analysing the collected *timestamps* of the different events, it is possible to define a normal usage profile. An alert module can then be created for issuing alerts to the user to situations where current usage profile differs significantly from what is defined as the normal usage. The implementation of this module requires setting boundaries that can be distinct for different types of actions (comments, likes and other) for the same user. Since different users behave differently, these boundaries must adapt to the usage profile of each subscribed user. There are several possibilities for an alert to be sent to user, either as he signs in the application or by sending an alert straight to the Facebook profile or even to the e-mail address associated to the profile.

By defining a set of events $E_t^i = \{e_t^i, t = t_1, ..., t_2\}$ as the set of events of type $i \in \{\mathcal{S}, \mathcal{L}, \mathcal{C}, \mathcal{P}\}$, in which each element denotes status updates, likes, comments, photos respectively during a time-window of analysis of width $t = t_2 - t_1$, we can compute the average of events of type i within the defined period of analysis as:

$$\overline{E_t^i} = \frac{\sum_{t=t_1}^{t_2} e_t^i}{T_{e_i}} \tag{1}$$

in which T_{e_i} denotes the number of time units (years, months, days,...) considered for analysis within the width of window of analysis defined by $t = t_2 - t_1$. By defining a threshold δ

$$\delta = \alpha.\overline{E_t^i}, \alpha \in \mathbb{R}^+ \tag{2}$$

as a function of the previously defined average, an alarm will be triggered if the average number of events, computed according to Equation 1, within the time period defined by $t' = t'_2 - t'_1 : t'_2 \neq t_2, t'_1 \neq t_1$ exceeds the value computed as:

$$E_{t'}^i > \overline{E_t^i} + \delta \tag{3}$$

In this manner, we are able to compute and compare the usage profile during different periods of analysis and to issue alarms in case of significant deviations. This threshold can then be dynamically redefined in order to avoid false positives.

C. Future Analysis Scenarios

The proposed framework allows the insertion of additional statistical modules which, by analysing the data stored in the database, can create and evaluate user behaviour models. Therefore, we envision the proposal of user models based on multi-scale analysis and modelling which was previously applied to traffic analysis and classification. Indeed, by using concepts that exploit different scales of analysis [15], we intend to extend these classification approaches and analysis in order to create a usage profile based on different scales of analysis to depict the several frequency components that are created by a user profile on social networks such as Facebook.

IV. Results

In this section we present the results obtained with the developed framework. We start by discussing some restrictions imposed by Facebook which prevent a more complete and thorough analysis of target profiles. Subsequently, since the framework and the Facebook application developed for data collection purposes are still not publicly available, we present

the results obtained with data extracted from the profiles of the authors of this work.

A. Development Issues

There are several issues associated to deploying such a framework and obtaining accurate and complete profile information from Facebook. Indeed, some profile related information is not obtainable even with an authentication token and permission to access all profile information. To begin with, although the social network allows the retrieving the list of friend requests, there is no possibility of obtaining the updated state of that request directly from it. Therefore, the only manner of detecting the creation of novel connections consists in checking if the user that send the request is in the current list of friends of the target user. In this case, it can be assumed that the request has been accepted otherwise the request might have been rejected or ignored. Another issue consists in obtaining the exact *timestamp* associated to the establishment of a connection between two different users since this information cannot be obtained from Facebook. Therefore, the application registers the system time on which the relation has been detected for the first time which implies that the degree of accuracy depends on the difference between the two performed time measurements for which the system checks checks for establishment of new user's friend relations. Two variables, *tiin* and *tout*, are stored in the database and account, respectively, for the *timestamp* in which a new friend is added to a user's friend list while the second accounts for the *timestamp* in which a connection was removed from the list of friends.

Facebook does not allow an application either to check which posts the user deleted, neither activities that have been marked with option "hidden from timeline" or "only me" (who can see the activity). Moreover, regarding to the applications subscribed by a user, Facebook is very restrictive and does not allow to know which applications an user has installed directly. However, it is possible to check if a user has accepted a specific application. Querying Facebook for each application is a very long process due to the amount of existing applications and a possible way to achieve this goal is to check the user's status messages metadata and get the field app_id. If this value exists, then the actual status has been published by an application and we can conclude that the user has subscribed on the corresponding profile.

B. Analysis of User Behaviour

As mentioned in section III, by requiring an authentication token from a target user, the framework is able to collect the *timestamps* of several events on the corresponding profile. For this work, the considered events were status updates, posted photos and *likes* to other Facebook pages. Using the approach discussed in section IV-A and by analysing the timestamps obtained for the posts shared on the target user's profile, for two different time-windows of analysis, we are able to infer, analyse and compare the behaviour of the user and detect possible deviations from what can be considered as a normal usage profile.

In figure 3, we represent the average number of posts obtained for two different periods of analysis which, in this case, are defined as the last 7 days and all previous days of current year. We define posts, in this context, as any of the events mentioned in the previous paragraph. However, this definition can be extended, or restricted, in order to include, or exclude, only one type of events, *e.g.* only status updates. We compute the average number of posts during all days of the current year and compare these values with the ones obtained for last week. In this manner, we can detect periods, in the considered windows of analysis, during which the user is not active and not posting while we can also detect periods of time in which the analysed user is more active in the social network by actively sharing contents with his/hers connections. Indeed, as shown in figure 3 and for the two considered analysis windows, the average behaviour of the user during the last 7 days follows the same pattern defined by the usage profile computed over the all days of the current year. Indeed, periods of low activity coincide in the two analysed time-windows while periods of a more intensive profile usage are also occurring during the same periods of the day which suggests that the user has not changed significantly his usage profile. Through the definition of thresholds, as outlined in section IV-A, it is possible to detect the exact time in which a suspicious usage of the profile occurred.

V. Conclusions

The number of social network users and profiles has increased immensely in the last years. This illustrates the increasing significance of these network as a means of communications and content sharing between different users. Indeed, social networks such as Facebook have become so widely adopted that several business and marketing models arose to adapt to the paradigms established by these new communication platforms. Moreover, since users can also share confidential information, the number of attacks and of fake or illicit profiles on social networks such as Facebook has also increased dramatically. Indeed, such networks have become a primary target for attackers who, by exploiting the trust relationships established by users, aim at disseminating compromised contents for purposes such as *phishing*, information gathering and profile hijacking. Therefore, users on social networks have, up to some extent, to protect their accounts from such attacks. Consequently, the task of detecting if a profile has been compromised, or *hijacked*, is of critical importance for the user security on social networks.

In this paper, we propose a novel framework that comprises a Facebook event collector which, by obtaining an authentication token from a target user, through a Facebook application, is able to export several information from the analysed profile and store it into a relational database for *posteriori* analysis and modelling. In order to guarantee the privacy of the analysed profile, only the *timestamps* of the analysed events are collected and stored. Subsequently, by

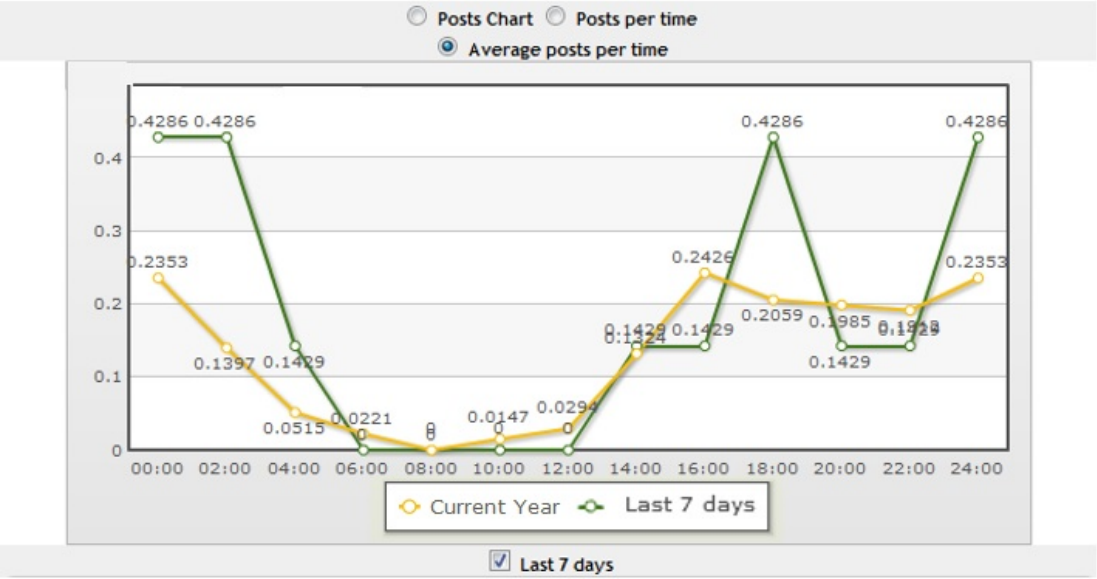

Figure 3: Average number of posts for two different periods of analysis.

analysing the collected *timestamps*, the proposed framework is able to compute usage profiles for user-defined periods of time and compare them in order to detect possible deviations from what can be considered as a normal profile usage. By defining thresholds based on the computed values, the framework is able to raise alarms to the target user indicating possible suspicious situations that may require the attention and inspection of the user. Finally, the framework also allows the addition of novel statistical analysis modules which can use the gathered data and create different user models.

ACKNOWLEDGEMENTS

This research is supported in part by Fundação para a Ciência e Tecnologia through the research project PTDC/EEI-TEL/2016/2012.

REFERENCES

[1] (2014, January) Facebook reports third quarter 2013 results. http://investor.fb.com/releasedetail.cfm?ReleaseID=802760.

[2] C. Perez, M. Lemercier, and B. Birregah, "A dynamic approach to detecting suspicious profiles on social platforms," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, 2013, pp. 174–178.

[3] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10. New York, NY, USA: ACM, 2010, pp. 1–9.

[4] (2014, January) Weka 3 - the university of waikato. www.cs.waikato.ac.nz/ml/weka/âĂŐ.

[5] H. Shen and Z. Li, "Leveraging social networks for effective spam filtering," *IEEE Transactions on Computers*, vol. 99, no. PrePrints, p. 1, 2013.

[6] M. Tubi, R. Puzis, and Y. Elovici, "Deployment of dnids in social networks," in *Intelligence and Security Informatics, 2007 IEEE*, May 2007, pp. 59–65.

[7] S. Mahmood, "New privacy threats for facebook and twitter users," in *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*, Nov 2012, pp. 164–169.

[8] (2014, January) Bitdefender safego - unfriend your phishy links! http://www.bitdefender.com/solutions/bitdefender-safego.html.

[9] W. Wei, F. Xu, C. Tan, and Q. Li, "Sybildefender: Defend against sybil attacks in large social networks," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1951–1959.

[10] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: Defending against sybil attacks via social networks," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 267–278, Aug. 2006. [Online]. Available: http://doi.acm.org/10.1145/1151659.1159945

[11] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, May 2008, pp. 3–17.

[12] (2014, January) Facebook developer products. https://developers.facebook.com/products.

[13] (2014, January) Mysql :: The world's most popular open source database. http://www.mysql.com/.

[14] (2014, January) Bringing mysql to the web. http://www.phpmyadmin.net/.

[15] E. Rocha, P. Salvador, and A. Nogueira, "Detection of illicit network activities based on multivariate gaussian fitting of multi-scale traffic characteristics," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–6.