Universidade de Aveiro
2013
Departamento de Eletrónica, Telecomunicações e Informática

José Francisco Mendes de Brito

Modelação Comportamental em Redes Sociais

Behavior Modeling in Social Networks

Universidade de Aveiro
2013
Departamento de Eletrónica, Telecomunicações e Informática

José Francisco Mendes de Brito

Modelação Comportamental em Redes Sociais

Behavior Modeling in Social Networks

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Doutor Paulo Jorge Salvador Serra Ferreira, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e coorientação do Doutor Eduardo Oliveira Estanqueiro Rocha, Investigador no Instituto de Telecomunicações de Aveiro.

Á minha família, namorada, amigos e a todos a aqueles

que ajudaram nesta caminhada

o júri

Presidente	**Prof. Dr. Luís Filipe de Seabra Lopes** Professor Associado, Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
vogais	**Prof. Dr. Joel José Puga Coelho Rodrigues** Professor Auxiliar, Dep. de Informática da Faculdade de Engenharia da Universidade da Beira Interior (arguente principal)
	Prof. Dr. Paulo Jorge Salvador Serra Ferreira Professor Auxiliar, Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)
	Dr. Eduardo Oliveira Estanqueiro Rocha Investigador Associado, Leipzig University Of Applied Science (co-orientador)

agradecimentos / acknowledgements

Esta dissertação representa o final de um percurso de quase 6 anos. Ao longo deste tempo foram muitas as pessoas que passaram na minha vida e que de certa forma contribuíram para aqui chegar.

Agradeço de uma forma especial:

À minha família, pelo seu apoio, incentivo e confiança, que foram fundamentais para a conclusão deste percurso.

A todos os meus amigos do Fundão, que já vinham de há muito e que estiveram presentes nos bons e maus momentos.

Aos colegas e amigos de Aveiro com quem tive o privilégio de partilhar este percurso e desempenharam um papel importantíssimo na minha vida.

A todos os meus amigos de Erasmus por terem marcado a minha vida.

Aos meus professores e orientadores, Paulo Salvador e Eduardo Rocha pela orientação e contribuição neste trabalho.

À minha namorada, por me ter acompanhado sempre, nos bons e maus momentos.

resumo

As redes sociais têm tido um crescimento viral nos últimos anos. No início do século XXI já se discutia a indispensabilidade da Internet e no presente, as redes sociais reforçam ainda mais esta ideia. O ser humano, ao longo da sua história, foi mostrando a necessidade de exprimir as suas ideias, os seus pensamentos, as suas alegrias, as suas tristezas… As redes sociais são assim um espaço onde as pessoas, de diferentes idades ou culturas, podem partilhar os seus pensamentos e experiências.

As redes sociais são desta forma um espaço apetecível para todo o tipo de ataques informáticos, especialmente de *phishing*. Nesta dissertação faz-se uma análise de diferentes redes sociais, das suas APIs e das formas de extrair informação das mesmas, dando especial enfâse ao Facebook. Como tal, foi desenvolvido uma ferramenta que utiliza esta informação e que permite monitorizar o comportamento de um utilizador, permitindo a verificação da legitimidade do seu comportamento.

Neste projeto foi utilizada a Graph API do Facebook, que se trata de uma API baseada no protocolo HTTP e que permite aceder à estrutura social (Social Graph) do Facebook, retornando os dados no formato JSON. Para fazer a ligação ao Facebook foi utilizado o Facebook PHP SDK. O script utilizado é independente do website e guarda toda a informação em JSON, estando os ficheiros organizados por tipo de conteúdo e pelo ID do utilizador. Desta forma o script pode ser facilmente reutilizado para outro tipo de ferramentas online ou offline.

abstract

Social networks are having a viral growth in recent years. The vital importance of the Internet has been under discussion since the beginning of the 21st century and social networks are reinforcing this idea. The human being, throughout its history, has been showing the need to express their ideas, their thoughts, their joys, their sorrows ... Social networks become then a tool which people, of different ages and cultures, can use for sharing their thoughts and experiences.

Social networks are an attractive place to all kinds of cyber-attacks, especially phishing. This dissertation analyzes different social networks, their APIs and how to extract information from them, giving more emphasis to Facebook. As such, a website was developed that uses this information and transforms it into a tool that allows users to monitor their behavior and to verify if it is legitimate.

In this model we used the Facebook Graph API, which is an HTTP based API that allows access to the Facebook Social Graph, returning data in JSON format. To connect to Facebook, the Facebook PHP SDK was used. The script is independent from the website and keeps all the information in JSON files that are organized by content type and user ID. In this manner, the script can be easily reused for other type of tools, online or offline.

Content

List of Figures

Chapter 1 - Introduction

1.1 Motivation and Context

Since the dawn of humanity that man has the need to communicate and share experiences, in other words, the need of interacting with a community. Social networks exist in this way for ages, since humans created the first forms of communication, such as painting, speech and writing.

With the arrival of the twentieth century the paradigm of social networking approaches to the concept which we know today, with the invention of television, radio and finally the Internet. It was in the late 90s that the first digital social networks, such as chats, began to emerge. The twenty-first century, driven by the strong growth of the Internet and technological development, marks the emergence of the large social networks of today, like MySpace, YouTube, LinkedIn, Twitter, Google+ and Facebook, where users can share their varied types of content such as photos, comments and personal data, thus becoming active producers of contents.

Social networks, due to its viral growth, became an attractive market for marketing, advertising and crime. The sharing of personal content from users enables the creation of a psychological profile of the same, which can then be exploited by people with good and malicious intentions. Thus, victims of social networks are increasing, derived of phishing attacks. These types of attacks are instigated by inexperience and unfamiliarity of many users, which make them difficult to detect. Indeed, most users ignore privacy and security settings, using their default and unsecure settings. According to recent reports, 1% of Facebook total users had his/her account hacked, which means 5 million of users were victims of social engineering. Despite all the efforts in the improvement of privacy and security settings, the number of attacks and victims is continuously increasing. It is in this context that the dissertation “Modeling Behavior in Social Networks” appears.

1.2 Objectives

The objectives of this dissertation are to understand the functioning of social networks and their APIs and how the creation of behavioral models based on user behavior, can prevent and detect compromised accounts or on the control of hackers.

It is also planned to address the security methodologies already implemented in social networks, especially Facebook, assessing their advantages and disadvantages and to what extent the creation of behavioral models can increase the safety and confidence in them. After these considerations we intend to create a website for presenting the obtained user models and evaluate its results.

1.3 Dissertation Structure

This dissertation is divided into the following chapters, excluding the present:

- **Chapter 2 – Social networks:** provides an introduction to the major social networks of today, focusing on Facebook, reflecting the dangers to which their users are exposed. This chapter also presents an overview of the APIs provided by social networks and discusses how it is possible to extract data from them.

- **Chapter 3 – Privacy and Security on Social Networks:** The viral growth of social networks, associated with the growing number of users, instigated an improvement of privacy settings and security. In this chapter we will address the privacy and security settings of Facebook and identify possible vulnerabilities.

- **Chapter 4 – Development:** In this chapter, the development strategy is presented. First, a general overview of the dissertation and of the project in which it is inserted in will be given and subsequently, all the details will be explained, using diagrams and figures for all the cases needed. It is also given a technical approach of the technologies used and how they were implemented.

- **Chapter 5: Conclusion and Future work:** this chapter presents some issues found during the development of this work, the achieved accomplishments and a proposal for further work improvements.

Chapter 2 – Social Networks

The concept of global village began to emerge when the Internet took its first steps. Social networks are nothing less than the culmination of this concept. Social Networks captivated many people from different ages. For some meant the first contact with the Internet. Nowadays approximately one sixth of the world population is connected to a Social Network. A Social Network site can be described as a web-based service that allows individuals to create a profile and connect with other individuals, creating relations between them. What makes a social network unique is not the opportunity of meeting new people (some security issues are related with strangers profiles that sometimes can be fake), but the opportunity to create connections that would not otherwise be made. During our lives we have relations with many individuals however, over the years some of these relations are lost. Social networks allow some of these relations to be reestablished while enabling the establishment of new ones. That possibility is embedded in its social graph architecture.

The features that Social Networks offer, shown in Figure 1, such as status messages, chats, creating profiles, creating pages and events, express interests (likes), make users inadvertently feel the need to project themselves to the Internet, so that people can notice their presence and interact with them.

Figure 1: Functionalities of a Social Network [1]

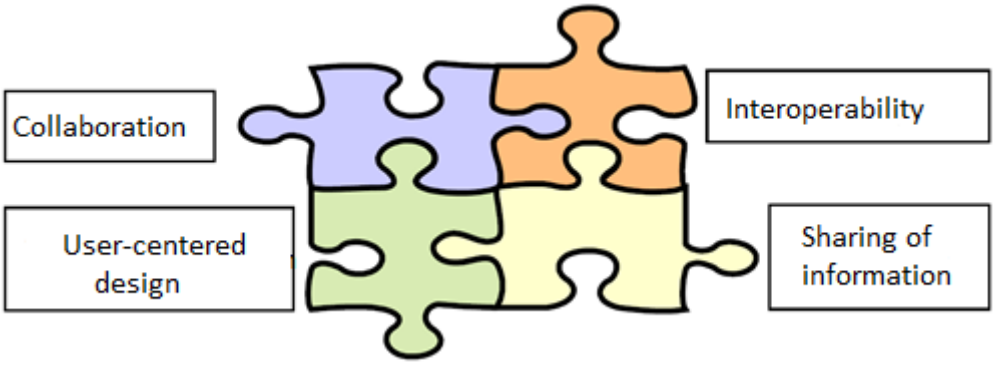

Figure 2: Characteristics of a social network [3]

Social networks go even further. According to the characteristics of a user profile, such as personal tastes, location, profession and friends, social networks suggest people that could be a "friend" [2]. We conclude that social networks features can be seen as a puzzle where the number of pieces is constantly increasing and having new applications, as visible in Figure 2.

All these features and the exposure of personal information make the users extremely vulnerable. This means an opportunity for many companies that use social networks as a huge market. The presence of companies in these networks creates a sense of approximation to the user / client, making it a powerful marketing tool. However, not everyone uses these vulnerabilities for “noble” reasons. Social networks users, due to the amount of sensitive information shared, can become daily targets of hackers whose intentions are unpredictable. The goal of this dissertation involves observing user behavior and creating behavioral models to identify users who may have been compromised.

2.1 Victims of social networks

Most social networks users are unaware of the dangers they are exposed to on social networks. Many profiles are public and can be easily found with a simple search query on Google. Facebook, as the biggest social network, has a huge concern about privacy settings. Their privacy settings allow the navigation through 50 settings with more than 170 options. This strategy offers precise control of the data shared however it brings more complexity, which will drive off a regular user [4].

On average, users of social networks see their privacy settings every 3 months, which does not imply that they change them. Recent statistics reveal that “nearly 13 million U.S. Facebook users are not aware of the social network's privacy control settings or simply do not use them at all” [5]. The result is that 28 percent of U.S Facebook users share all, or almost all, of their wall posts with more than just friends (Figure 3).

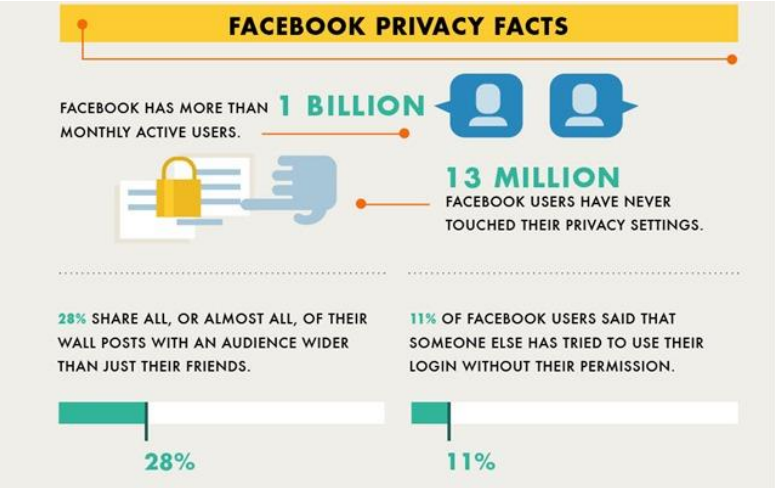

Figure 3: Facebook Privacy Facts [6]

Andrew Noyes, Facebook's manager of public policy communications, after being confronted with these facts said that Facebook "believe that more than 900 million consumers have voluntarily decided to share and connect on Facebook because they provide them options and tools that place them in control of their information and experience". Another important matter to take into account is the existence of fake profiles. Around 80 million of Facebook accounts are fake or duplicated, around 8% of the active users. Some of these accounts represent pets or organizations, but the real problem are the accounts used for spam or for phishing attacks. According to Barracuda Networks study, 97% of fake profiles are women and have more than 700 friends [7]. Adding a fake profile into users' connections makes them desirable target to malicious people, as can be seen in Figure 4.

Figure 4: Risk of posting in Social Networks [8]

There are some good habits to avoid becoming a victim of scam or phishing. Most of these habits are trivial for advanced users however, the biggest percentage of Social Network users are inexperienced and have low informatics skills [9]:

- Do not enter personal information on a website if there is not certain that is genuine. Always keep an eye on the browser address bar.
- Check the privacy settings regularly and check how much information user information is available on the internet (use a search engine to test).
- Be careful about what personal information is put on the internet. Giving away too much information can give clues to scammers for brute force attacks to user credentials.
- Check a user profile before accepting his/her friend request (do not accept "strangers" and remember, online "friends" may not be who they say they are).

- If an unexpected request for money is received, from what appears to be a friend, try to contact that friend or their family or friends to verify the request.

These security habits take special importance due to recent statistics. According to Barracuda Networks, 91,9% of social network users received spam, 54,4% were victim of phishing attempts, 23,3% received malware, 16,6% had their accounts used to send spam and 13% had their accounts hijacked or had their password stolen. [10]

2.1.1 *Social Engineering*

Social Engineering, in computer science, is the term used for practices seeking to acquire personal and confidential information by manipulation and persuasion, taking advantage of the naivety of people. A social engineering attack exploits specific attributes of human decision-making and relies on the fact that people are not aware of the value of the information they possess and share, which makes Social engineering the hardest form of attack to defend against because it targets the weakest link in every security system, the human being [11]. Due to the tricky atmosphere it creates, social engineering is also referred as a con game (confidence trick). The process behind a con game is well defined. The first step is the preparation. The social engineer sets the target and builds a plan. After planning every step, the social engineer starts the most delicate phase, the approach to the victim. In this stage the prior objective is getting the confidence of the victim, rousing his greed and clouding his judgment. The last stage will be the defraud process.

One of the most practiced attacks methodologies in social engineering is phishing. In this practice, the attacker is passed by a trusted person or organization to obtain confidential information. The attacks usually occur via email, through websites or through malware. The most used techniques are DNS spoofing (consists in forging a response from a DNS server, making the URL of a site point to a different server than the original) and Forged URLs (false URLs, usually extensive, designed to deceive the user, looking credible). Figure 5 shows a possible example of these two techniques.

Figure 5: Example of phishing using a false Facebook Login Page [12]

The previous figure apparently seems to be a reliable form from Facebook however, if we look at the URL on the browse bar we see that it is a faked version. This example dates back to 2007, this is the current form of Facebook (Figure 6).

Figure 6: Facebook Login Page (December 2012)

Analyzing the two figures we can see that their appearances are very similar, even taking into account that the look of Facebook has changed in the 5 years that separate the images. For many users these differences go unnoticed and, in this case, the result would be turning over the Facebook login credentials.

The concept of Social Engineering attacks is being increasingly addressed, especially in the business and organizational world where employees are often victims of these kinds of attacks, due to the ownership of confidential information. However, most organizations already have in place some prevention mechanisms, fact that does not happen with the majority of the social networks users.

2.1.2 *Identity Theft*

Identity theft, as the name suggests, consists in using personal information from someone else to impersonate this person. It is usually the next step to a phishing attack.

Identity theft on social networks can have different goals. One of the most ingenious attacks was directed to Bryan Rutberg, former director of communications at Microsoft. In 2009, a professional hacker accessed his Facebook account. The hacker invented a scenario where Bryan and his family (to be more credible, he used the names of family members, as they were in the Facebook friends list of the victim) had been mugged while travelling in London and were without money to return. With some prodding, the hacker managed to fool one of the victims' friends, which sent $ 1,200 - this money was raised in London by someone claiming to be Bryan Rutberg. [13]

This it is a known case; however, there might be even more severe cases whose release is prevented to avoid smear of Facebook or to prevent victims to be even more vulnerable.

2.2 Facebook

Facebook was founded by Mark Zuckerberg in 2004. Initially the purpose of Facebook was bringing together people and cultures around the world. Rapidly, due to strong adhesion of people, these noble objectives became the background, making Facebook a leading global Internet business. Nowadays, Facebook has around 1 billion active user. An average user spends about 23 minutes a day on the website, clicks on the "Like" button 9 times per month, shares around 415 contents every year and has around 140 friends. Each one of these characteristics is increasing day by day. Other interesting statistics reveal that Facebook has around 50 million pages, 10 million apps and since its launch, the like button was clicked more than 1,13 trillion times.

2.2.1 Facebook API

In this section it will be addressed the tools provided by the Facebook API and also its characteristics.

2.2.1.1 Graph API

The Graph API is the primary tool to get and post data to Facebook's Social Graph [14]. It is intended to be the next generation of the Facebook API, replacing the FQL and the legacy REST API. Basically the Graph API is a low level HTTP-based API used to query data. High level toolkits like Facebook's PHP, iOS, Android and JavaScript, allow developers to easily deal with the Graph API, abstracting some actions that might be problematic [15].

Moreover, the Graph API allows retrieving fields and connections from the Social Graph. To have a better understanding of the subject, the following topic will summarize the key aspects of the Social Graph.

Facebook Social Graph

A Social Graph is the core of a social network. It is a graph that portrays all the connections and relationships between users. The following image is an example of a social graph.

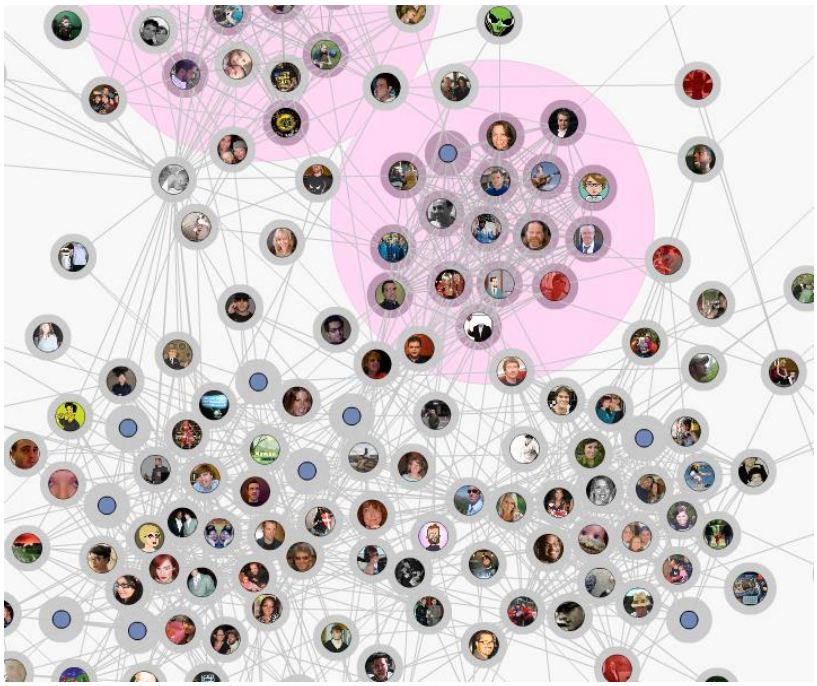

Figure 7: Example of a Social Graph [16]

This graphs seems really complex however, the idea behind it is really simple. What makes it look so complex is the number of nodes and respective aggregations. Basically a node can be a person, a place, a page, an artist etc. A connection establishes the relation between nodes, for example, the family relation. The next graph shows a simple example of a relation.

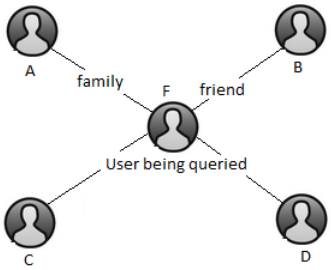

Figure 8: Relations between nodes

User F is the user being queried. User A,B,C and D share the connection friend with user F. Only the user A shares the connection family with user F. Now we are going to use the Graph API to query the node and both connections. The following images show the respective results (all the data is retrieved in JSON format):


```
{
  "name": "User F",
  "birthday": "04/02/1989",
  "hometown": {
    "id": "107577052605143",
    "name": "Fundão, Portugal"
  },
  "gender": "male",
  "id": "100000563031180"
}
```


Figure 9: Result of the user F query with the fields name, birthday, hometown and gender

```
{
  "name": "User F",
  "id": "100000563031180",
  "friends": {
    "data": [
      {
        "name": "User A",
        "id": "508214233"
      },
      {
        "name": "User B",
        "id": "512170703"
      },
      {
        "name": "User C",
        "id": "513090752"
      },
      {
        "name": "User D",
        "id": "514917418"
      }
    ],
  }
}
```


Figure 10: Result of the user F query for friends connection


```
{
  "name": "User F",
  "id": "100000563031180",
  "family": {
    "data": [
      {
        "name": "User A",
        "id": "508214233",
        "relationship": "brother"
      }
    ],
  }
}
```


Figure 11: Result of the user F query for family connection

In conclusion, the Graph API is a powerful tool to get data into and from Facebook Social Graph. It allows users to customize the retrieved data, in other words, it is retrieved only the information wanted without retrieving anything that was not queried, what may result in a better performance of the application.

2.2.1.2 Facebook Query Language

The Facebook Query Language (FQL) allows developers to query Facebook data using the Graph API. FQL is based on SQL syntax:

SELECT *data* FROM *table* WHERE *condition*

However, unlike SQL, FQL can only retrieve data from a single table. A possible solution to this issue is adding sub queries so that the access to multiple tables is achieved. FQL can handle simple match, basic boolean operators, AND or NOT logical operators, ORDER BY and LIMIT clauses.

To check which tables and columns we can call, Facebook provides the FQL Table Reference. The following query is an example of a FQL query for the table user.

SELECT name, birthday, hometown_location FROM user WHERE uid=me()

Like in the Graph API, all the results provided are returned in JSON format. In this case the result of this query would be:


```
{
  "data": [
    {
      "name": "Francisco Brito",
      "birthday": "April 2, 1989",
      "hometown_location": {
        "city": "Fundão",
        "state": "Castelo Branco",
        "country": "Portugal",
        "zip": "",
        "id": 107577052605143,
        "name": "Fundão, Portugal"
      }
    }
  ]
}
```


Figure 12: Result provided from the Graph API Explorer

In conclusion, FQL is an easy way to retrieve data from Facebook and most important, FQL allows the customization of the retrieved information, like in the Graph API.

2.3 Twitter

Created in 2006 by Jack Dorsey, Twitter is a social network with a fairly simple concept: to allow users to share written text messages up to 140 characters. These text messages are known as Tweets. Unlike other social networks, Twitter allows non-users to read Tweets published; however, only users can write and share.

Recent statistics indicate that Twitter has over 140 million active users and that are published about 340 million Tweets per day. (Twitter 2012)

2.3.1 REST API

REST (Representational State Transfer) defines a set of architectures used in distributed systems. The RESTful architecture follows the following principles [17]:

1. Addressable Resources. Every “thing” on the network should have an ID. With REST over HTTP, every object will have its own specific URI.

2. A Uniform, Constrained Interface. When applying REST over HTTP, stick to the methods provided by the protocol. Simply put, REST enables to access information or resources through a simple HTTP invocation, using the methods GET, POST, PUT and DELETE.
3. Representation oriented. The interaction with services is done using representations of that service. An object referenced by one URI can have different formats available. Different platforms need different formats.
4. Communicate statelessly, there is no client session data stored on the server. This means that all requests are treated as an independent transaction. In this manner, all requests have the necessary information to complete a transaction.

The following picture represents a REST Web-oriented architecture.

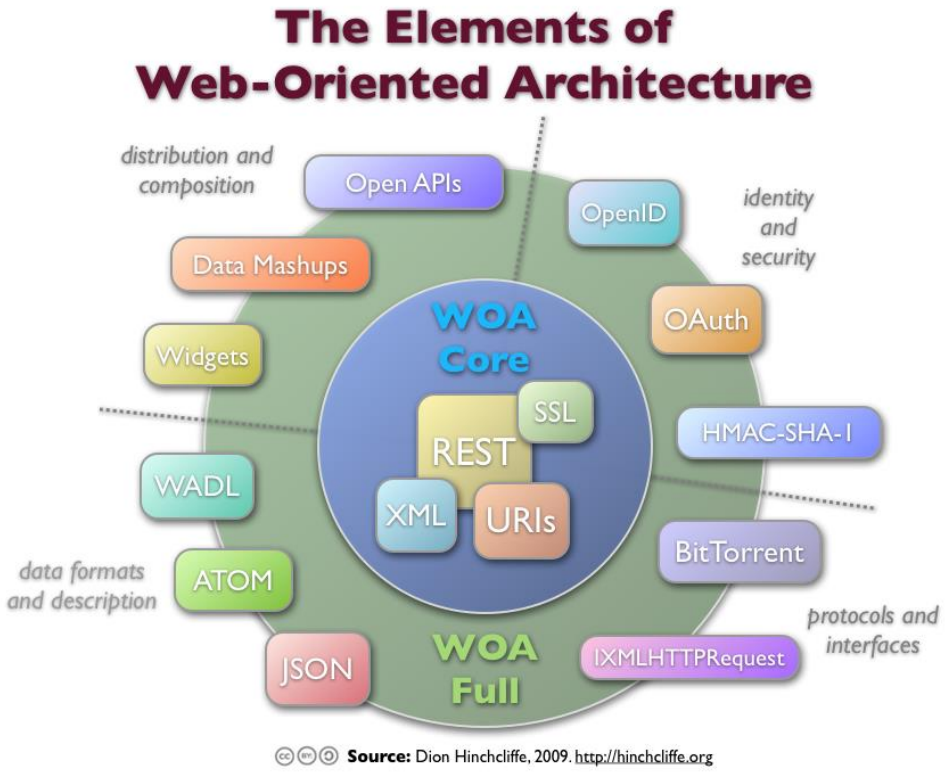

Figure 13: Elements of Web-Oriented Architecture [18]

Twitter's REST API allows other web applications to access resources on Twitter. The following image shows the example of an API request, like most Social Network APIs the result is returned in JSON :

```
GET    https://api.twitter.com/1.1/statuses/mentions_timeline.json?count=2&since_id=
       14927799
```

Figure 14: Twitter API GET method

2.3.2 Streaming API

The Streaming API is an alternative to Twitter's REST API however its goals are slightly different. It is a tool that offers developers low latency access to Twitter's data. It offers real time updates and unlike the REST API, does not need to constantly poll REST endpoints, which can generate high latencies.

Twitter offers three different types of streaming endpoints, each customized to certain used cases [19]:

- **Public streams**: Streams of the public data flowing thought Twitter;
- **User Streams**: Single-user streams, which contains almost all data corresponding to a single user.
- **Site streams**: The multi-user version of user streams.

Both APIs use HTTP however the architectures are different. The following image shows how the REST API operates.

As we can see, the user does a request and Twitter gives the response. In this situation Twitter can only respond to a user request. The Streaming API uses a daemon that handles the streaming connection process.

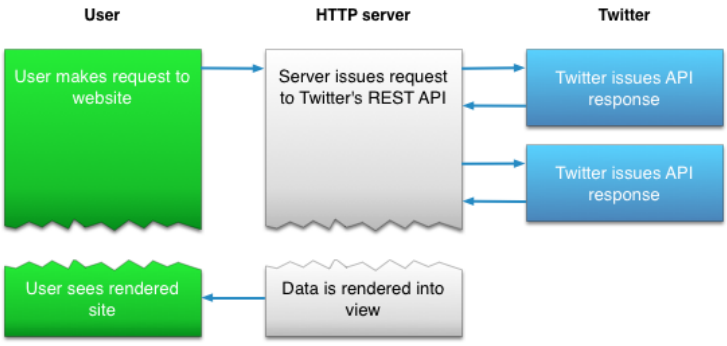

Figure 15: Streaming API message flow [20]

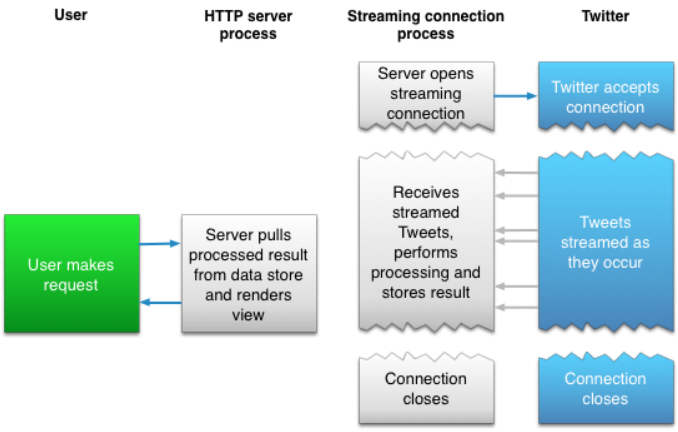

Figure 16: Streaming API message flow [20]

In this situation, the daemon responsible for the streaming connection process, opens the streaming connection and receives real time updates, which will be processed and stored. The HTTP server process queries the data stored in response to a user request.

The Streaming API is more complex but the benefits from having real time stream of Tweets data makes its integration worthwhile for many types of applications.

2.4 Google+

Google+ is a social network launched by Google on the 28th June of 2011. One of the goals of this social network was to aggregate other Google services and most important, it was a response by Google to the Facebook dominium in the social networking.

Google+ had a benefit in its architecture since it had a chance to learn with other social networks weak points, which consisted on more simplified privacy settings. As explained before in this chapter, privacy is the biggest concern in social network. Facebook offers a huge amount of privacy settings, making its customization really complex. On Facebook each type of content has its own privacy settings. Therefore Google+ created a new way to control the shared information, the circles. A circle is similar to a group, where it is possible to define the friends circle, the family circle, the work circle… In this manner is possible to share content with any of these circles independently, in other words, if information is shared in the friends circle, other circles will not have access to these data. Another interesting feature is the Hangouts. Unlike Facebook, that uses the Skype API for video calls, Google+ allows free video calls up to ten users.

Google+ brought some innovations as a new social network and unlike Facebook, it is free of ads. However, Facebook has already around 900 million active users and many of them might not have interest in migrating from a social network to another. That might be a concern for Google+, which number of active users is around 250 million.

2.4.1 REST API

Like other Social Networks, Google+ offers an API that allows developers to integrate Google+ features within their apps or websites. It follows a RESTful API design, using HTTP methods (GET, POST, PUT and DELETE) to retrieve information. The information retrieved comes in JSON format.

```
GET https://www.googleapis.com/plus/v1/people/{userId}
```

Figure 17: HTTP Invocation for user profile retrieval [21]

2.5 Conclusions

In this chapter we started by defining social networks vulnerabilities giving examples of the typical approaches followed by hackers. After we introduce three of the biggest social networks on the Internet, explaining how they work, introducing their features and presenting some of their APIs capabilities. These APIs allow developers to integrate social network features within their website, extending somehow the dominium of the social network. Therefore most websites on the Internet are connected with social networks, displaying plugins like the like button or the plus button, or including data from the social graph. The possibilities are uncountable and these APIs give innumerable ways to explore the social graph. The Facebook Graph Search is probably one of the most recent and revolutionary explorations of all the data contained in a social network like Facebook.

Later in this thesis we will introduce and explore a model of a security tool based on the data extracted from Facebook Social Graph.

Chapter 3 - Privacy and Security on Social Networks

The viral growth of social networks, associated with the growing number of users, implied an improvement on privacy and security settings. In this chapter we address the privacy and security of Facebook and identify possible vulnerabilities.

3.1 Privacy

The Facebook privacy settings have been constantly changing as the social network has grown and as users' critics emerged towards some privacy issues. This is due to the characteristics of the network itself, which provides more and more resources to its users.

Issues related to the privacy of user data have generated some controversy. This is due to the fact that for many users, most of the information shared becomes public. Again, the source of the problem is the lack knowledge of users. Most users do not read the Facebook Data Use Policy but agree with it during the registration. The result is that many users use the default privacy policy at the time of their registration and are unaware that these settings can be customized.

If the user chooses to make their information public, he/she allows anyone, even if not registered Facebook, to have access to his/her profile information. The result is that information:

- may be associated with the user in question (name, profile pictures, chronology, ID, etc.);
- can be displayed in a Facebook search or even in a public search engines;
- can be accessible by all kinds of Facebook embedded applications and sites and will be accessible to all people using the Facebook APIs.

As already discussed, Facebook allows the customization of privacy settings however, due to the increase of features in the social network, the privacy setting became really complex. The base of every privacy setting is the audience selector. Like the name suggests, it is possible to choose who can see users' posts:

- **Public** – anyone in the Internet can see the profile information.
- **Friends of friends** – the user profile is visible to all friends and also to friends of friends. According to social network characteristics this relation can have an huge growth therefore, this setting is the most vulnerable after the public profile.

- **Friends** – that is the most common audience in Facebook. The user shares his post and profile information only with his/her friends.
- **Only Me** – this option is really unusual. The user is visible only to himself/herself. However if someone is tagged in a post with Only me option, they will be able to see the post.
- **Custom** – in this option is possible to select who can see a user post. The user can select a specific person or groups inside his/her circle of friends, like family or best friends.

The complexity comes when the general setting are divided into timeline and tagging setting and when these setting are divided in even more settings, basically Facebook has privacy settings for everything. It is too much to handle for a regular user. Many users are not even familiar with these concepts. One good example is photos posts. Facebook groups user photos in albums however, a user cannot set an audience for an album, he/she has to set the audience for each photo posted. There is no general setting to set an audience for all photos.

The name, profile picture, cover photo, sex, name and user ID are information that is always publicly available. [22]

For testing purposes, we created a Facebook account. The privacy settings were not changed, in others words, we used Facebook's default settings. After observing all the parameters of privacy policies it was possible to conclude that most of the user information was public.

Hacking Facebook accounts is security issue; however, ignoring the privacy settings makes the user more vulnerable to such attacks.

3.2 Security

With the increase of attacks on social networks, especially on Facebook, security tools have been improved. The types of attacks are increasingly diverse (adware, malicious scripts, malware, phishing), so it is necessary that the deployed safety measures also become more diverse.

3.2.1 Secure Connection (HTTPS)

HTTPS is an implementation of the HTTP protocol that uses an additional security layer (SSL / TLS). HTTPS provides bidirectional encryption between the client and server. It is used in websites that deal with sensitive information because it gives protection to man-in-the-middle attacks. Five years ago, a network protocol analyzer such as Wireshark was sufficient to figure website credentials, especially within a local network, because unlike today, some authentication processes were done with the HTTP protocol, without any additional security layer. Therefore, Facebook allows HTTPS browsing so that all the data exchanged between the

user and Facebook server are encrypted, making that data very difficult to intercept and decrypt. The following figure shows the HTTPS packet flow during an HTTP request.

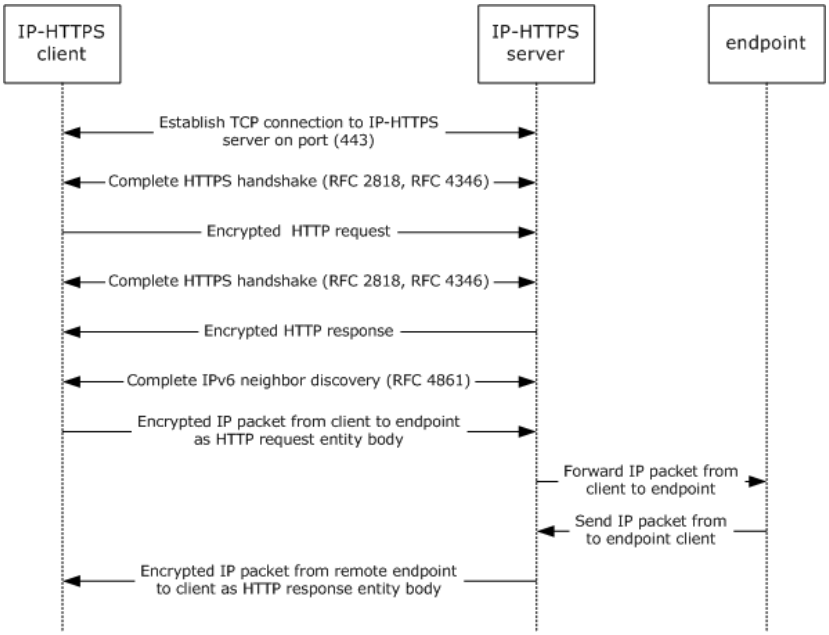

Figure 18: HTTPS packet flow [23]

3.2.2 Login Notifications

This tool allows the user to receive a notification, by email or text message, reporting if the account was accessed from a computer, mobile phone or other device that has never been used before. This is an effective tool because it allows the user to know in real time if his/her account was accessed, so that he/she can act quickly, minimizing any potential malicious acts.

3.2.2.1 Recognized Devices

Whenever the user logs into Facebook in a new device, it is possible to name it and recognize it as legit. In the future, whenever using Facebook from such devices, the user will not receive notifications of access. Also, whenever a device is recognized the user receives an email and / or text message.

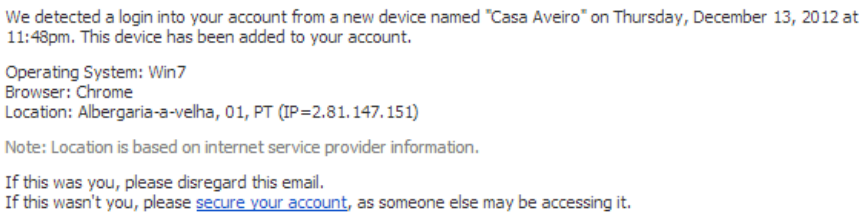

Figure 19: Notification example of a recognized decide

Figure 19 shows an example of a notification received by recognizing a device. If there is an access to the account by an unrecognized device such notification will have the following appearance (Figure 20).

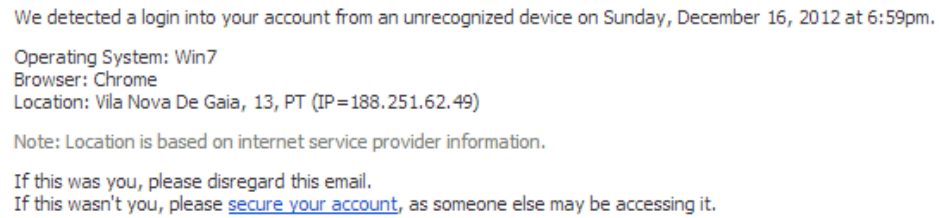

Figure 20: Notification example of an unrecognized device

3.2.3 Active Sessions

Active sessions, as the name suggests, shows the user the devices where their session is initiated and also allows users to finish the referred sessions. This tool complements the previously mentioned tool.

3.2.4 Getting access to an hacked account

The previous topics are preventing security tools yet Facebook has extra security tools to recover hacked accounts. After hacking an account, the first set is usually to change the password. This will mean that the legit user would lose is account and all respective information. To avoid it Facebook came up with a few solutions that can authenticate the true owner. We will explain briefly how they work:

- **Security Question** – The security question is a common security tool used in most of the authentication processes. The process is really simple, the account owner sets a personal question and the respective answer, after saving changes, the security question is set and cannot be changed. If the owner cannot access to his/her account he can recover the account giving the correct answer to the question. However this security tool can only be effective if a user sets a question that is impossible to guess. Another issue is that if the true owner to not set the security question, the hacker might do it, giving one less option for the true owner to recover the account.

- **Trusted Contacts** – In this security setting the user has to select three trusted contacts. If the account owner cannot access to the account or email, the selected friends can help him/her to recover account password. If the user selects this recovering method a URL will be given to him/her. After that he/she has to call the trusted contacts and give the URL. This URL will provide a code to each contact. To end the process the trusted contacts have to give their codes to the owner so he/she can complete the recovery. It is recommended that the user meets the trusted contacts in person or talks with them by phone, only that way they will know that they are giving the codes to the right person.

3.3 A new approach

In this section we will address some strategies that approach to the security tools based on user behavior.

3.3.1 Facebook Social Authentication

In the beginning of 2011, Facebook introduced a new security feature for testing: the Social Authentication. The concept of this security method is based on the intuition that a user can recognize his/her friends while a stranger cannot. At first sight the idea seems promising; however, we can easily find some problems.

First, it is important to understand what Social Authentication pretends to achieve. The first goal was to replace the traditional *captcha*, where the user has to decipher random words that a computer cannot. The problem is that some words are really difficult to decipher even for humans. However the true purpose of Social Authentication is to prevent hackers from performing phishing attacks [24]. So what is the precept for this security method? A hacker might know the user password but they do not know who the user friends are and how they look like. The following image shows how this method looks like.

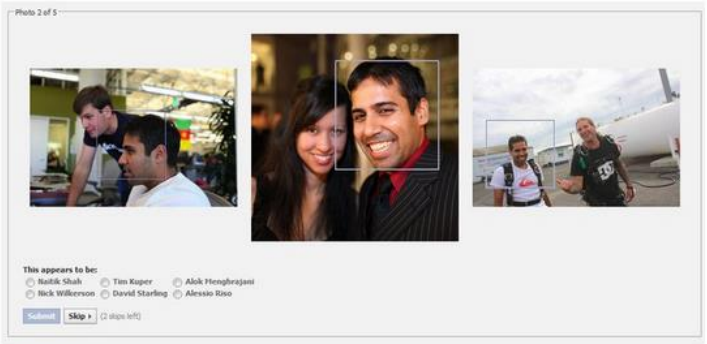

Figure 21: Social Authentication [25]

As we can see, the Social Authentication Security method provides 5 different challenges to the user, each one with 3 photos from a Facebook friend. According to the images the user has to put the correct name of the friend. During the test, it is given the idea that if a challenge is failed the login will not be allowed; however, that is not true. Even with failed challenges it is possible to login, however that will be recorded as an odd behavior. Most users never had a chance to perform this test because it only appears in accounts with odd behavior. For example, if Facebook detects a login in Portugal and one hour later detects another login from the same account in Australia, they will perform the Social Authentication method to verify the identity of the user, to be sure that the account has not been compromised.

Having the goals of this new security system in mind, let us now consider the problems within this method. Probably, the biggest issues are the photos with unidentifiable faces. Facebook chooses profile photos randomly. Even if only photos with faces are selected, the photo might not contain the user referred or even if it does, his face can be unidentifiable. Another issue is that Social authentication may be effective against pure strangers. However friend's information might be not private enough. All phishing attacks are based in social engineering, which means that the attacker has already a lot of information about the user being attacked or that he can be already inside the user social circle, for example, pretending to be friend. In both situations the attacker might have the information needed to pass the challenges. Even guessing (racial appearance can increase the guessing probability) can originate a security flaw, although it is not a common method for professional hackers. On the other hand this method can also give even more information to the hacker about the user being attacked.

Social Authentication Security Method appeared in January of 2011. It was an approach to human behavior security systems however due to the bad criticism, Facebook decided to put this feature in stand-by.

3.3.2 Fake Profiles Detection

Facebook fake profiles are one of the biggest security breaches in Facebook Security system. Facebook has around 83 million fake profiles. Some of this profiles are duplicate accounts, spammers and pets however, the biggest concern are the fake profiles used for social engineering attacks. These accounts owners usually use fake accounts to get closer to the target and gain is confidence.

Most Facebook users accept strangers as friends and do not even check the profile before accepting the friend request. However before accepting the friend request, users can only see public information. That way the only possibility is to accept the friend request and check it. Usually fake profiles have many differences when compared with real profiles. According to Barracuda network study, these are some of the differences:

- 97% of Fake profiles identify themselves as females. On the other hand, real profiles have on average only 40% female profiles.
- 58% of Fake profiles are interested in males and females. For real profiles the percentage is only 6%.
- The average number of friends for a Fake profile is 726 against real profiles 130.
- 68% of Fake profiles claim to attended college. Taking in account only real profiles this percentage decreases to 40%.
- Real profiles have on average 1 tag for every four photos. Fake profiles have 136 tags for every four photos.
- 43% of Fake profiles never had their status updated.
- Fake profiles do not have any interests (page likes) or have only few.

With these characteristics is possible to create a tool to detect Fake profiles. It is possible that Facebook already has an algorithm to detect fake accounts however allowing this tool for public use could generate some controversy because of the possibility of finding false positive results. However creating an independent tool could be a solution. In conclusion, decreasing the number of Facebook fake profiles would reduce the number of social engineering attacks and therefore increase the reliance of users in social networking.

3.3.3 Detection of Identity Clone Attacks

In the previous section we present a solution to detect fake profiles however there is another similar threat, cloned profiles. Detecting a cloned profile can have some similarities with detecting a fake profile yet they pose different problems. While a fake profile creates a profile based on a fake or random person to get closer to the target, a cloned profile is an example of Identity Theft. The attacker steals the profile information of the victim in a social network or in the victim homepage and after that he forges it in another social network site where the victim is not registered yet. After forging the profile, the attacker starts to rebuild the relations of the victim by contacting his friends who have accounts on both sites. Due to the lack of awareness this method can pose a huge security threat. If the attacker gains the trust of the victims circle of friends is not only a privacy attack, this may cause financial loss or severely after the friendships build on social networks.

To perform the attack, the attackers can follow different strategies based on the information they have:

- **Attribute value as a target** - An adversary creates a faked identity that has several similar attributes to those of the victim.
- **Privacy setting as a target** - An adversary creates a faked identity that has several similar attributes to those of the victim but also follows his privacy settings.

To detect cloned profiles there are three suggested ways, attribute similarity, friend network similarity and basic profile similarity. The attribute similarity, like the name suggests, compares the attributes of both profiles and calculates the similarity between them. The friend network similarity calculates the similarity of two identities friend network. It is usually taken into account not only the friend list but also the recommended list of friends. Yet users can set their friend network as private. Such activity may defend them from identity clone attacks however this is not a common practice since it makes a profile less popular and unlike

privacy and security setting, this would be a concern for many users. The basic profile similarity uses both of the previous strategies. This method detection is based on the number of similar attributes and the number of mutual friends in the two identities. Due to the possibility of false positives is necessary to establish a threshold. This threshold will define which profiles will have to pass the validation process.

The validation process is the last step in the detection of cloned attacks. In this phase all the suspicious profiles have to validate his identity. One possible approach to identify users is asking for their real world IDs. However if the attacker does his job well, he might know the user ID, so this method is usually discarded. MysafeFriend is a third-party application on Facebook that sets five levels of trust when identifying and identity. The application asks an identity to choose his friends to verify himself. The more friends verify this identity, the more points it gets and eventually it gets promoted to a higher level. Usually, at the highest levels, the application checks the credit card identity to verify it. However there is some controversy around this method. Asking a set of friends to validate an identity is not secure enough. These can be eluded by the cloned profile. One possible solution would be to ask the trusted friends to design some questions for the user and validate the answers from him. This way the possible attacker would be challenge with more personal data. Another possible approach is Social Authentication. This method was discussed before in this chapter. [26]

In conclusion, identity clone attacks are becoming a significant threat in Social Networks. The users lack of awareness made them ignore this threat that can severely affect the trust relationships among users in social networks and also expose personal information. Detecting these attacks will increase the users trust in social networking.

3.4 Conclusions

The Security tools provided by Facebook can be efficient if used by an experienced user that is regularly connected to the social network, which in most cases does not happen.

The main problem of Facebook security tools and other social networks is that they cannot adapt to the user, in other words, an inexperienced user will make the security tools become ineffective. There are several conditioning factors; one of the most relevant is the age distribution of the users (Figure 21). Usually younger users are unaware of the dangers that social network can pose therefore they will ignore any security tool.

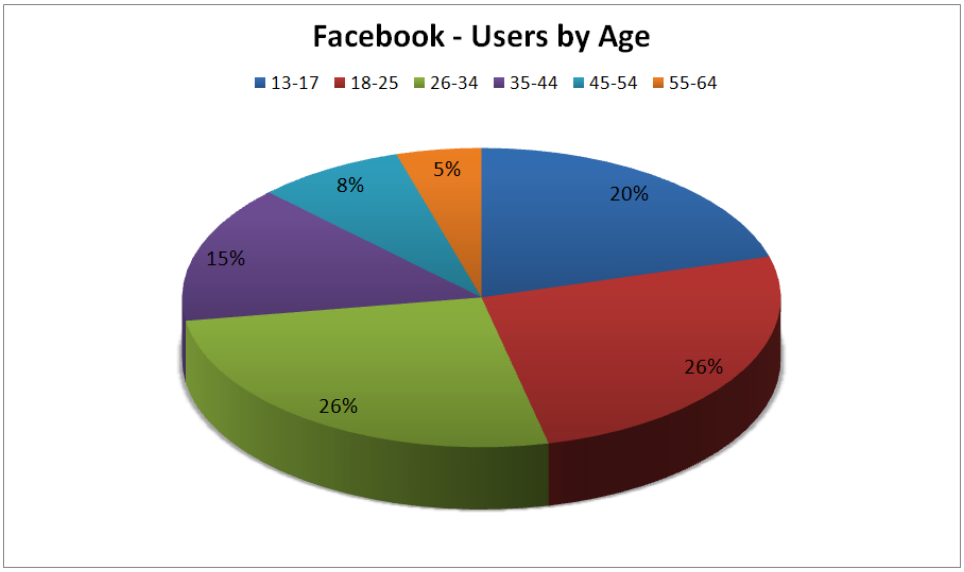

Figure 22: Distribution of Facebook users by age

Creating security tools that take into account user behavior can be seen as a support to the tools listed in this chapter. Most of the tools discussed are related with authentication and if an account is hacked this can pass unnoticed for the user even with security tools like login notifications enabled. On the other hand, these tools can also have prevention as a goal. Detecting fake accounts, which is based on user behavior, will reduce the user visibility, that way it will be harder to be hacked. In conclusion, all security tools can be effective, if used. According to recent studies 25% of Facebook users ignore security and privacy settings. Therefore the first step to suppress these attacks is too educate and aware all users for the risks of social networking.

Chapter 4 – Development and Usability Tests

In this chapter, the development strategy is presented and explained. We will present an overview of the research project, of the implementation process and of the technical approach for the selected technologies.

4.1 General overview

There are many definitions for behavior modeling, most of them have as objective predict peoples' behavior. In this thesis we will use a similar approach. Our objective is to create, through social network user data, a fingerprint of the user behavior.

Most of the related work in this area considers data from groups of users and never considers a model based in only one subject. Most users have a routine in social networks, for example, post comments, post photos, like a friend's photo or comment etc. Facebook allows the retrieval of most user information such as the number of likes, the number of wall posts, the number of friends, the number of friends' requests and many more. If we include a timeline and perform a temporal analysis with this data, we can analyze the user's behavior by his previous actions and model it. The result would be something like: the user X has an average number of likes per week, average number of wall post per week and an average number of friends request per week. If a great deviation from this behavior is detected, the account might be hacked or under attack.

In this chapter we will show a possible model and application using this concept of human behavior security method and discuss its benefits and its problems.

4.2 Model Proposal

As explained in the general overview, the prior objective of this project is to create a tool that helps the Facebook users to detect accounts that can be compromised, through the analysis of user's behavior. The idea of this dissertation is not to create a Facebook client with similar functionalities, but to create a tool based on Facebook data that can give a better overview about the user behavior than Facebook does.

The first step was to identify which aspects could be improved. Facebook mixes all kinds of data in the user timeline so it becomes complex for the user to control his activity or to find anomalies. Therefore, creating a timeline splitting information like status messages, likes,

photos and links can help the user to control these components. However, the best way to control user's behavior is to analyze it. For Pages or Apps, Facebook offers the Insights tools - "Insights provide comprehensive data about Page or App performance" [27].

Figure 22: Insights statistics of a page [28]

The previous figure shows an example of Facebook Insights for a page. However this tools is not extended for a singular user behavior. With so many cases of phishing or other attacks occurring on Facebook, it becomes difficult to understand why Facebook has not tried to use a similar tool for users. Untrusted apps which the user gave permission to post on his/her behalf usually spam the user friends list. Most users do not notice it, since posts on other profiles do not appear on the user timeline. With a simple chart showing the daily number of posts would be easy for the user to notice that his profile was being used for spam.

In this dissertation we will work on these considerations, creating a tool with a website as a front-end. The development of this platform will be divided in 3 stages:

1. Extraction of the information
2. Treatment of the information
3. Visualization of the results through a website

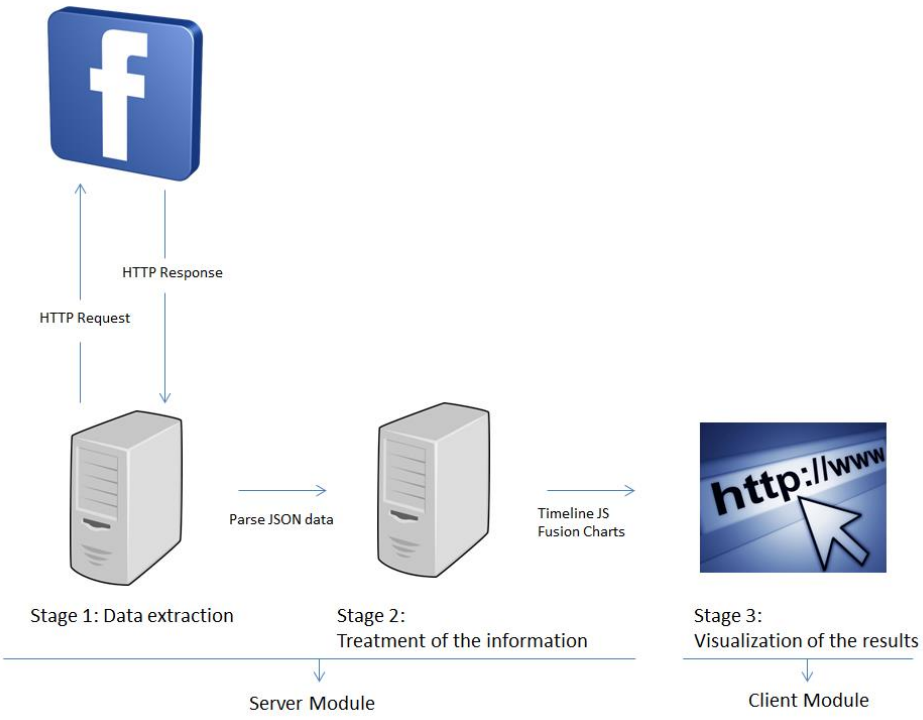

Figure 23: Project Overview

In Stage 1, the data is extracted every time a user logs into the website. The Graph API is used to acquire that information. However, to get or post data to the Social Graph, an access token is required. According to Facebook, “an access token is a random string that provides temporary, secure access to Facebook APIs” [29]. A token identifies a User, App or Page session and provides information about granted permissions. It also includes information about when the token will expire and which app generated the token. ” [29] All the data is returned in the JSON format.

In Stage 2, that data acquired is parsed and saved according to the Client Module needs. The application will use APIs such as Timeline JS and Fusion Charts that require specific data organization. In this stage we will also treat the information and use it for statistics, with the objective of helping in the representation of the user behavior.

The final stage will be the website, where the results from the previous stages will be shown.

4.2.1 Requirements

In this section, the requirements of this tool will be presented.

Usability

Usability is one of the most important factors when building a website interface. The primary objective of usability is making a product, in this case a website, easier to use, and to guarantee that the user needs and requirements are fulfilled. Usability is about effectiveness, efficiency and satisfaction. Making a graphical user interface is a process that is not

recommended to be accomplished without participation of the end user itself. According to Usability.gov, “usability refers to how well users can learn and use a product to achieve their goals. It also refers to how satisfied users are with that process “Usability measures the quality of a user’s experience when interacting with a product or system” [30]. Therefore, when building an interface, the following guidelines have to be considered [31]:

General design

1. Provide Useful Content
2. Establish User Requirements
3. Understand and Meet User's Expectations
4. Consider Many User Interface Issues
5. Focus on Performance Before Preference
6. Use Personas

Optimizing the user experience

1. Do Not Display Unsolicited Windows or Graphics
2. Standardize Task Sequences
3. Reduce the User's Workload
4. Design for Working Memory Limitations
5. Display Information in a Directly Usable Format
6. Provide Feedback When Users Must Wait
7. Do Not Require Users to Multi-task While Reading

Hardware and Software

1. Design for Common Browsers
2. Design for Popular Operating Systems
3. Design for User's Typical Connection Speed
4. Design for Commonly Used Screen Resolutions

The Homepage

1. Enable Access to the Homepage
2. Show All Major Options on the Homepage
3. Create a Positive First Impression of Your Site
4. Communicate the Website's Value and Purpose
5. Limit Prose Text on the Homepage
6. Ensure the Homepage Looks like a Homepage
7. Limit Homepage Length
8. Announce Changes to a Web Site
9. Attend to Homepage Panel Width

Page layout

1. Place Important Items Consistently
2. Place Important Items at Top Center
3. Structure for Easy Comparison
4. Establish Level of Importance
5. Align Items on a Page

6. Use Fluid Layouts
7. Set Appropriate Page Lengths
8. Use Moderate White Space

Navigation

1. Provide Navigational Options
2. Differentiate and Group Navigation Elements
3. Provide Feedback on User's Location
4. Place Primary Navigation Menus in the Left Panel
5. Use Descriptive Tab Labels
6. Present Tabs Effectively
7. Use Appropriate Menu Types

Scrolling and Paging

1. Eliminate Horizontal Scrolling
2. Facilitate Rapid Scrolling While Reading

Headings, Titles, and Labels

1. Use Clear Category Labels
2. Provide Descriptive Page Titles
3. Use Unique and Descriptive Headings
4. Highlight Critical Data
5. Use Descriptive Row and Column Headings

Links

1. Use Meaningful Link Labels
2. Link to Related Content
3. Match Link Names with Their Destination Pages
4. Repeat Important Links
5. Ensure that Embedded Links are Descriptive
6. Use Appropriate Text Link Lengths
7. Indicate Internal vs. External Links
8. Clarify Clickable Regions of Images

Text Appearance

1. Use Black Text on Plain, High-Contrast Backgrounds
2. Format Common Items Consistently
3. Ensure Visual Consistency
4. Use Bold Text Sparingly
5. Use Attention-Attracting Features when Appropriate
6. Use Familiar Fonts
7. Use at Least 12-Point Font
8. Emphasize Importance
9. Highlighting Information

Graphics, Images, and Multimedia

1. Use Simple Background Images
2. Label Clickable Images
3. Use Video, Animation, and Audio Meaningfully
4. Include Logos
5. Graphics Should Not Look like Banner Ads
6. Limit Large Images Above the Fold
7. Limit the Use of Images
8. Include Actual Data with Data Graphics
9. Display Monitoring Information Graphically
10. Introduce Animation

Writing Web Content

1. Make Action Sequences Clear
2. Use Familiar Words
3. Use Abbreviations Sparingly
4. Limit the Number of Words and Sentences
5. Limit Prose Text on Navigation Pages
6. Write Instructions in the Affirmative
7. Make First Sentences Descriptive

Content Organization

1. Organize Information Clearly
2. Ensure that Necessary Information is Displayed
3. Group Related Elements
4. Minimize the Number of Clicks or Pages
5. Design Quantitative Content for Quick Understanding
6. Display Only Necessary Information
7. Format Information for Multiple Audiences
8. Use Color for Grouping

Performance

Some non-functional requirements rely on performance issues. Usability is probably the most affected one. The strategy implied in this tool was to extract all the information when the user logs in. This will allow the user to have a fluid navigation.

Scalability

The website server should support several clients consuming the website.

Data integrity, privacy and security

The application ensures that the connection between Facebook and the website server are secure. Since the data is extracted from Facebook, the website trusts that Facebook considers this requirement.

4.2.2 Use cases scenario

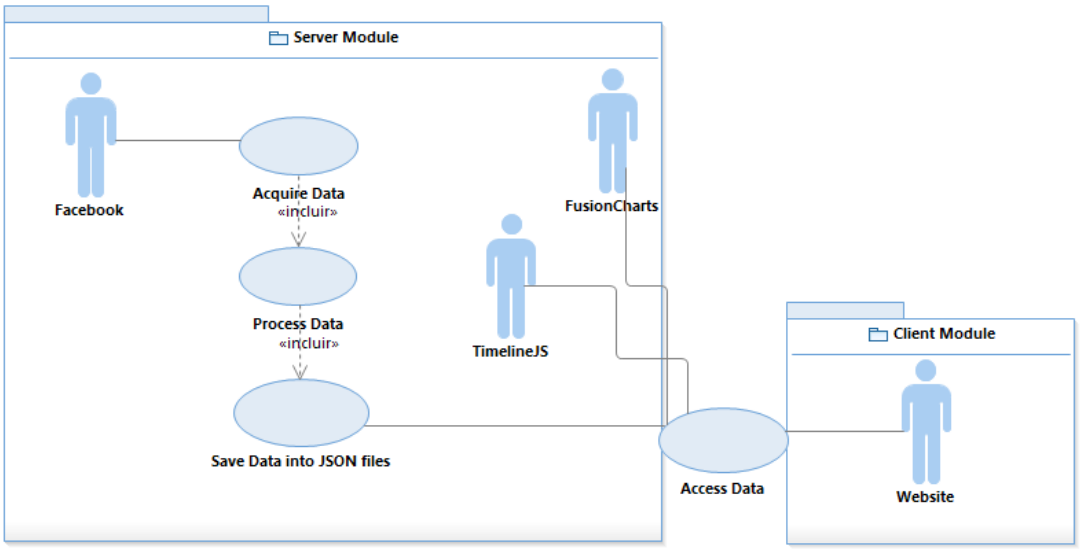

Figure 24: Use Case Diagram

Figure 25 represents the Use Case Diagram of the whole system. In the following section, each of its use cases is explained separately.

Use case 1- Acquire data

Goal - Acquire data from Facebook Graph API.

Actors - Facebook.

Pre-conditions – Login credentials and Access token.

Typical sequence of events – Acquire data**.**

Use case 2- Process Data

Goal – Parse the data from the Graph API GET.

Actors - Facebook.

Pre-conditions – All the acquire data process has to be concluded.

Typical sequence of events – Acquire data -> Process data**.**

Use case 3- Save data into JSON files

Goal – Save the data into JSON files according to data type and Facebook User ID.

Actors - Facebook.

Pre-conditions – There must be processed data.

Typical sequence of events – Acquire data -> Process data -> Save data into JSON Files**.**

Use case 4- Access Data

Goal – Access the data saved in JSON files.

Actors – Website, Timeline JS and FusionCharts.

Pre-conditions – There must be saved data.

Typical sequence of events – Provide data access**.**

4.3 Implementation

4.3.1 Server Module

4.3.1.1 Creating a Facebook Application

The first requirement to extract information from Facebook is to create a "Facebook Application". This requirement is necessary because a "Facebook Application" provides all the necessary tools to deeply integrate with the Facebook core, the Social Graph.

All Facebook users are allowed to create Facebook applications on the Facebook Developers page [14]. When the application is created, an App ID and an App Secret will be generated. These two keys will be needed to connect the created application to Facebook. In order to fully understand how applications work, it is important to understand the following concepts (only the key aspects to this project will be included):

Site URL: This is the URL of the website. For security reasons, Facebook will only redirect to this URL.

Permissions: Permissions define which information the user must provide to authenticate the application in development. The following picture shows this process.

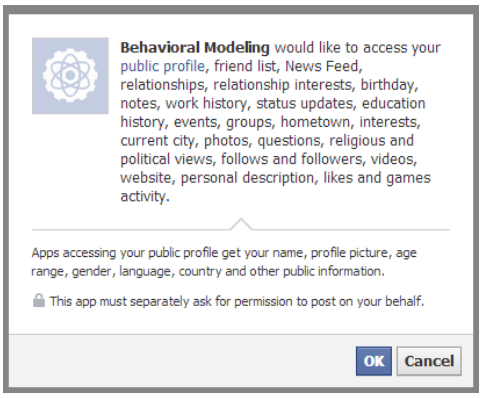

Figure 25: Facebook permissions request

Access Token: This token is created during the login flow and according to the permissions given by the user. This will be the key that will allow the application to extract user information from the Graph API.

```
graph.facebook.com/100000563031180?fields=posts&access_token=BAACEdEose0cBAJbPj5LTc5
ZATuSYjWY8TKneYHco7f8bzWaLhv4IIdeyZApDjIQSHEbqgvJ5CV08dhhYxIMGD5ENCx3eQRf6Dg7nb
HDnEjMyTZADTWKfstUvUedrt6lI4ZB9U3dmsM5haI36EPnSCLhnvombGFAoSQJte3nZAzF5o96XZCD
DgJeQdDHM5XYJxcIzd8JZC4yQ22jQ2ZBWAgwm
```

The previous textbox shows how a Graph API request looks like. In the next topic we will explain all the information flow and purpose, according to the final objective.

4.3.1.2 Extraction and treatment of information

As explained in the previous topic, an access token is required to get access to user information. The first step to have it is to create a “Facebook Application”. This access token is given during the authentication process. To create the authentication process, we used the Facebook PHP SDK. Facebook SDK works like a service, it will instantiate a Facebook objet with following methods [32]:

api() method can call a Facebook method or an query depending on the parameters passed in.

getAccessToken() get the current access token being used by the SDK. This may be a user access token or an application access token.

getApiSecret() returns the APP Secret being used by the SDK.

getAppID() returns the APP ID being used by the SDK.

getLoginStatusUrl() returns a URL based on the user's login status on Facebook. It is possible to get a different URL depending on whether the user is logged in, not connected, or logged out of Facebook.

getLoginUrl() returns a URL that, when clicked by the user on the client-side, will redirect the user to login to Facebook and authorize the referred application, if necessary. It will then redirect back to the application. If the user did not successfully log in, or did not authorize the application, the user will be redirected via an HTTP 302 redirect to the redirect_uri with error, error_reason, and error_description parameters in the URL.

getLogoutUrl() returns a URL that, when clicked by the user, will log them out of their Facebook session and then redirect them back to the application.

getUser() returns the Facebook User ID of the current user, or `0` if there is no logged-in user.

setAccessToken() sets the current access token being used by the SDK. This is useful if an access token was acquired via other means and would like the SDK to use the same token.

setApiSecret() sets the app secret that the SDK is currently using. The app secret is associated with the app id, so *setAppId()* should also be called if this method is used.

setAppID() sets the app ID that the SDK is currently using. To call this method is also necessary to call *setApiSecret()*.

It is important to keep in mind that this token is created for the application, according to the permission requested. Therefore, it is valid for all users that are signed to the application (authorized the application to access their profile). This is an important aspect since an access token can be acquired only during the login flow; however, if we simulate the administrator login, the given access token allows the application to have offline access to the other users' information. This will allow, for example, detecting profile changes and sending an alert to the user by email or even by SMS.

In this tool we used the access token to have access to 6 different types of information:

1. *Profile information* gives all the user profile information (name, e-mail, birthday, genre, location, education, relationship status, religion).
2. *Posts* return all the user activity.
3. *Status messages* return all the messages written on the user wall by himself links
4. *Likes* return all the pages liked by the user
5. *Links* return all the links posted on the user wall by himself
6. *Photos* return all the photos posted by the user

The information is stored in JSON files according to the type of information and user ID.

Before we mentioned that Facebook timeline aggregates all kinds of activity. Therefore, the first step in this website was to create a new timeline and perform a temporal analysis, separating the different types of activities. The objective of this timeline is to give an easier overview of the posts done by a user. To implement the timeline we used TimelineJS (see appendix A) and the final result is represented in the next figure.

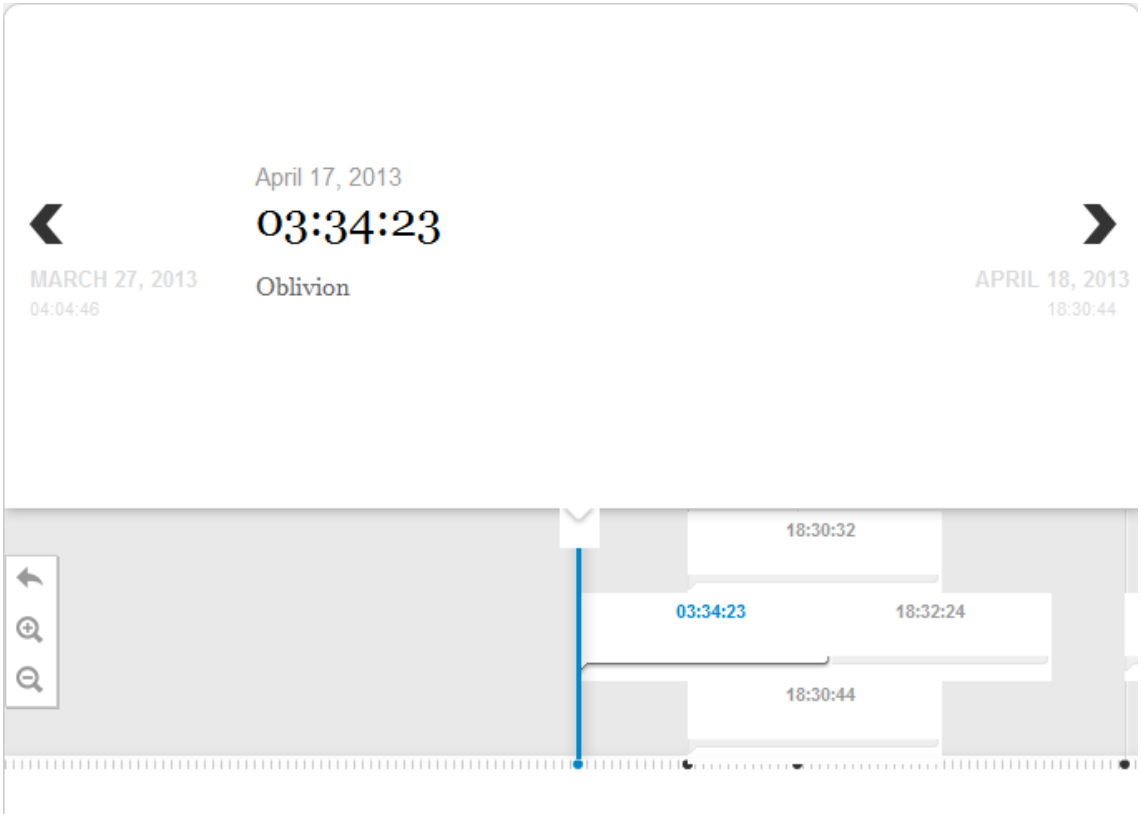

Figure 26: Timeline

Unlike Facebook, the user can easily check the exact date of a post and can also scroll all his activities and see if they are legit. However this tool was only added to simplify the interpretation of user behavior charts, which will be explained in the next topic.

4.3.1.3 Modeling strategies applied

As an user tool, this tool has to be simple enough to deliver its objective to all types of users. There are many strategies that can be applied to model user behavior; however, some of them require some knowledge or are hard to analyze. Therefore, it was important to create

a model to fulfill both of the website purposes, be a simple and understandable tool and at the same time give enough functionalities to detect compromised accounts.

The first tool implemented was a profile change detector. Each time the user logs into the website, the user profile is recorded and compared with the previous record. If the records are different, then the user will be notified and will have a shortcut option to Facebook profile section. The script behind the website is prepared to run offline, therefore it is possible to run it periodically, for example twice a day, and run the profile change detector. That way, it becomes possible to notify a user that his profile has been changed. Facebook also allows real time updates for applications however this tool will not be implemented. The reason why it was not implemented is because the website does not have any tool that depends on real time updates and since the script can run on a schedule defined by the administrator, it was decided not to implement this API extension.

After surpassing the profile challenge, the user can check a section with some profile data, including the last profile update date. After that the user can check information about his activity, comparing his average posts per month with the last month's number of posts or his average posts per week with last week number of posts. The next figure shows the result:

Figure 27: Statistics section

Looking into this example, let us now discuss a possible interpretation of these results. The first comparison that most users will do is between last month posts, average posts per month and last week posts, average posts per week. In this case, in both of them the recent activity is a slightly higher than the average activity, however the results are close, so it will be considered normal. By itself, this information cannot fulfill our purpose. Consequently we used Fusion Charts to integrate charts in the website. Charts are a great tool to represent behavior, therefore we will explain which charts were implemented and what they represent.

The first chart includes the number of posts by day. This does not show all posts, only shows around 1 month of activity. The objective of this chart is to show possible cases of spam. To explain the next chart we will use the same data from the previous examples.

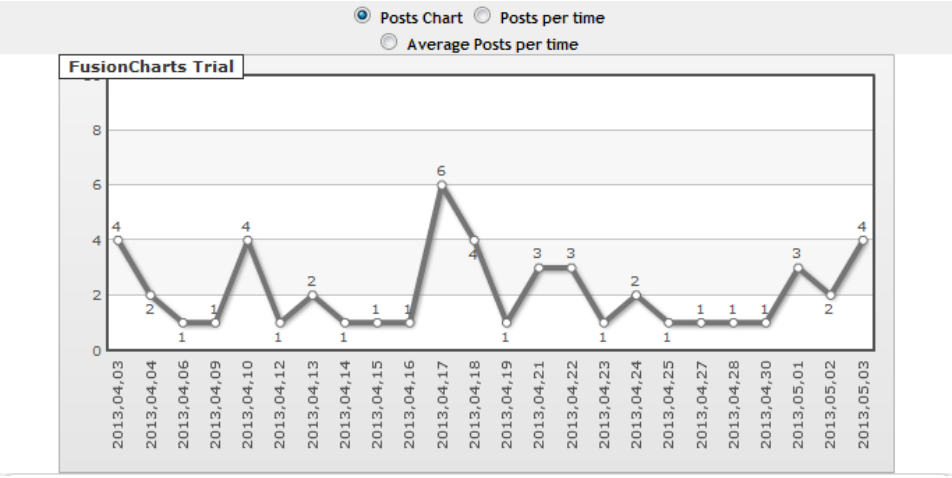

Figure 28: Distribution of posts per day

Taking a close look to figure 29, it can be seen that the average posts per day is 1,5 and the standard deviation is also 1,5. Checking the chart, we can see that the highest value is 6. This value is not included in the standard deviation however is not high enough to cause doubts. Anyway, the user can always check the timeline and see the motive of this amount of posts. Still, not all hacked accounts are used for spam. The hacker can also use the account for identity theft or to denigrate the user.

Figure 30 chart shows the distribution of posts during a period, t. The following equation represents the strategy applied. N_p represents the total number of posts and $P_{d,t}$ is the number of posts for each period t.

$$N(t) = \sum_{d=0}^{N_p} P_{d,t} \quad (1)$$

To show only the recent activity we divide the number of posts by 6 as represented in the next equation. As previously mentioned, N_p represents the number of posts. The algorithm reads the number of posts from the most recent to the oldest. Therefore, to represent the recent activity we just need to get a sample of posts. For example, if a user has 120 posts, dividing by 6 will give us a sample of 20 posts and since the algorithm reads the posts from the most recent to the oldest, these 20 posts will represent a sample of recent posts. In these charts, we do not take into account the interval of time, in other words, 20 posts can represent a month. That is the reason why we have another chart that represents the activity in the last 7 days.

$$N(t) = \sum_{d=0}^{N_p/6} P_{d,t}$$

The final result will be the following chart:

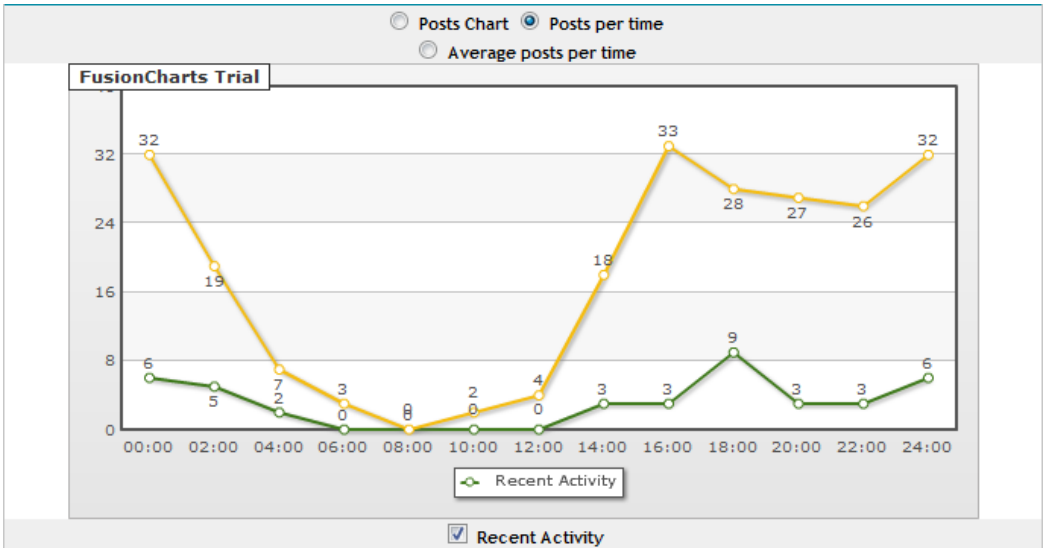

Figure 29: Distribution of posts by time

The yellow line represents the user behavior since January 2013 while the green line represents the most recent posts ($N_p/6$). As we can see in the yellow lines, the user has specific periods of activity. Between 4 am and 12 pm the user has almost no activity. The purpose of showing the recent activity is to show if this behavior has changed. As we can see, both lines have similar curves, which means that the user did not change his behavior. Following the same idea the third chart represents the average posts distribution for the same periods of time. It can be defined as follows

$$A(t) = \frac{1}{D}\sum_{d=0}^{d} P_{d,t} \tag{3}$$

in which *A(t)* represents the average posts for periods of time, *D* represents the number of days between the first and last posts and *d* represents the sample space. In this case *D=d*. To show only the activity in the last 7 days we apply the following equation:

$$A(t) = \frac{1}{7}\sum_{d=0}^{7} P_{d,t} \tag{4}$$

The result is shown in the following figure.

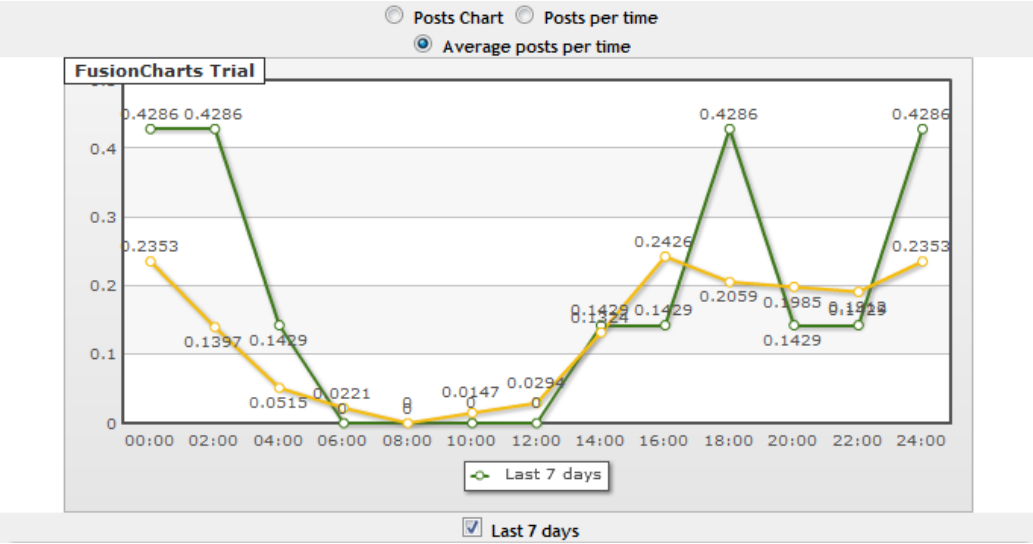

Figure 30: Average Distribution of posts by time

However, all these tools require time and interpretation from the users, which can result in a usability issue for many users. To solve this issue, an *Alerts* section was created. This section will run an algorithm that will track the user's behavior and investigate if there is any unusual behavior. Basically the algorithm is based on information extracted from the graph show in Figure 31 and computes high deviations between the average posts and the average posts for the last 7 days. For testing purposes, we considered 2*AveragePostPerDay as the threshold to create an alert. The alert will highlight the odd behavior showing when it happened. That way the user does not have to fully understand the user behavior charts, with the alert information he can check the timeline and verify if the behavior is legit.

In this section we only gave examples for posts however, we followed a similar strategy for Status, Likes, Photos and Links. However, the *alerts section* will only work for posts since posts represent all user activity, in other words, a like is a post, a new status is a post, a new photo is a post and a new link is a post. Therefore, if the account has unusual amount of likes, they will be represented in the posts and an alert will be set.

4.3.2 Client Module

The primary objective of this thesis was to get data from Social Networks and to analyze it by creating models that would help in the detection of possible Network Accounts intrusions. The best way to test our tool is to implement it in a real scenario, like a website. In this chapter we will show the walkthrough of the website most important functionalities.

This website was developed in PHP, a free server-side scripting language and a powerful tool to build dynamic and interactive web. PHP commands can be embedded directly into an HTML source document.

Each time a user logs into the website the user profile is compared with previous recorded. If they are different, the user will see an alert message on the screen.

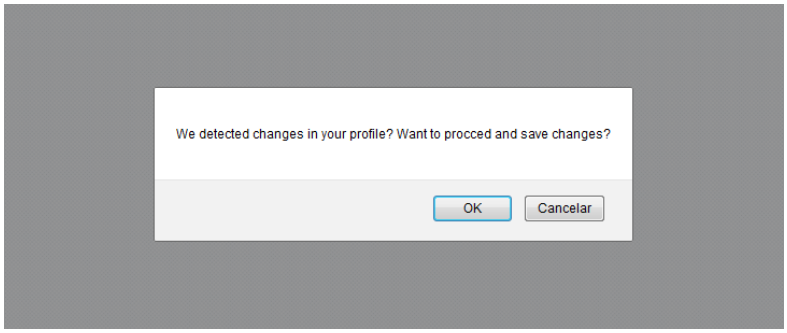

Figure 31: Profile checker alert box

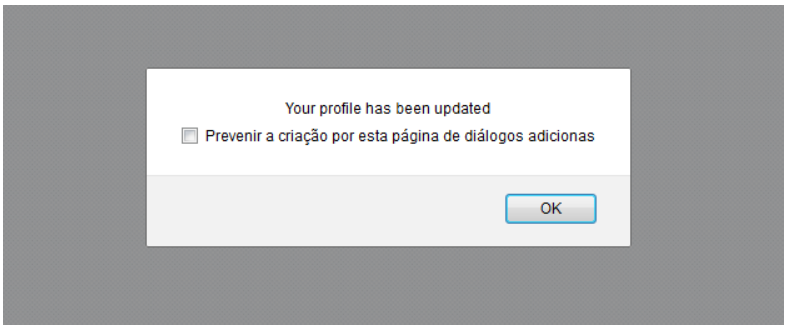

Figure 32: Profile checker alert box

If the user clicks “OK” the profile change will be recorded, if he clicks “Cancelar” a new window will be opened on Facebook user profile page. After that process the user will be redirected to the Home Page. The Home Page has 4 different components: the menu bar, the statistics field, the charts field and the timeline. The menu bar shows all the types of data provided and allows selecting one. The following image represents the menu bar.

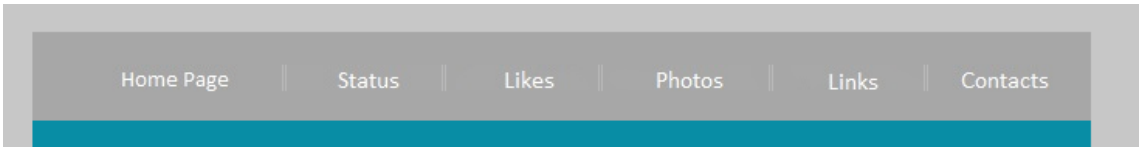

Figure 33: Menu bar

Since posts represent all user activity, these are shown on the Home Page. The other links will filter the information by Status post, Likes post, Photos post or Links posts. Therefore, the data represented on the statistics, charts and timeline will be related with the selected link.

Figure 34: Home page statistics section

The statistics field shows some important information about the user profile. Along with the user name, birthday and location, it is possible to check the Last profile update and also the user behavior by posts per day and posts per month. It also shows the Alert link (if the user has alerts the link will be red), the Check Events link and the Check Subscribed List link.

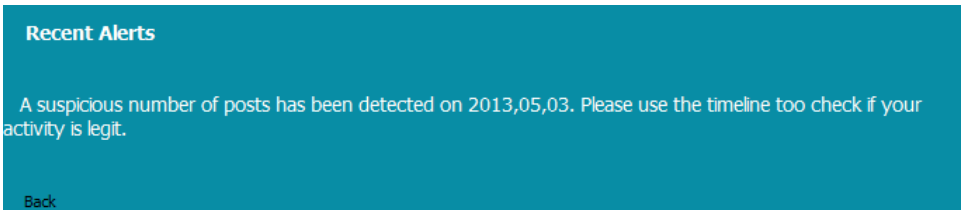

Figure 35: Alerts box

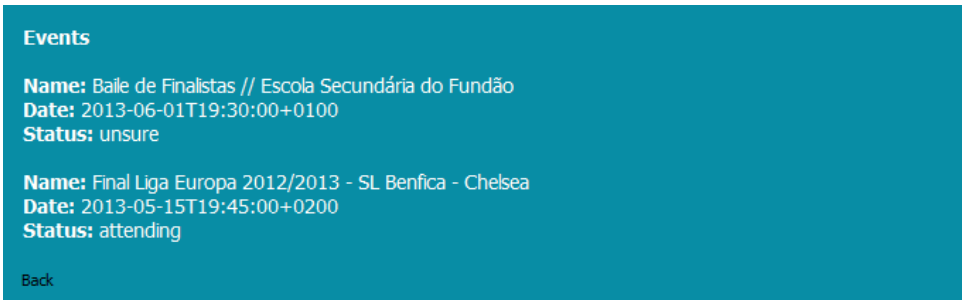

Figure 36: Check events box

Figure 37: Check subscribed list box

Figure 36 shows if the user has alerts. An alert is issued when a user has a high deviation from the usual behavior. Figure 37 shows the events replied by the user and Figure 38 shows the user subscribed list. The other components (charts and timeline) were already covered in the previous section.

4.4 Usability tests

Making a graphical interface is a process that requires the participation and evaluation of users. Usability tests are an excellent way to test the reliability of an interface. Performing test to users with different backgrounds and informatics skills can result in different results. Analyzing these results can give many answers and can also define the target audience for our tool.

To perform these tests the user will have to do a few tasks that represent possible scenarios. Meanwhile the observer collects data on the participant success and satisfaction. After the tests, the results will be provided to the developers so that the necessary changes can be performed. Following iterative methods for these tests can result in a better website. To perform the test, the participating users were asked to perform 5 tasks (see appendix B). These tasks will challenge the usability of the website main functionalities.

The users chosen to perform the tests had no special informatics skills, just regular users of social networks and a wide age distribution. The test was performed to 6 males and 4 females.

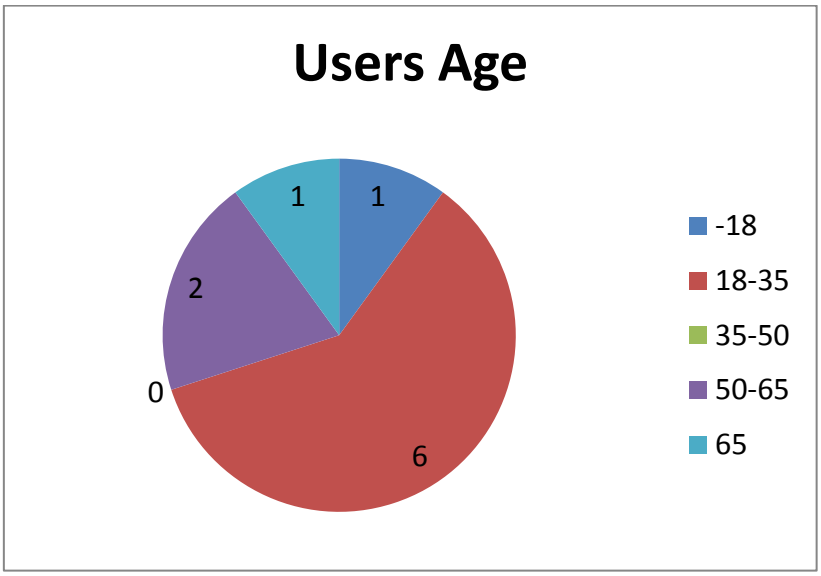

Figure 38: Users Age Chart

The tasks performed could be classified in the following level of difficulty:

- 1- Very Difficult
- 2- Difficult
- 3- Acceptable
- 4- Easy
- 5- Very Easy

The following figure shows the results of the 5 tasks performed. The results were better than the expected and our main goals to build a simple and effective tool were accomplished. Since the website is in English and not all tested users are familiar with this language, the only help given by the tests observer was related with that. The results of the Post-Task Questionnaire (appendix C) were extremely positive. Most of the users found the website visual design very attractive and easy to use. It was also highlighted the importance of the information provided.

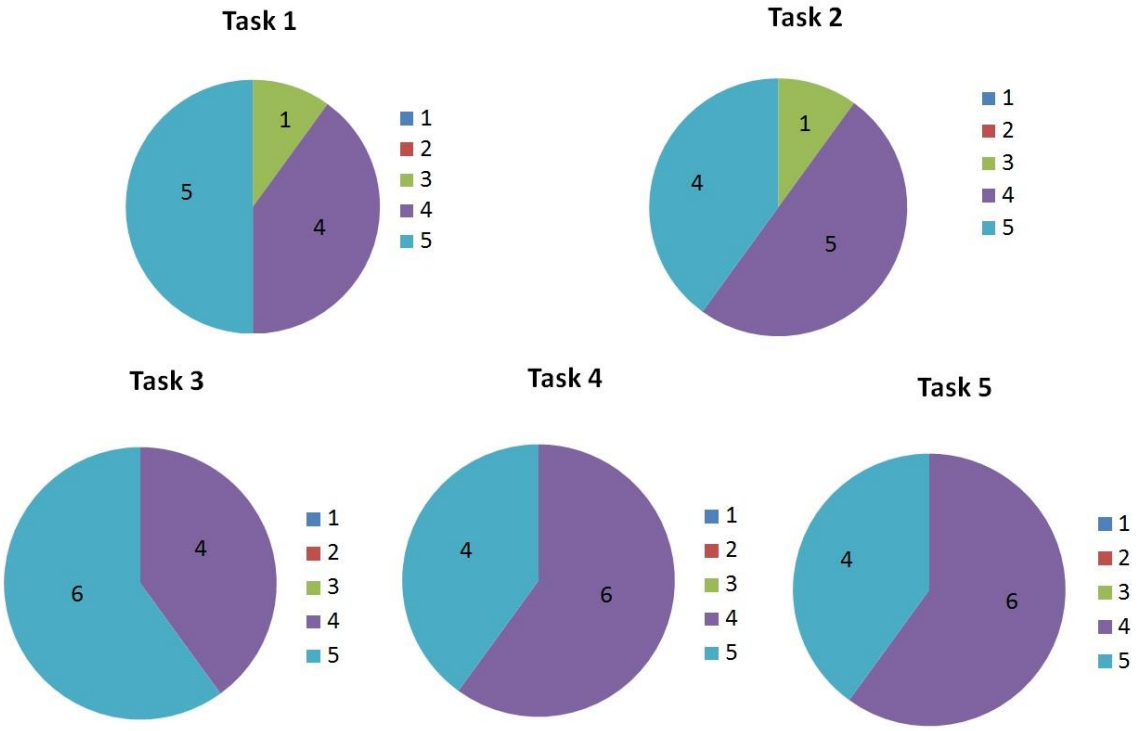

Figure 39: Tasks results

4.6 Conclusions

In chapter 3 we presented Facebook Security tools. It is a fact that the tool we proposed offers more information and can be more effective in some of the attack's phases. However, like other security tools, our model requires some informatics skills, some time and interpretation of the results. Most users will ignore tools like that; the only option was to create a similar module embedded in Facebook with real time notifications. However this way, Facebook notifications could be read by the attackers, letting them know that their odd behavior was being noticed. Doing an independent tool can diminish the target audience but improves the tool effectiveness.

Chapter 5 – Conclusion an Future Work

In this chapter, the accomplishments achieved with this project are identified and related issues that may be addressed as future work are identified.

5.1 Accomplishments

This dissertation was developed across distinct phases, with several outcomes:

- Identified the prior art and existing studies in this area. During that phase we concluded that the main concept of this dissertation was quite unique and quite limited information is available.
- Analysis of the most popular Social Networks (Facebook, Twitter, LinkedIn), their security threats, technologies and APIs. We chose Facebook for the implementation, since its API offers a lot of features compared with other Social Network APIs.
- Investigated, tested and took conclusions about Facebook security and privacy features. We concluded that existing Facebook privacy features are either limited or controversial and the most advanced features are currently "on hold".
- Specified the user stories, technical architecture and developed a website, connected to Facebook, to support end users identifying security issues with their account and understanding their typical behavior. We concluded that security tools based on user behavior can be an effective addition to other conventional security methods.
- Created an alert system to reduce usability issues. This alert system defines a threshold based on user average behavior and automatically calculates deviations between the average behavior and recent behavior. If the deviation is higher than the threshold an alert will be issued to the user.
- Performed tests and analyzed the results, using recommended industry practices. The users tested finished all the tasks given with success and gave good feedback to our tool.

This dissertation also gave origin to the paper "Detecting Social-Network Bots Based on Multiscale Behavioral Analysis". This paper considers strategies to detect behavior differences between bots and humans.

In conclusion, all the objectives were achieved or exceeded, even that the developed tool has potential for several new features that will be discussed in the “Future Work” section.

5.2 Issues

Since there are no similar projects, the first issue was the website design. Getting data from Facebook is quite simple however using that data to represent the user behavior and with that give the necessary tools to detect hacked accounts, is not simple. The website was made from scratch, with no other tools as support. That was a concern because the website might work in the developer vision, but not for the typical Social Network user. The first contact with user was on the usability test and the results were quite enthusiastic. Another issue is our tool dependency from Facebook. The information is limited by what Facebook Graph API gives. The retrieval of users’ security and privacy settings are not available and this data could improve our tool.

All the other issues are related with the user behavior, not exactly with the development. Considering the experience gained in the analysis of Facebook security features, where we learned that most users ignore security tools specially the preventing tools (in which we include ours), it might be needed to increase awareness about security threats within the target audience, to have higher acceptance of these tools.

5.3 Future Work

The best way to catch user attention is giving real-time updates. Nowadays’ mobile networks are increasing along with the use of smartphones which means that a lot of Social Networks users have Internet access on their mobile phones.

Considering that perspective, it is possible to notify users in real time. The script implemented for our website allows offline access to users’ information. If the script is executed several times per day or if we include Facebook real-time update feature we can notify users by SMS or email. The warning can include detailed information and also images that will help to clarify what changed in the user usual behavior. On the other hand, our alert system can be improved. In our tool it is established a threshold based on user posts behavior, therefore it is only considered deviations from average posts per day. There are other complex strategies to detect compromised accounts, thus improving the alert algorithms will reduce usability issues and will also improve the effectiveness of the global tool.

Another possibility for future work could be the integration of a fake profile detector. In chapter 3 we mentioned the possibility of creating a tool that based on specific attributes could detect fake accounts in a user’s circle of friends.

The first step to implement this tool is to acquire the necessary information. The Graph API can returns the list of friend requests however a user cannot see the profile that sent the request before accepting it - the only option is to check the profile after accepting the request. That way the script will have to run the user list of friends.

In our tool we set the Facebook application to ask for all users permissions so we can access to all user data. To implement this tool and reutilize the same script used for the website we have to change the permissions and ask also for friends' permissions. This way the access token provided for the application will also allow the retrieval of user friends' data. After having a valid access token we have to define which attributes we will take into account to detect a fake account. After investigation and following the topics exposed in the section 3.2.2, we came up with the following attributes:

- **Birthday date (A_1)** - Fake accounts are known to use birthday dates likes 1/1/XX or 31/12/XX.
- **Genre (A_2)** - 97% of fake profiles are women.
- **Interested in (A_3)** - 60% of fake profiles are interested in both men and women.
- **Numbers of friends (A_4)** – Fake accounts are known to have a lot of friends (more than 726).
- **Number of photos (A_5)** – Fake accounts usually have only 1 or 2 profile photos. It is also interesting to check the number of albums and see if there are any mobile phone uploads album. Usually fake accounts have only one album and no mobile photo uploads.
- **Status update (A_6)** – Most fake accounts do not update their status regularly.
- **Interests (likes) (A_7)** – Fake accounts have a really low number of interests. Usually less than 3 per profile.
- **Mobile Contact number (A_8)** – most real profiles do not share mobile contact number in their profiles. Usually fake profiles do.
- **Family (A_9)** – Fake profile do not update family relations.
- **Relationship Status (A_{10})** – Most fake profiles do not have relationship status.

Combining all these attributes together, allows to generate a probability of a profile being fake. However there are a lot of factors to be considered:

- Some real profiles will have some attributes typical from fake profiles.
- There are birthday dates with 1/1/XX.
- There are users interested in both men and women.
- There are users with only one profile photo, etc.
- There users with no family relations.
- There are users that do not update their status regularly.

Basically every attribute can contribute to create a false positive case. However it is important to consider that some of these attributes contrast with each other. Some examples:

- If a profile has 800 friends, birthday date 1/1/XX , only one profile photo and the status is not regularly updated it has a high probability of being a fake profile.
- If a profile has 50 friends, only one profile photo and the status is not regularly updated this profile will have low probability of being fake - probably it is not a regular user of Facebook.

- If a profile has 800 friends, it is unlikely to have no family relations or no interests.

Therefore to create an effective detection it is fundamental to consider these relations between attributes. That way attributes should have different values according to the relations they have.

Considering ten attributes the following formula will serve has base:

$$P_T = \frac{w_1.A_1 + w_2.A_2 + w_3.A_3 + w_4.A_4 + w_5.A_5 + w_6.A_6 + w_7.A_7 + w_8.A_8 + w_9.A_9 + w_{10}.A_{10}}{10} \quad (5)$$

P_T is the probability of a profile being fake, A_1 to A_{10} represent the attributes considered and w represents the weight of each attribute. The value of the sum of all attributes has to be equal or less than 10 (number of attributes and consequently number of possible cases). Status updates (A_6) and number of friends (A_4) will be considered the most relevant attributes based on the investigation done. The value of w_4 and w_6 are 2,5 for each. Therefore if the profile has more than 600 friends and no wall posts (status, links or photos) for a month, $A_4 + A_6 = 5$ (50% chance of being a fake profile). The other attributes will have a weight of 0,625. For all attributes (A_1 to A_{10}) the result is 1 if they are true and 0 if they are false. This formula is relatively simple because the relations between attributes can have more variables. Testing the author of this thesis profile the result would be:

- Name: Francisco Brito;
- w_1A_1 = 0, w_2A_2 = 0, w_3A_3 = 0, w_4A_4 = 2,5, w_5A_5 = 0, w_6A_6 = 0, w_7A_7 = 0,w_8 A_8 = 0,w_9 A_9 = 0, $w_{10}A_{10}$ = 0;

For this profile $P_T = 0{,}25$ therefore this profile has 25% chances of being fake according to the formula proposed. Only the attribute A_5 had a true value. Thus based on this formula it is unlikely that this profile is fake. Unfortunately after some investigation we were unable to find a fake account and test it. Most fake accounts when discovered are removed.

In conclusion this formula can offer already some interesting results however only with the implementation and further tests will be possible to calibrate it and offer better results. The highest improvement that has to be done it is in the weight of attributes. We set the weights of values based on their importance however we do not correlate them with the results. On the profile tested the weight of the A_4 is 2,5 however since all the other result are false we could reduce the value of w_4. With this approach we believe the results will be more trustworthy. On the other hand there are other solutions that can improve fake profiles detection. The article "Detecting Social-Network Bots Based on Multiscale Behavioral Analysis", that is related to this thesis, suggests a different approach based on the comparison of user and bot behaviors through time.

References

[1] Image source: http://damodablog.blogspot.pt/2010/10/voce-sabe-o-que-sao-redes-sociais.html.Accessed in 2013-01-17.

[2] Wikipedia 2013. Social Networking Service. Accessed in 2013-01-17 in http://en.wikipedia.org/wiki/Social_networking_service.

[3] Gonçalo Gabriel Martins Ribeiro. Redes Socais. Monography, Recuperação de informação course.

[4] The New York Times. Facebook Privacy: A Bewildering Tangle of Options. Accessed in 2013-05-23 in http://www.nytimes.com/interactive/2010/05/12/business/facebook-privacy.html?_r=0.

[5] Courteney Palis. Facebook Privacy Options Ignored by Millions of users. Accessed in 2013-05-22 in http://www.huffingtonpost.com/2012/05/03/facebook-privacy-consumer-reports_n_1473920.html.

[6] Marketo. How Facebook Graph Search Affects Your Privacy. Accessed in 2013-05-22 in http://blog.marketo.com/blog/2013/05/how-facebook-graph-search-affects-your-privacy.html

[7] Andrew Couts Facebook still has 76,9 million “fake” users. Accessed in 2013-05-23 in http://www.digitaltrends.com/social-media/facebook-spammers-fake-users/.

[8] Dale Pearson. The Risks of Posting in Social Networks. Accessed in 2013-02-13 in http://www.subliminalhacking.net/2012/10/17/the-risks-of-posting-in-social-networks-by-trend-micro/.

[9] ScamWatch. Phishing scams on social networking sites. Accessed in 2013-05-23 in http://www.scamwatch.gov.au/content/index.phtml/itemId/762345.

[10] Softpedia 2013. 54% of Social Networking Customers Encountered Phishing Attempts. Accessed in 2013-05-23 in http://news.softpedia.com/news/54-of-Social-Networking-Customers-Encountered-Phishing-Attempts-227446.shtml.

[11] SANS Intitute. A Proactive Defence to Social Engineering. Accessed in 2013-05-23 in http://www.sans.org/reading_room/whitepapers/engineering/proactive-defence-social-engineering_511.

[12] Image source: http://hackingpedia.wordpress.com/. Accessed in 2013-01-23.

[13] Bob Sullivan. Facebook imposter scam a growing concern. Accessed in 2013-05-11 in http://redtape.nbcnews.com/_news/2009/10/06/6345712-facebook-imposter-scam-a-growing-concern?lite.

[14] Facebook Developers 2013. Graph API. Accessed in 2013-03-02 in https://developers.facebook.com/docs/reference/api/.

[15] Facebook Developers 2013. Getting Started: Graph API. Accessed in 2013-03-02 in https://developers.facebook.com/docs/getting-started/graphapi/.

[16] Image source: http://youthspeak.blogspot.pt/2010/12/future-of-social-networking-and-idea.html. Accessed in 2013-03-10.

[17] Bill Burke. Introduction to Rest. Accessed in 2013-05-11 in http://java.dzone.com/articles/intro-rest.

[18] Image source: http://hinchcliffe.org/default.aspx. Accessed in 2013-05-12.

[19] Twitter 2013. The Streaming APIs. Accessed in 2013-03-11 in https://dev.twitter.com/docs/streaming-apis.

[20] Image Source: https://dev.twitter.com/docs/streaming-apis. Accessed in 2013-03-11.

[21] Image source: https://developers.google.com/+/api/. Accessed in 2013-03-16.

[22] Facebook 2013. Data Use Policy. Accessed in 2013-03-16 in https://www.facebook.com/about/privacy.

[23] Image source: http://msdn.microsoft.com/en-us/library/dd303381.aspx. Accessed in 2013-05-11.

[24] Alex Rice. A continued commitment to Security. Accessed in 2013-04-07 in https://www.facebook.com/blog/blog.php?post=486790652130.

[25] Image source: https://www.facebook.com/blog/blog.php?post=486790652130. Accessed in 2013-04-07.

[26] Lei Jin, Hassan Takabi and James B.D. Joshi. Towards Active Detection of Identify Clone Attack on Social Networks.

[27] Paul Chaney. Using Facebook's Insights Analytics to Expand Your Audience. Accessed in 2013-04-10 in http://www.practicalecommerce.com/articles/3553-Using-Facebook-s-Insights-Analytics-to-Expand-Your-Audience.

[28] Image source: http://www.practicalecommerce.com/articles/3553-Using-Facebook-s-Insights-Analytics-to-Expand-Your-Audience. Accessed in 2013-04-10.

[29] Facebook 2013. Access Tokens. Accessed in 2013-05-11 in https://developers.facebook.com/docs/facebook-login/access-tokens/.

[30] Usability 2013. Usability basics. Accessed 2013-05-02 in http://www.usability.gov/basics/.

[31] Usability 2013. Guidelines. Accessed in 2013-05-02 in http://www.usability.gov/guidelines/index.html.

[32] Facebook 2013. Facebook PHP SDK. Accessed in 2013-05-11 in https://developers.facebook.com/docs/php/gettingstarted/.

Appendix A – Timeline JS form Example


```
"timeline": {
"headline": "Facebook Feed",
"type": "default",
"text": "Check your Facebok wall timeline",
"startDate": "2010,04,16",
"date": [
    {
        "startDate": "2013,05,05",
        "endDate": "2013,05,05",
        "headline": "15:54:58",
        "text": "",
        "asset": {
            "media": "https:\/\/fbcdn-sphotos-e-a.akamaihd.net\/hphotos-ak-ash3\/s720x720\/575572_612459235449505_1405368533_n.jpg"
            "credit": "",
            "caption": ""
        }
    },
    {
        "startDate": "2013,05,05",
        "endDate": "2013,05,05",
        "headline": "15:52:50",
        "text": "",
        "asset": {
            "media": "https:\/\/fbcdn-sphotos-b-a.akamaihd.net\/hphotos-ak-ash3\/s720x720\/480357_612458528782909_383183334_n.jpg",
            "credit": "",
            "caption": ""
        }
    },
```


Figure 40: Timeline JS form

Appendix B – User test form

User test form / Teste com utilizadores
Website

Task 1 **Tarefa 1**	***Ver a média de número de posts por dia e a respectivo desvio padrão.*** Muito Difícil 1 2 3 4 5 Muito Fácil
Task 2 **Tarefa 2**	***Ver a lista de eventos respondidos pelo utilizador.*** Muito Difícil 1 2 3 4 5 Muito Fácil
Task 3 **Tarefa 3**	***Verificar se existem alertas no comportamento recente do utilizador.*** Muito Difícil 1 2 3 4 5 Muito Fácil
Task 4 **Tarefa 4**	***Verificar se existem alterações de comportamento utilizando o gráfico Average Posts per time.*** Muito Difícil 1 2 3 4 5 Muito Fácil
Task 5 **Tarefa 5**	***Ver a data em que foram efectuados mais likes e de seguida vê-los na timeline.*** Muito Difícil 1 2 3 4 5 Muito Fácil

Observações :

_____

_____

_____

Figure 41: User test form

Appendix C – Post-task Questionnaire form

Post-task Questionnaire / Questionário Pós - Tarefa

1. Personal Info / Dados pessoais

Genre / Género: ☐Female/Feminino ☐ Male/Masculino

Age / Idade: _____

2. General opinion about the system / Opinião geral sobre o sistema

After using the system and given its final assessment, put a tick in the circle that best reflects your opinion regarding the use of the system. If you feel that these measurements are not applicable, NA choice.

Após a utilização do sistema e tendo em conta a sua avaliação final, assinale com uma cruz o círculo que melhor reflecte a sua opinião em relação à utilização do sistema. Caso considere que estas quantificações não são aplicáveis, escolha NA.

2.1. Opinion about the system / Opinião sobre a utilização do sistema

It's easy to navigate in the system / É fácil orientar-me no sistema	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
I easily find what I'm looking for / Encontro facilmente o que procuro no sistema	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The system is slow / O sistema é lento	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The system is pleasant to use / O sistema é agradável de utilizar	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
This system has some annoying characteristics / Este sistema tem algumas características irritantes	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
There is consistency in layout and content presented / Existe consistência na disposição e nos conteúdos apresentados	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
I need help on some features / Sinto necessidade de ajuda em algumas funcionalidades	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
Use of the system requires deeper knowledge or previous experience. / A utilização do sistema requer conhecimentos mais aprofundados ou experiência anterior.	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The size of the characters on the screen makes them easy to read / O tamanho dos caracteres no ecrã torna-os fáceis de ler	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The most important information is highlighted / A informação mais importante possui um bom destaque	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The amount of information that can be displayed per screen is adequate / A quantidade de informação que pode ser apresentada por ecrã é adequada	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The information disposal that can be displayed per screen is adequate / A disposição da informação que pode ser apresentada por ecrã é adequada	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The displayed icons are intuitive / Os ícones apresentados são intuitivos	Disagree/ Discordo	○○○○○	Agree / Concordo	NA
The visual design is attractive / O aspecto gráfico é atractivo	Disagree/ Discordo	○○○○○	Agree / Concordo	NA

Select, from the following, the two features that most pleased you during the system usage / Escolha, de entre as seguintes características, as duas que mais o/a agradaram aquando da utilização do sistema:

☐Ease of use / Facilidade de utilização
☐User interaction / Interacção com utilizadores
☐Information displayed / Informação Disponibilizada
☐Information disposal / Disposição da Informação
☐Menu arrangement / Disposição dos Menus
☐Graphical design / Aspecto Gráfico
☐Colours used / Cores Utilizadas
☐Buttons / Botões
☐Other / Outra _____

3. Final comments / Comentários finais

_____
_____
_____

THE END / FIM

Thank you for your collaboration / Muito obrigada pela sua colaboração

Figure 42: Post-task Questionnaire form