Radboud Universiteit Nijmegen

Faculty of science
Information Science

Academic year 2012-2013
26rd of March 2013

Master Thesis

Profiler:
Deriving a digital profile from open source information

Author:	Joost Hendricksen
Email:	J.Hendricksen@student.ru.nl
Student number:	4047702
Supervisor:	Theo van der Weide
Graduation number:	180 IK

ABSTRACT

This thesis describes the research that has been done on the possibilities to use Open Source Intelligence (OSINT) in criminal profiling.

This is realised through the following phases:

i) Literature study to determine the state of the art of OSINT in general, processing suspect information, profiling the suspect and an analysis of which information can be used in a criminal investigation;
ii) Requirements analysis, to determine the requirements for the model and prototype application;
iii) Model implementation in prototype; and
iv) Final model development, changing the initial model according to the shortcomings determined in the previous phases.

This process resulted in an OSINT model that supports data analysts in semi-automatic gathering of information. This generic model is tailored to OSINT profiling, although it can be used for digital profiling applications in general. The initial model is implemented in a prototype application. The prototype supports searching for suspects on Facebook, Twitter, LinkedIn and Google. It retrieves and extracts data from these distinct sources, gathers it in an integral relational database. This process results in an aggregated profile of a certain individual.

In the second iteration of model development we correct the shortcomings of the initial model. We discovered that techniques from enterprise architecture; a data-warehouse architecture combined with a rule-based expert system contribute to the flexibility of the model. It also contributes to the transfer of knowledge of the domain experts to the model.

We end with a discussion and conclude that although the model developed is valuable for OSINT and helps in terms of both quality and time, the automation of OSINT based profiling is still in its early years. Further, we conclude that OSINT and in particular profiling is an important emerging application domain for information sciences, as it requires input from various stakeholders next to the analysis and engineering of information systems.

Table of Contents

1 INTRODUCTION

This master thesis reports on the research that has been done on the topic of digital profiling during my internship at TNO in Delft. At TNO I was part of the Media and Network Services (MNS) department.

1.1 Context

Over the last decade the Internet has become an important communication platform in the Western world. Due to the emergence of mobile broadband connections, smartphones and tablets people spent more time online. Encouraged by Online Social Networks (OSNs), the ease of sharing information on the Internet has become an extension of people's lives. Anything can be shared in communities, weblogs, social networks and forums, 24/7. While the majority of users use communities to share experiences, people with wrong intentions use those platforms to exploit criminal or illegal activities.

Because of the growth of shared information, digital criminal investigation becomes an important part to the field of criminal investigation. Social media is obviously a valuable source of information. Law enforcement agencies are interested in utilizing this information to contribute in criminal prosecutions. As part of exploring the possibilities of valuable open information sources this research focuses on investigative profiling of an individual. This process has been part of classical forensic research for years. Finding clues and social information about an individual may eventually contribute to the prosecution of a suspect. In this research we explore ways to apply this technique to open information sources.

1.2 Relevance

Because we leave traces about our lives on the Internet, law enforcement want to have access to this information in a convenient manner. At this time a data analyst manually searches for personal data on open information sources on the Internet. A data analyst uses different specialised public search engines and information sources to supplement a user profile.

Because the process of digital profiling is time consuming and prone to errors law enforcement agencies are looking for an instrument to assist them in their job. Due to the huge volume of available online sources, it is necessary to assist the data analyst with an application that filters important information. By applying information retrieval and web mining techniques to the domain of criminal investigations this process can be improved. We want to support the data analyst without taking away the human component and its analysing strengths. We will present a model that partly automates the process of online profiling and utilises the human input.

1.3 Research question

The field of (digital) forensic science is widespread; therefore it is virtually impossible for researchers to have enough knowledge of social media and techniques in the field of information science to find the right and most complete information about a certain individual.

For that reason forensics would like to be able to use this valuable source of information that leads to the following research question:

> *Can we provide an Open Source Intelligence model to support a data analyst in the process of digital profiling?*

To answer this question the following sub-questions need to be answered:

> *What is open information and can it be used in court?*
>
> *What is the current state of digital profiling using Open Source Intelligence?*
>
> *What are the requirements for a prototype application to assist data analysts?*
>
> *Can we provide an adaptive model to support digital profiling using different sources?*

1.4 Method

To answer the main research question, the sub questions have to be answered first. To be able to do this, existing literature as well as input from our stakeholders and domain experts will be used. To fit this thesis into the current state of research references to literature will be used throughout this thesis. Literature references can be found in the 'Bibliography' chapter.

We will introduce online information sources, Open Source Intelligence (OSINT) and we will outline the possibilities of using data from this source from a law as well as a privacy perspective. After that we will evaluate existing OSINT models. We will explain our view on searching and gathering personal information and constructing an online profile out of this information.

To provide data analysts in a forensic investigation with the right instruments to support the process of digital profiling, we will do requirements analysis for a prototype application. After the need of the stakeholders is defined we are able to investigate which techniques from the field of information science can be applied in our model to benefit the process of online user profiling.

The results will be implemented in a model that will be developed through a process of requirements analysis and feedback on the prototype. As a proof of concept and a starting point for a usable system this theoretical model will be (partially) integrated in a prototype application.

1.5 Results

The end result of this research is a theoretical model that is able to assist in searching for personal information about a certain individual and construct a user profile out of scents of information collected from open sources on the Internet. This model will be (partially) implemented in a prototype application as a proof of concept.

1.6 Document structure

We will start by evaluating open information sources and existing OSINT models in chapter 2 using literature. We will also verify the usefulness of this information in a criminal investigation. In chapter 3 we will do a requirement analysis to determine the problem and the requirements for our model. After that we will present our first model as a prototype application in chapter 4. Chapter 5 reports on the final theoretical model. In chapter 6 we will validate our research process. In chapter 7 we will conclude our report by summarizing the conclusions and suggest future work on this topic.

2 ONLINE INFORMATION

2.1 Finding online information

The most common way to access information on the Internet is through web search engines. To get an estimation of how much of the Internet is accessible through web search engines we will introduce the way search engines work; how documents are found on the web and made searchable. We will conclude with an estimation of the ratio of documents that can or cannot be found using web search engines.

In the early days of the Internet, the World Wide Web consisted of static HTML pages that were linked to one another through hyperlinks, clickable text elements that lead to another web page. The architecture of search engines is based on that structure with the assumption that web pages contain links and have other pages linked to them. A web crawler retrieves all pages from an initial list of URL's; the indexer parses the HTML pages and creates a set of word occurrences that typically contain the location of the word in the document, an approximation of the font size and capitalisation to determine importance. The indexer also extracts hyperlinks and their anchors for link analysis and as input for the crawler (Brin & Page, 1998). This process and classification of attributes is altered and improved over the years but is still the basis for a search engine.

The word occurrences are transformed into inverted indexes; mappings from the word itself to the location of the word in the document. Out of the set of inverted indexes for a certain document a frequency distribution of words is constructed; a dictionary of all words in a document and the times they occur. As a final step this dictionary is transformed into a normalised feature vector which enables a search engine to compare different documents, independent of the size of the document. If a search query is executed on a web search engine, the query is transformed to a normalised feature vector and compared to document feature vectors on similarity (Manning, Raghavan, & Schütze, 2008). The results are presented to the user, ordered by a combination of page ranking (e.g. PageRank (Page, Brin, Motwani, & Winograd, 1999)) and document relevance based on the similarity.

Due to the advent of databases and functionality in web servers to serve dynamic pages (PHP and ASP) large data producers and new Internet-based firms choose to serve their information on the web (Bergman, 2001). Information in database-driven dynamic websites is often found and accessed through query forms instead of through static URL's. Therefore, search engines are not able to crawl the underlying databases and the data remains hidden from users, this part of the Internet is often referred to as hidden web or deep web.

Research has been done on the size of the un-indexed web but there is no single answer because it is hard to reliably measure it. In (Bergman, 2001) a set of ten database search websites was used to identify sources of deep web, after that they tried to find the same sources using surface web search engines and made an estimation of the number of deep web sites. The conclusion was that in the year 2000 the size of the deep web was 400 to 550 times larger than the surface web and faster growing as well. In (He, Patel, Zhang, & Chang, 2007) another approach was used; they randomly downloaded the

content of 1.000.000 IP addresses over the web and crawled those pages for a search query form to determine whether or not the site was database driven and therefore deep-web. They concluded that the deep-web consists of 307,000 sites, 450,000 databases, and 1,258,000 interfaces and is rapidly expanding. We want to remark that one particular IP-address can serve many websites based on the URL (Virtual Hosts), those websites are not covered in this method.

Both (Bergman, 2001) and (He, Patel, Zhang, & Chang, 2007) use the same qualification of a 'deep web' website, a website containing at least one search query form. We would like to specify this qualification by extending it with the condition that search result pages obtained through this search query form may not be found using regular (shallow) web search engines. With this condition we exclude websites that have a search query form but are fully indexed on popular search engines using sitemaps (Schonfeld & Shivakumar, 2009).

An important source for social activity and personal information are social media networks, they can be considered deep web according to our definition. Therefore, we cannot rely on search engines only to find user profiles and we will have to use other approaches to extract information from those media.

2.2 Using online information in criminal investigations

The role of the Dutch police on the Internet is currently topic of discussion, the investigative powers of the police are described in Article 2 of the Police Act 1993. These articles are twenty years old and describe the powers of police officers in the physical world but do also apply to the digital world, though they are not designed for that application. The law describes the police surveillance powers when investigating a criminal suspect; these powers are limited by a trade-off between the privacy of the suspect and the seriousness of the offense.

This definition differs in practice between the physical and the digital world, where privacy in the latter case is taken less strict because the user has placed the content itself and is ought to know this information is in the public domain. The existing system for digital investigation for governmental organisations, the Internet Research Network (iRN), is already widely used by numerous agencies, from the police forces to the immigration and naturalisation service. A total of 700 + workstations and + / - 4500 users (Verduurzaming iRN/iColumbo, 2011) shows digital forensic instruments are valuable to these organisations . The use of this system for tracking purposes is permitted for research on individuals but may not be used for the systematic mapping of social networks. Public information and information that is accessible only through user registration are allowed to be used by the police for 'digital surveillance' purposes, even if this involves creating a fake profile (Oerlemans & Koops, 2012).

Over the last three years we see a rise in the number court decisions in the Netherlands using social media, it has been used directly by a page printout or through witness statements. Figure 1 shows the number of court decisions that used OSN information. The source of this data is Rechtspraak.nl, the official website of Dutch courts and tribunals, this website publishes all Dutch court decisions online.

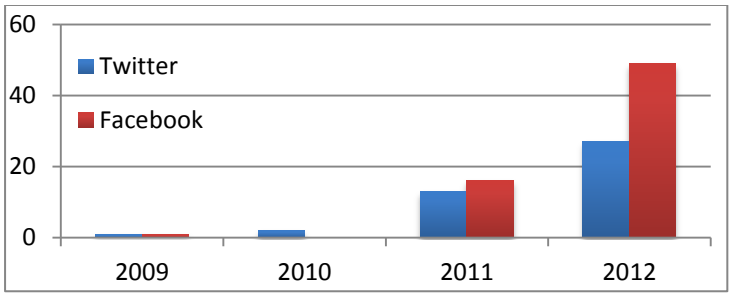


FIGURE 1 NUMBER OF COURT DECISIONS USING ONLINE SOCIAL NETWORKS. SOURCE: RECHTSPRAAK.NL

In such publications, privacy of suspects and convicts should be ensured and personal information should be anonymous. However, publications sometimes contain full social media messages that could be traced back to a certain profile or individual (See: Appendix IV: Privacy implication example).

2.3 Open Source Intelligence

Signal Intelligence (SIGINT) and Human Intelligence (HUMINT) are the most important forms of intelligence, forms that both derive intelligence from classified domains. Open Source Intelligence (OSINT) was introduced in 2006 as the topic of using public information as a source for intelligence. OSINT is defined by the United States government as "intelligence that is produced from publicly available information and is collected, exploited, and disseminated in a timely manner to an appropriate audience for the purpose of addressing a specific intelligence requirement." (National Defense Authorization Act for Fiscal Year 2006, 2006).

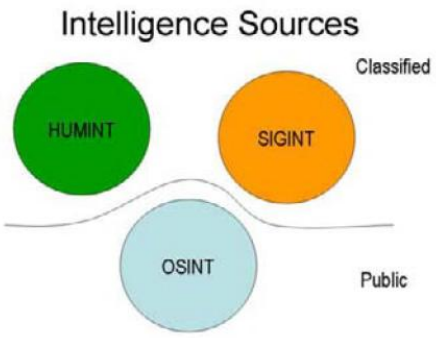


FIGURE 2 DIFFERENT SOURCES OF INTELLIGENCE (BEST, 2008)

As Figure 2 shows, a difference between OSINT and the other sources of intelligence is that it derives intelligence from the public instead of the classified domain. One could say the horizontal axis describes the amount of human labour involved in the type of intelligence. HUMINT derives intelligence from human sources, which is a laborious job since there are techniques like interviews or undercover operations involved. On the other side, SIGINT derives intelligence from structured signal sources like telephone-taps and the like, which is less laborious. OSINT is the in between, the data on Online Social Networks (OSNs) is structured but human input is required to classify the resulting profiles.

The core of this research is to assess whether or not OSINT can improve the process of digital profiling. Focus will be on designing a tool and theoretical model to semi-automate this process. The model should be using valuable human input of a data analyst together with information retrieval techniques to improve the quality of this aspect of criminal investigation.

Existing models (state of the art)

Several models for different applications of OSINT have been proposed over the last decade. In this paragraph we will give an overview of those models and an assessment whether or not they are applicable to our research.

Pouchard, Dobson and Trien (2009) propose two models and two prototype tools that use two different sources: the Internet in general and the DNI Open Source Center where the latter is an United States government intelligence service that aggregates open data sources (Central Intelligence Agency, 2005). The first model provides functionality to collect data from open sources, saving it in a local database and focuses on the visualisation of data.

The second tool stores and processes open source data and is able to extract metadata: topic, city and geographical coordinates. It implements the SeRQL query language with the RDF repository to compose search queries. The search results will be analysed by a named entity recogniser and be stored in a local database. The focus of the tool lies on the ease of use of the interface, the data is provided by the DNI Open Source Center and web search engines. The paper does not validate the results generated by the tool and proposes future work on the named entity recogniser and the visualisation of information.

A validated named entity recogniser for purpose in law enforcement is proposed in (Brett Crawley & Wagner, 2010), based on rule based entity guessing (grammar based), regular expressions and machine learning they aimed specifically on recognizing locations, persons, telephone and credit card numbers, simple dates, email, URL and IP addresses. The named entity extraction within the architecture is used for recognizing locations and persons; the algorithm is trained on English and German corpora realizing high scores on recall and precision. This technique is applicable to our model but performance is not guaranteed because the model has to be trained on Dutch corpora. Telephone and credit card numbers, simple dates, email addresses, URLs and IP-addresses are extracted by using regular expressions and are applicable to our model after minor localisation modifications.

Another interesting source of information is social media networks because they are closely related to real life activities and communication. (Zainudin, Merabti, & Llewellyn-Jones, 2011) analysed social networks and made an overview of which attributes of an online profile are typically found in social media communities. Zainudin et al. analysed existing models for digital forensics and extended them with the components from their own research on social networks and propose the functional requirements for a prototype founded on their model.

Baldini, Neri and Pettoni (2007) describe an extensive model to perform multi language data mining on unstructured text. Their approach is based on Natural Language Processing and has the ability to perform multi language lexical analysis on large sets of documents. Their model is able to extract functional relationships within a document that are indexed on a conceptual level and can be searched

or browsed by term and can be visualised in a tree view. They have created a search engine based on the same functional relationships. The free text search query a user enters is analysed, the system responds with the conceptual expansion of the query based on the concepts extracted from the document collection. The data analyst selects relevant concepts after which a list of resulting documents is displayed to the user. For our case, parts of their approach can be used to improve precision in the process of searching for personal information on regular web search engines though it is not specifically designed for extraction of personal profile attributes. The paper does not report on any validation of their model though they state military and civilian personnel of Italian Defence are using it.

Colombini and Colella (2012) approach the process of digital profiling by mapping it to the process of traditional profiling and therefore bridge the gap between traditional profiling and digital profiling. They created a model to assess whether or not different mass media devices (e.g. mobile phones, laptops and desktop computers) belong to the same person. They propose a method based on set theory where designated features are extracted from different devices, those features are than compared to a sample profile (set of features) do determine whether or not they are similar. However, their approach is rather specialised to specific devices and operating systems and is not properly validated with real cases and therefore not applicable to our case.

The models found in existing literature propose different techniques for extracting information from a set of existing documents and perform data mining, data extraction and analysis on them, techniques that are relevant to the field of data mining and information retrieval and are applicable to many different applications besides OSINT. However, the discussed models do not perform an ad-hoc search on online sources, which is inevitable when searching highly dynamic sources like social media websites. Another approach would be to crawl and index OSNs ourselves but to accurately keep up with the pace of expansion of those networks would require a large and expensive distributed computing grid. Since we do not have access to a computing grid our model will be based on an ad-hoc search approach on open online sources.

2.4 Finding personal information

As stated before, search engines are an important entry point to (shallow) websites; they provide powerful Boolean operators to specify the search query to gain precision in search results, this can be of great use in search of personal information. Google's advanced Boolean operators are well documented (Long, 2008). Those operators enable the user to execute an advanced search query and search on, for instance, specific sites, specific file types, combinations of those operators can be really powerful and can be used in complex search queries to enable us to benefit from the search engine's wealth of indexed online information.

Online social network search

Because the process of digital profiling is about personal information we consider Online Social Networks (OSNs) the most valuable sources for our goal. The first step in retrieving profile information from an OSN is to find the profile(s) belonging to the person under investigation. OSNs usually address profiles by a username or user-id, to get access to personal information one has to find the user-id or

username corresponding to that person. In most cases it is also possible to find a user by email address but is difficult to implement in an automated system since it usually involves uploading an address book or by performing a search on the social media network's website (Balduzzi, Platzer, Holz, Kirda, Balzarotti, & Kruegel, 2010).

In broad, there are three approaches to find a suspects' username or user-id: using a specified query on a web search engine, using OSN's application programming interface (API) or by executing a search query on the OSN website. Another approach would be to construct possible usernames out of the information available about the subject, for instance the nickname or first and last name. We will describe those approaches in the next paragraphs.

Searching users on Online Social Networks using a web search engine

An OSN can be seen as an undirected graph, where the nodes are entities and the edges the relationships between those entities, every page can therefore be seen as the representation of a node. Entities come in various types, differ per network and often relate to common entities in real life such as: people, events, groups or companies.

By using advanced operators on web search engines we are able to aim the search query on a specific website, in our case an OSN, thereby improving the precision of the results. Performing such queries on a web search engine results are a list of entry nodes to the network that somehow matches our search query. By extracting all (user) ID's, the node's edges, from the result pages we are able to create a list of user profiles, those profiles can be compared with the search query to filter mismatches.

Not all profile pages are indexed on a web search engine. This can be caused by a user who has excluded him-/herself from being indexed by search engines in their privacy settings. If so, a public profile page does not even exist. Since this strategy is searching the indexed web only it will not find those profiles. However, this strategy does find user ID's that match protected profiles if they are found on public pages. In example, if a protected user posts a comment on a public page, that comment is public information.

Extracting information from an OSN

The most popular OSNs in the Netherlands provide an Application Programming Interface (API) to enable developers to create their own applications on the platform or get access to data from the platform.

#	OSN	Dutch members	Dutch unique visitors (per month)
1.	Facebook	7.553.800	8.977.000
2.	YouTube	N/A	8.627.000
3.	LinkedIn	3.500.000	3.907.000
4.	Twitter	1.260.000	3.495.000
5.	Hyves	9.800.000	3.099.000

TABLE 1 OSN USAGE IN THE NETHERLANDS (OOSTERVEER, 2012)

In the early days of OSNs there were a numerous smaller networks instead of a few big OSNs, to connect those networks together the initiative to standardise and organise the interconnection of OSNs was initiated. It was founded by Google and named OpenSocial (Häsel, 2011). The OSNs participated in this project include LinkedIn and Hyves, members of the five biggest OSNs in the Netherlands. The consortium developed a public specification for Application Programming Interfaces (APIs) including the open authorisation standard OAuth (2.0), which is used for authentication throughout the popular OSNs today.

The APIs of all mentioned OSNs in Table 1 are built according to the OpenSocial specification, which means that communicating with those networks is the same in general; authorisation (User→OSN→App) is handled by OAuth 2.0 and getting and posting data occurs in a RESTful manner. This means that, after authentication, an information request through an HTTPS-call returns profile data in a standardised data format (XML or JSON). A HTTP GET request is used for downloading data from the network, a HTTP POST request is used to upload data to the network.

Although communication through the APIs is quiet similar on all OSNs, the data model under the hood of every OSN differs so the HTTP requests differ as well. An API enables us to extract data from an OSN without having to parse HTML pages and the use of text mining techniques that will only slow down the process. To download profile information of a certain individual one has to know the related user-id that can be found in ways described in the previous paragraph.

Remarks

When performing a search on an API, protected profiles are not found, even as an authenticated user. To maximise recall we propose to use a combination of parsing usernames, described in the next paragraph, extended by a web search engine search on the particular OSN.

Parsing usernames

Not all OSNs let the user choose their own username but generates it out of the user's first and last name appended by an optional index number. Thereby, in an analysis of 2.6M Google profile usernames in 2011 we have learned that 69.7% of self-chosen usernames in the Latin character set is a combination of first and/or last name, optionally appended by a digit, which can be age, year of birth or an index number (Perito, Castelluccia, Kaafar, & Manils, 2011). We suggest implementing this strategy in our model to improve recall in searching users on OSNs taking into account that this is only proved on the Latin character set, which is common in the Netherlands.

Future

Facebook recently announced Graph Search (Stocky & Rasmussen, 2013), an advanced search engine on Facebook that enables users to search for persons within the network by profile attributes. This feature enhances the possibility of using Facebook in criminal investigations; at the moment the process of profiling on social media networks can only be initiated if one knows the email address, username or full name. With Facebook's Graph Search it is possible to find all users connected to a certain sports club, age group, gender or even liked pages.

Shallow web search (search engines)

Search engines can be used to directly find personal information as well; interesting sources within the context of the person under investigation include sports club websites, relevant forums, student union websites, corporate websites and other websites where the suspect is mentioned. Using advanced search engine operators, as described before, we are able to search the indexed web specifically and exclude search results from the OSN's websites that we already examined in the previous chapter. Web searching for personal information is done based on a subset of the attributes described in (Zainudin, Merabti, & Llewellyn-Jones, 2011). Valuable starting points in search of personal information are listed in paragraph 2.5.

Extracting information

The first step in extracting information from the search results is retrieving the page and extracting relevant data from it; therefore, we will split the content of the page in text elements and images on a HTML level by using a subset of the HTML element types. By applying the techniques discussed in the previous chapter (Brett Crawley & Wagner, 2010), we are able to extract locations, persons, telephone and credit card numbers, simple dates, email, URL and IP addresses from the page. Furthermore, images can be extracted from websites by parsing all image tags and retrieving the source, those images, together with the elements extracted from the page can be judged by a data analyst and be used to extend the profile of the person under investigation.

2.5 Digital profiling

The term profiling has different meanings in forensics so we will describe this term first with respect to this research. The term digital profiling is based on the term forensic profiling that has two parts;

Forensic: refers to information that is used in court as evidence (Geradts & Sommer, 2006),

Profiling: "The process of 'discovering' correlations between data in databases that can be used to identify and represent a human or nonhuman subject (individual or group), and/or the application of profiles (sets of correlated data) to individuate and represent a subject or to identify a subject as a member of a group or category" (Geradts & Sommer, 2006).

Forensic profiling is merely used to aggregate data from different governmental information systems to be used in court. This model should be extended by including public personal information sources.

Profiling, as such, generally refers to offender profiling, the process of accurately predicting and profiling the characteristics of unknown criminal subjects or offenders and thus does not apply to our research. The model described in this report will support the process of digital profiling; it supports the end user in finding, gathering, aggregating and analysing personal information.

Because the focus of our model is on OSNs we want to determine which types of information scents, attributes, are typically found on an OSN user profile. We will be using the following list of types, determined by examination of several OSNs (Zainudin, Merabti, & Llewellyn-Jones, 2011):

- Name
- Profile picture
- User ID
- Gender
- Birthday
- Religious
- Political views
- Education history
- Work history
- Hometown
- Current location
- Friend requests
- Family and relationships
- List of friends
- Networks
- Music
- TV
- Movies
- Books
- Activities
- Groups
- Website
- Status updates
- Links
- Notes
- Events
- Photos/videos
- Tagged photos and videos
- Messages in inbox
- Posts in News Feed
- Chat

To get a more in-depth view of the availability of those attributes across different OSNs and to verify the list of attribute types we plotted the attribute distribution over the OSNs by using data extracted from (Chen, Kaafar, Friedman, & Boreli, 2012), the diagram is ordered by the average percentage of availability. The dataset that is used for this diagram was obtained through the APIs of the mentioned

OSNs and therefore respects the user's privacy settings; this data was acquired in the period between May and August 2011.

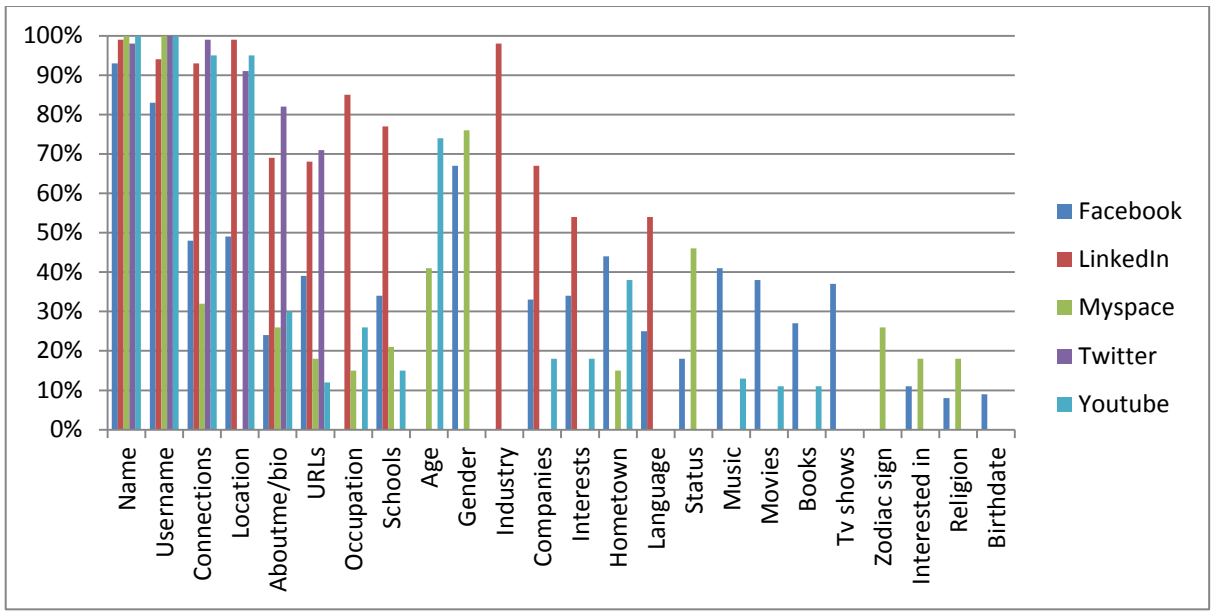


FIGURE 3 ATTRIBUTE DISTRIBUTION ACROSS OSNS (CHEN, KAAFAR, FRIEDMAN, & BORELI, 2012)

From this distribution of attributes we conclude that the first attribute types, name and username, have a high availability on all OSNs, which makes them candidates for comparing profiles on the different networks. The widespread distribution of the other attribute types shows us the importance of crawling multiple OSNs in an investigation to get as much information from a suspect as possible.

The main result of the process of digital profiling is a report on a certain individual giving an overview of all attributes related to the person under investigation with their sources. However, other interesting features can be extracted or calculated from this data, for instance, a social activity timeline that can tell us something about the whereabouts of the individual over time (Huber, Mulazzani, Leithner, Schrittwieser, Wondracek, & Weippl, 2011).

Information found on the Internet can be used in court to support a case, like in (Demmers, 2012), where several printouts of OSN pages are used to support the court statement. To be able to use this information in court it is important to keep a (digital) printout of the original source page, this should be supported by the model.

3 Requirements analysis: The intelligence case

To take into account the different visions and interests of various stakeholders in this project we perform a requirements analysis. This will help us to implement the stakeholders' wishes in the resulting model. We will use (Kulak & Guiney, 2003) as a guideline for the requirement analysis process though we will not elaborate every aspect since their approach. It focuses on building an end application where we want to analyse the requirements for a theoretical model.

3.1 Problem statement

Signal Intelligence is an important form of intelligence since most of our communication occurs through GSM, landlines and analogue connections that are typically easy to eavesdrop and analyse in an automated manner. Due to the emergence of the Internet more communication seems to occur through different open and closed sources on the Internet. Dutch intelligence agencies are particularly interested in analysing the open source side of the Internet and improve their OSINT capabilities in the field of digital profiling. A process that is currently performed manually.

Digital profiling

Digital profiling is an important part of a criminal investigation; it is the process of gathering information about suspects or accomplices. To implement this model in an application we designed the following cyclic process of digital profiling in a criminal investigation:

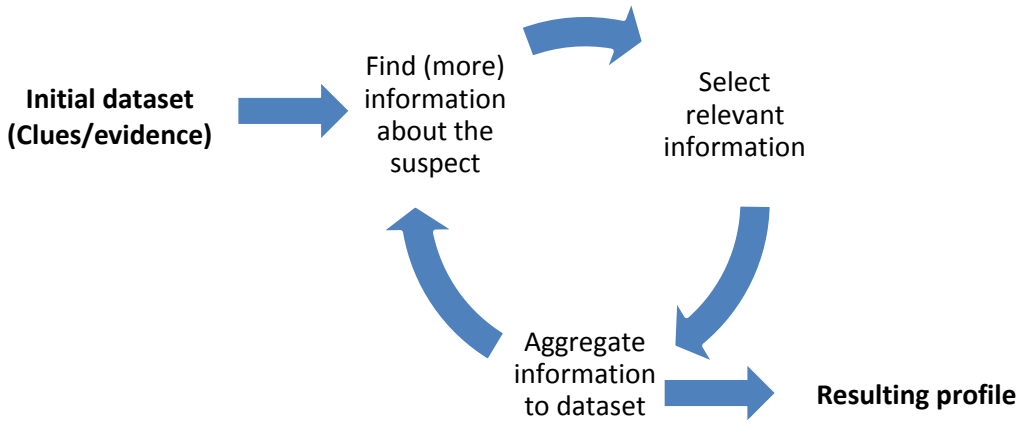


FIGURE 4 ITERATIVE PROFILING PROCESS

The data analyst starts off with clues, evidence or leads that is derived from other sources of intelligence. With this initial data the data analyst starts searching on different sources with different techniques in an iterative manner. Each iteration might provide additional information about the suspect that can extend the user profile, (a combination of) found profile attributes might result in clues that can be used in next iterations.

In classic criminal investigations, investigative profiling plays an important role in mapping the user's social activities and increases traceability of the individual. Due to the rise of the Internet, loads of

attributes that complement a user- or suspect profile can be found online. This information can be used in a criminal investigation to traverse between physical- and online identities as well as within online identities (a suspect does not necessary have one distinct online identity).

Current state

Digital profiling is currently a manual process, performed a data analyst. His task is to search the Internet to find different pieces of information from different sources that could contribute to an user profile. To prevent web-services from detecting the Police on their systems by browser fingerprint or network address the Internet Research Netwerk (iRN) is set up. In the process of digital profiling the data analyst connects to the Internet through iRN, this system prevents the Police from being detected or blocked (Verduurzaming iRN/iColumbo, 2011).

The manual process of digital user profiling is rather time consuming and prone to errors because the data analyst has to examine large volumes of data on a long list of different sources (Appendix I: Data Sources).

Therefore, better exploitation of both human reasoning and information technology is desired. The main goal of the application is to reduce the workload of data analysts and preferably increase recall and precision of the process.

Existing model evaluation

The existing models described in the previous chapter show various methods and algorithms to derive OSINT or perform OSN analysis. However, when looking at the specific process of profiling and a practical implementation, there are a few shortcomings. We do not have the ability nor capacity to systematically crawl and index OSNs, every investigation has to be an ad-hoc operation. This requires modules for integration of every specific source that should be supported by our model. Furthermore, the model should facilitate the possibility to apply proven techniques and algorithms from our literature study and relevant research at TNO in the field of (cross-) network profiling.

Existing applications

There are several commercial solutions available to examine open information sources varying from stand-alone applications to online social network aggregation services. We will briefly evaluate those solutions in this paragraph.

Maltego

Maltego is an open source intelligence and forensics application. It was initially designed to relate and examine different websites. The GUI supports analysis of a (social) networks by using a graph representation. Nodes can be manually added or by applying transformations to existing nodes. Transformations differ, depending on the type of the source node. If the source node is, in example, a website possible transformations are: find email addresses on websites, find outgoing links on the website and the like.

The application excels in user interaction, representing entities from different types as nodes with different colours in a graph keep even more comprehensive networks manageable and clear. However,

the capabilities to investigate online social networks are limited to a few elementary Twitter and MySpace transformations. It is possible to extend the application's transformations with local scripts. For the transformations the application depends on the Paterva Transform Distribution Server and are executed remotely which could cause availability and integrity issues. Data aggregation and export features are not supported. However, the interface and user interaction is clear and applicable to our model.

Pipl

Since the attendance of online social networks profile aggregating websites were developed. Websites like Pipl, Wieowie and 123people systematically crawl and index online social networks, other online communities and personal websites. At start, those websites were mainly focussing on generating traffic and generating money by online conversion. By now, their cross-network analysis and profile mapping capabilities are evolved and surprisingly accurate. Pipl even provides an (paid) API for developers and can therefore be considered a source for our model. The Pipl API supports searching on different input parameters (clues) and returns a list of sources that might relate to the search query. Sources are analysed, per source a match probability is calculated and search suggestions are generated. Though the precision of the results is not sufficient, Pipl can be used as source for our model since human analysis will improve precision.

3.2 Stakeholder analysis

To analyse the interests within this project we will give an overview of the stakeholders.

TNO will be responsible for research and development of the application and the architecture for this project.

The Dutch Government is financial stakeholder since this project is financed by subsidies.

The Dutch National Police (KLPD) (Actor) will be using the application and act as end-user in this project.

3.3 Requirements

Use cases

Use case survey

Name	Description
1. Case overview	Gives an overview of previous cases, analysed by the user that is currently logged in. Each case will have a hyperlink to its designated detailed case file page.
2. Report	Presents a detailed case file for a specific case.
3. Search	Presents a form to the user to start a new investigative user search.

4. Results	Presents an overview of found profiles to the user.
5. Export digital case file	Export resulted profile as case file including page 'printouts', the search log file and images.

Features/wishes

Zainudin, merabti and Llewellyn-Jones (2011) described functional requirements for supporting the process of forensic investigation on online social networks. This set is categorised in three levels of importance and extended with additional functionality.

Must haves

1. Ability to extract data from OSNs: the model should be able to implement the use of different sources for searching and data extraction.
2. The ability to search and filter data: the model should be able to automatically search for data on pre-specified sources and should be able to determine and filter irrelevant results. It should order search results by relevance.
3. The ability to cope with multiple users: the resulting model should be able to cope with different users to work at the same time on the same system.
4. Implemented privacy measures: avoid keeping data attributes in the database that are not relevant for any investigation.

Should haves

1. Ability to report comprehensively: The resulting model should be able to create a report based on the search process and should provide log files from user actions, an aggregated profile and all found images related to the user.
2. Process management: the user should be able to alter the process of deriving intelligence from open source information.
3. Ability to rapid prototype: the resulting model should form the foundation for a working prototype.
4. Ability to export case files: the model should support exportation of the case report.

Could haves

1. Ability to perform batch analysis: some OSN analysing techniques are time consuming due to OSN API rate limiting and should therefore be performed in batch jobs.

Individual use cases

Use case name	**1. Case overview**
Summary	Gives an overview of previous cases, analysed by the user that is currently logged in. For each result the following attributes will be showed:

	• Profile source • (Profile picture) • User ID on source • Found profile attributes • A hyperlink to its designated detailed case file page
Basic course of events	1. The user starts the profiler application 2. The system presents cases related to the current user
Triggers	The data analyst wants to see an overview of previous cases.
Preconditions	The user is logged in.
Post conditions	The user is informed about previous cases.

Use case name	**2. Report**
Summary	Presents the user an unambiguous detailed view of a case. Each data attribute is presented with the source(s) it came from. Any inconsistencies between sources will be shown.
Basic course of events	1. The user starts the profiler application 2. The system presents cases related to the current user 3. The user selects the case of interest 4. The system shows the user the detailed report of the selected case
Triggers	The data analyst wants to see the details of a specific case.
Preconditions	The user is logged in.
Post conditions	The user is informed about a specific case in detail.

Use case name	**3. Search**
Summary	Presents a form to the user to enter search query details. Query form parameters are the most common attributes: • Name • Username

	• Email address As well as an input to specify the desired search target. At least one of the fields has to contain a query term and at least one target has to be selected in order for the form to be submitted.
Basic course of events	1. The user starts the profiler application 2. The user triggers the search button 3. The system presents the user the search query form 4. The user enters the query terms and search target and submits the form 5. The system will start to search
Triggers	The data analyst wants to start a new investigation case.
Preconditions	The user is logged in.
Post conditions	The user is informed about existing profiles on the selected targets that might belong to the suspect of the investigation.

Use case name	**4. Results**
Summary	Presents an overview of found profiles to the user. For each result the following attributes will be showed: • Profile source • (Profile picture) • User ID on source • A selection of found profile attributes • A checkbox to mark the result as relevant
Basic course of events	1. The user starts the profiler application 2. The user triggers the search button 3. The system presents the user the search query form 4. The user enters the query terms and search target and submits the form 5. The system will perform the search operation and present the results as described to the user
Triggers	The data analyst wants to see an overview of matching profiles on specific sources.
Preconditions	The user is logged in.
Post conditions	The user is informed about profiles on different open information sources.

Use case name	5. Export digital case file
Summary	After a case is investigated, the system should be able to export the digital case file (report) for prosecution purposes. This digital case file should at least contain the following elements: • Search log file • Collected images and data elements and their sources
Basic course of events	1. The user starts the profiler application 2. The system presents cases related to the current user 3. The user selects the case to export and triggers the export button 4. The system will generate a compressed digital case file, which is available for the user to download.
Triggers	The data analyst wants to download a digital case file.
Preconditions	The user is logged in.
Post conditions	The user is able to download the compressed digital case file.

As an example and a requirement for functional testing we defined a typical scenario in Appendix III: Requirements scenario.

4 Initial model – The Prototype application

The possibilities of OSINT are an on-going topic of research within TNO, it is used often within different projects in- and outside the department of media mining. As a foundation for further research and as a proof of concept of our model we decided to partly implement our model in a prototype application. This chapter will give an overview of the prototype and practical implementation issues.

4.1 Used technologies

We choose to implement the model in a web application because of cross-platform compatibility and multi-user support. The basis for the implementation is the Django web framework; this platform has usable features for rapid prototype development and scalability.

The data model is defined in our Django project and deployed on an SQLite database. Because Django is written in python we extended it with libraries for authorisation on OSNs (OAuth 2.0), HTML-parsing (BeatifulSoup), URL handling (urllib2) and many more. Because we built a web application with smooth user interaction we wanted to use AJAX, therefore we used HTML, CSS and jQuery.

4.2 Implemented features

Because we are developing a prototype we decided to limit the scope by only implementing the three biggest OSNs at this time: Facebook, Twitter and LinkedIn. The following paragraph will give an overview of the implemented features. Because we choose to use Django we adapted and used the following features from the framework: database abstraction, (multi) user management, the automatic administration interface and access control. The search process consists of the following steps and user actions that are supported by the prototype.

In the first view, the OSN Search, the data analyst enters a search query (Name, Email or Username) and one or more target network(s).

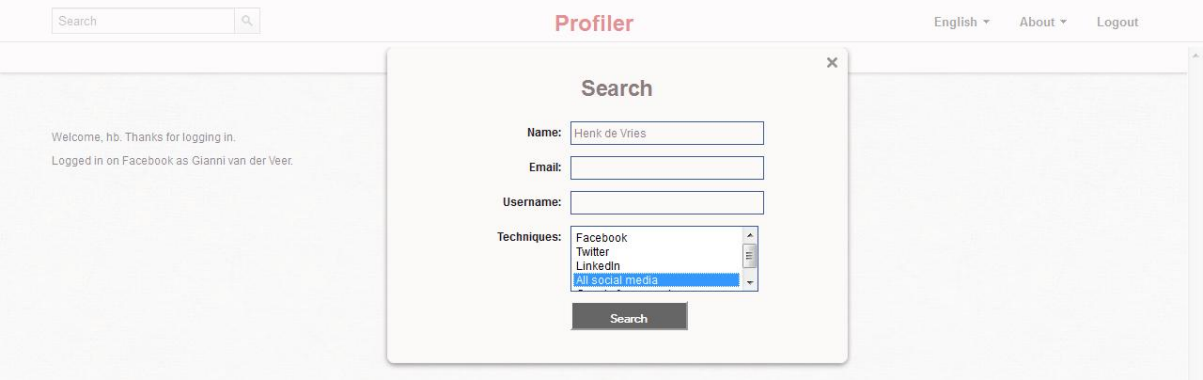


FIGURE 5 PROTOTYPE: OSN SEARCH

After submitting the form, the application will perform an automated search for profiles on the selected networks and will display the found profiles in an overview that looks as follows:

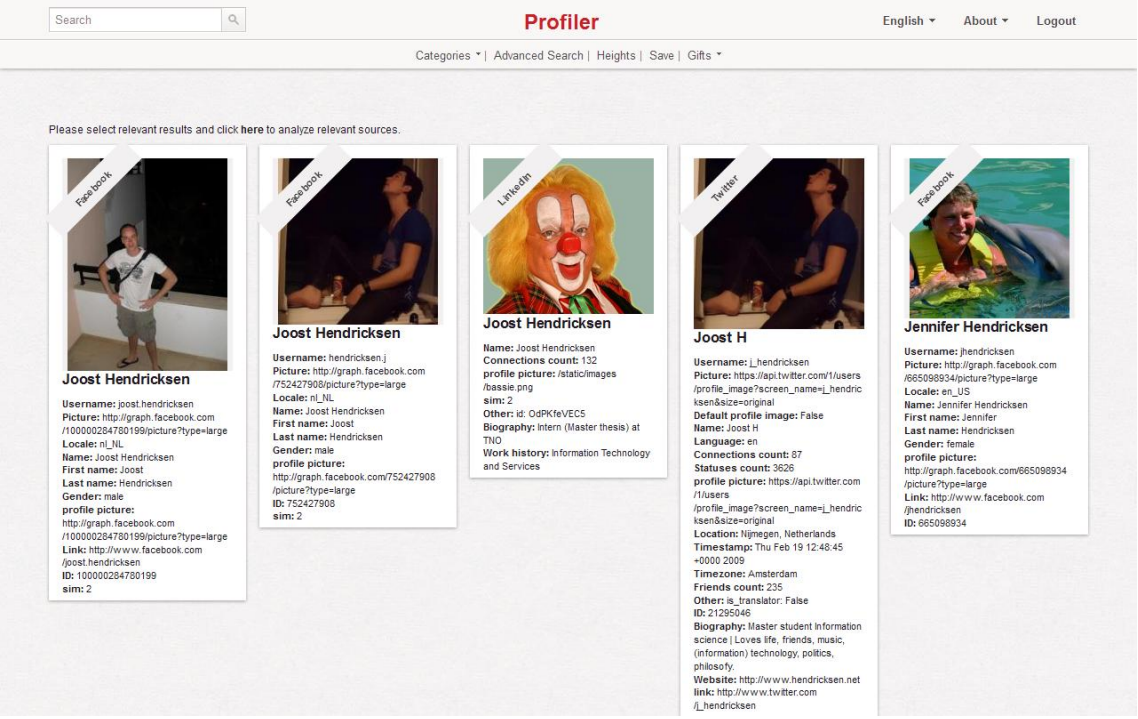


FIGURE 6 PROTOTYPE: PROFILES OVERVIEW

In the profiles overview the results are ordered by relevance, based on the search query. In this view the data analyst makes a pre-selection of relevant profiles.

After submitting the pre-selection, the system will retrieve all data from the OSNs of the selected profiles. In the full profiles overview, showed in figure 6 (cropped picture), all found profile attributes are displayed. The data analyst can select additional attributes that are saved to the suspects' profile after submitting.

The last step, showed in picture 7, gives an overview of the aggregated suspect profile, composed of attributes that were marked relevant by the data analyst during the process.

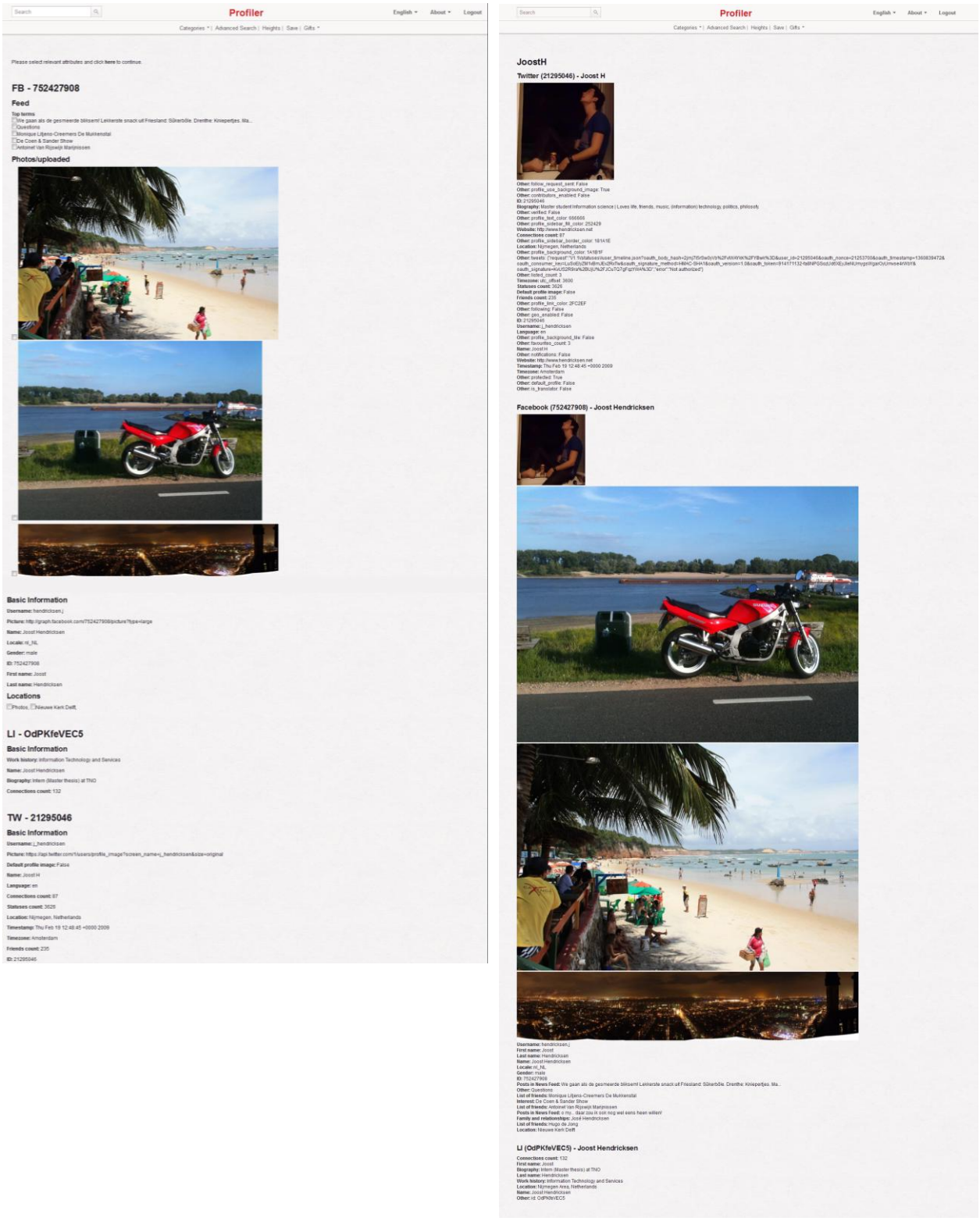


FIGURE 7 PROTOTYPE: FULL PROFILES CROPPED (LEFT) AND AGGREGATED PROFILE (RIGHT)

Architecture

Our web framework is using a Model View Controller (MVC) architecture pattern that separates different aspects of an application implementation. The model consists of the data model, operations regarding the model and validation rules. The view describes an output representation of data, for instance HTML or JSON. The controller translates user input to the model or view. To be able to implement our model to a MVC architecture we had to apply minor changes that resulted in the following architecture:

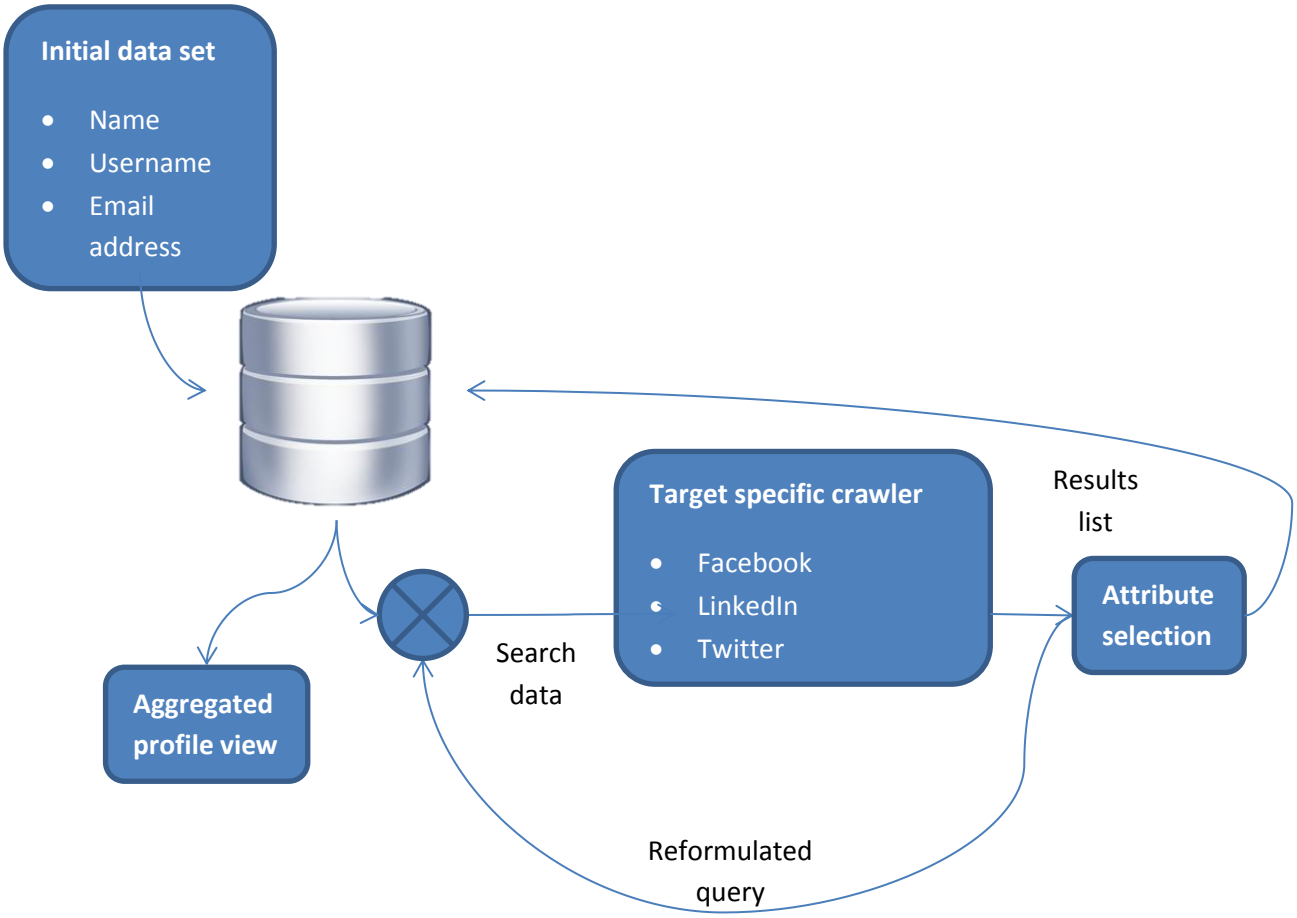


FIGURE 8 PROTOTYPE ARCHITECTURE

In this architecture the search, extraction, loading and transformation for each OSN is realised in the target specific crawler.

Target specific crawler

The target specific crawler will perform a set of OSN specific search strategies to find relevant data on the designated target. Its processing pipeline contains the following modules:

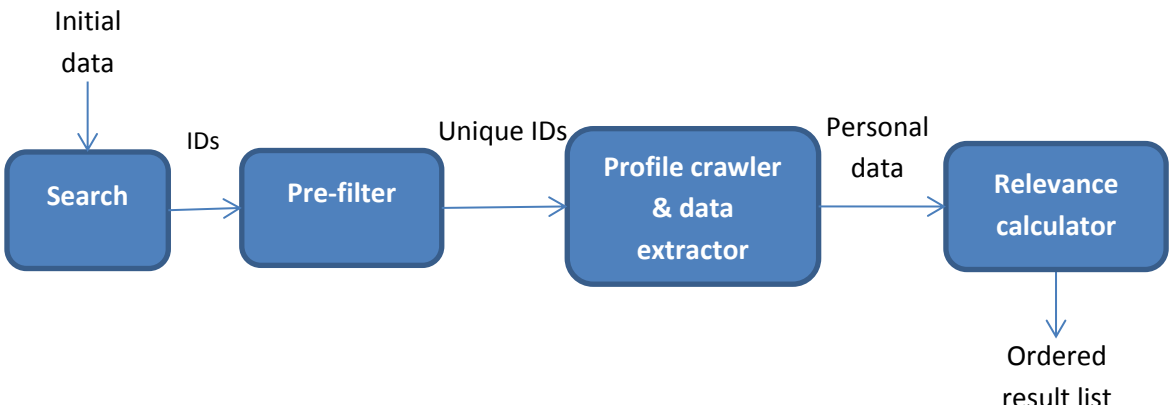


FIGURE 9 TARGET SPECIFIC CRAWLER PIPELINE

When the target specific crawlers are initiated they will authenticate with the OSN's API, since this authentication (OAuth 2.0) will eventually time out the system will ask each user to log in and grant the profiler application permission to access the OSN account.

By supplying initial input parameters to the target specific crawler it will request all profiles found from the OSN API and translate attribute types to our general attribute types. The relevance calculator will add a relevance ratio (0 … 1) to the result and order all results based on the input parameters.

Search module

The search module uses the initial data to perform a search query on an OSN, depending on the target OSN it applies different search strategies. Every crawler will perform a search request on the OSN's API, depending on the API specification data might need to be transformed before submitting it.

As described in chapter 2, strategies include user name parsing or specific web search engine searches to improve recall; those strategies are implemented in the search module. Parsed usernames are appended to the result list.

For web search engines the search results in the HTML source are placed in class identified DIV elements. After performing a search query on a web search engine, the result page is parsed to extract the URL's of the search result. To extract user ID's from each search result the page is parsed and all hyperlinks are extracted, further examination of the URL classify whether or not a URL is linking to a user profile. If so, the user ID is extracted and appended to the result list.

Pre-filter

The list of user ID's from the search process contains duplicate usernames and user IDs. The pre-filter will create a distinct list of unique IDs that is passed on to the profile crawler.

Profile crawler & data extractor

Depending on the target the parser will either use the API or parse the HTML content of an URL to extract user profile data from a page or a profile. The HTML parsing scheme is hardcoded in the

application. Each found attribute type on the target would be translated to our general types by an array of dictionaries.

Relevance calculator

Because different strategies are used to find user profiles the results might not all be relevant. To calculate the relevance of a profile we used the following approach: the presence of the terms from the search query in the resulting profiles are calculated and normalised by dividing it by the total number of terms in the profile.

4.3 Implementation issues

Performing search queries on a web search engine in an automated manner will cause availability problems if used often in a short period of time. This problem can be solved by waiting a random time between queries, but will introduce a delay in delivering results to the user.

If a search process is started on multiple OSNs at once it can take long time before results are returned. To speed up this process threading functionality of python can be used to perform searches on multiple OSNs in parallel. A detailed analysis of implementing an OSN can be found in Appendix II: Facebook, a case study.

4.4 Privacy considerations

To make the online profiling tool most efficient personal data is being saved and aggregated during usage. In order to avoid privacy issues special functionality is implemented to protect users privacy.

During search on a specific suspect all found profile attributes are saved in the database in order to calculate relevance before displaying it to the user. The user selects profiles or profile attributes that are relevant to the current case, those profiles and profile attributes are related to a case in the data-model.

Our system is using different API's and various sources to extract data from. As mentioned before, each source is limited by number of requests in given time. In order to avoid outages of our sources we cannot permit to request a single profile over and over again. Therefore we keep profiles in our database for two days before removing them. Each day a batch job removes profiles that are not linked to any case and are in the database for at least two days.

5 Final model - Profiler

Our OSINT model has to be integrated in the field of forensics described in chapter 2.3 because it cannot operate standalone. Input parameters (clues) from the other sources of intelligence, HUMINT and SIGINT are required (e.g. username, email address or full name). The data analyst is responsible for providing those parameters to the model and will act as the hub between the different sources of intelligence, as visualised in the following diagram.

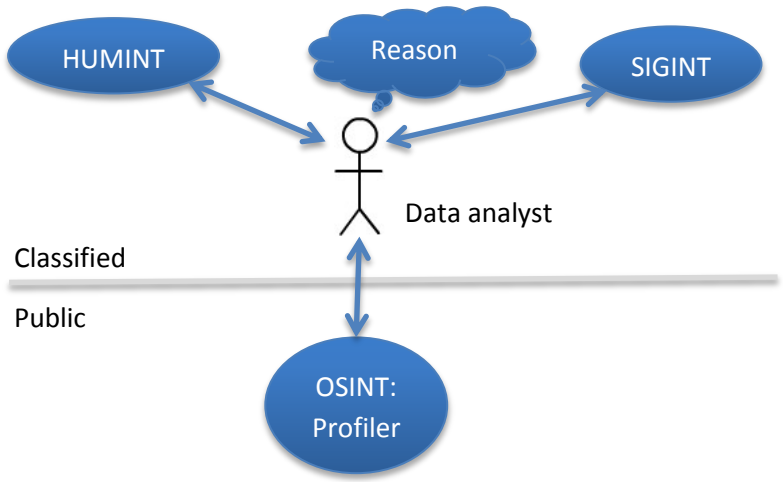


FIGURE 10 THE DATA ANALYST ACTS AS A HUB IN DIFFERENT INTELLIGENCE SOURCES

The final model will extend the initial model described in chapter 4. The initial model supports all must-have functionality of the requirements defined in chapter 3. The user interface, data extraction modules (Figure 9) and user interaction patterns will remain in the final model. To enable the model to facilitate the should-have functionality, the data model will be altered and extended. This will improve the strength of evidence and the ability to change the process of deriving evidence from various sources.

5.1 System overview

Because we use a variety of different dynamic sources with different attributes we choose to apply the separation of concerns design principle. The distinct sections of our model are the open source information part and the intelligence part, which nicely resolves to OSINT. This enables us to store retrieved profile information as found, preserving authenticity of the source data and making derivative intelligence traceable. To realise this we created an abstraction layer. The system architecture to support this process is described in the following picture:

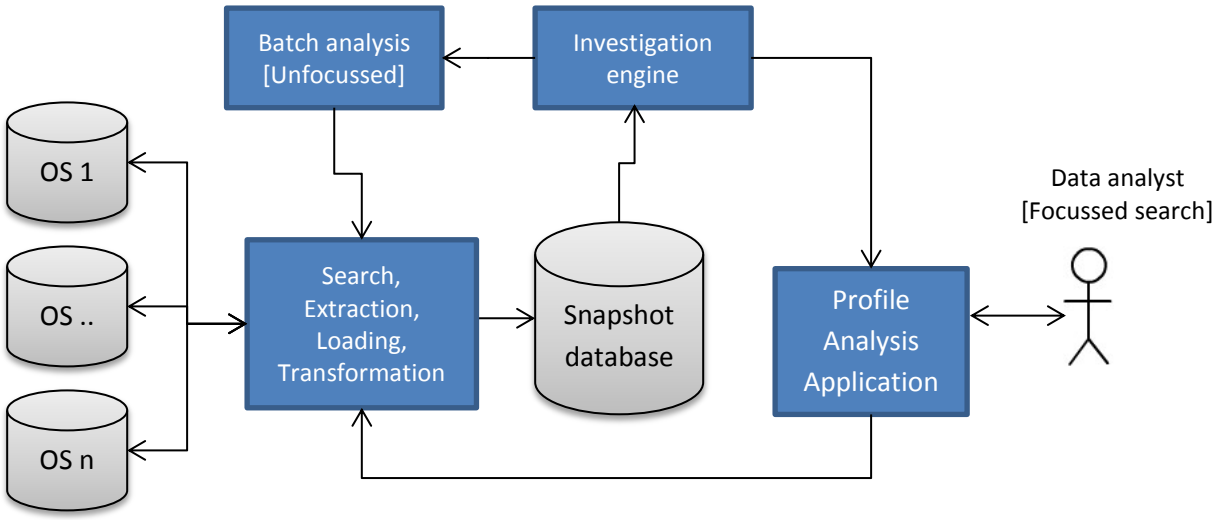


FIGURE 11 ARCHITECTURE OVERVIEW

The architecture is based on a data warehouse architecture, an architecture that originates from enterprise information technology and is used to collect data from different business processes and save them to a central database. This database is used for business intelligence purposes and to facilitate decision support systems.

This architecture fits to the process of digital profiling where data from several sources in various forms and structures has to be reduced to a more generalised form: attribute value pairs. Separating the data extraction- and the analysis process enables us to model highly dynamic sources while the analysis algorithms can be applied on generalised data through the abstraction layer. This enables us to implement features from the other models discussed in chapter 2 and still support various sources of personal information and an ad-hoc search approach. To enable the end user to influent the search process the application controls the search, extraction, loading and transformation module.

5.2 Model

To specifically support the process of profiling for intelligence purposes we developed the model described in this chapter. The proposed model is tailored to OSINT profiling but can be applied to digital profiling in general. The data and process model are strictly divided in an Open Source Information part and the Intelligence part. The first is designed to secure and save information as found on an open information source. Information in this part of the model can only be saved and not be altered to secure the sources of evidence. The intelligence part of the model is used to save the intelligence derived from those open sources. To provide traceability the evidence relates to its source data and includes a log message that describes how the evidence is derived.

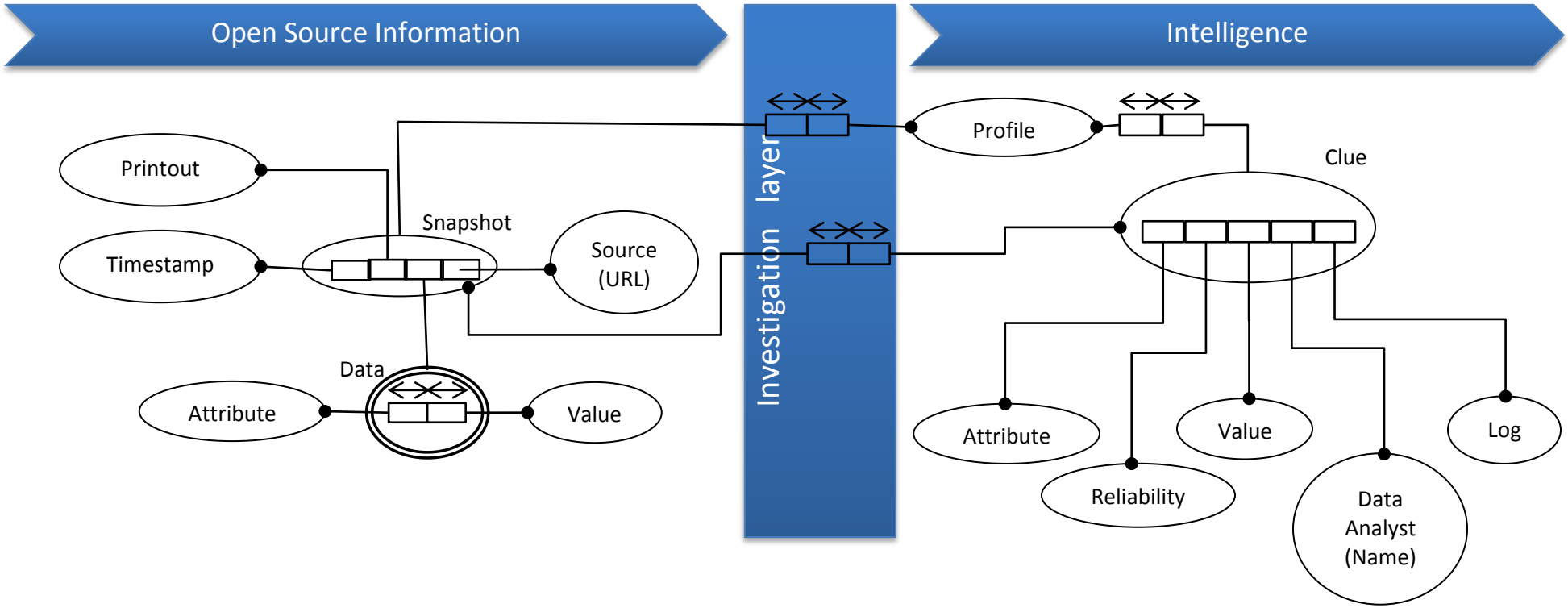


FIGURE 12 THE OSINT MODEL

5.3 Open Source Profile Information

In this model the source data is saved as a set of singular Attribute-Value pairs as found on the source. The set of Attribute-Value pairs from a specific source are saved to snapshot that contains meta-data to support traceability and reproducibility. Because the snapshot contains a timestamp it can be retrieved more often over time to analyse profile changes. The source URL is saved in the snapshot for identification and to be able to retrieve (new) data from the same source again.

Snapshot

A snapshot represents a certain data source at a specific time. It consists of the set of Attribute-Value pairs found on a certain Source at a certain time (timestamp). It is our model's representation of a suspect profile on a certain source at a certain time.

Source (URL)

A source in this model is considered an open source information (OSINF) source described by an URL. The URL is used to identify an information source and contains the information needed to recapture the data from the source.

An URL can be decoded to a hostname, path and a query represented by query parameters. For clarification, the URL *https://graph.facebook.com/2983478392?fields=id,name* can be decoded to hostname *graph.facebook.com*, path *2983478392* and query *fields=id,name.* The hostname determines the source type and therefore specifies which specific crawler applies, in this case the Facebook crawler that uses the Facebook API to retrieve the data. If no specific crawler applies the general web crawler can be applied to download the HTML content. After text extraction by a HTML parser a named entity extractor, as described in (Brett Crawley & Wagner, 2010), can be applied to extract snapshot attributes.

Printout

In court, printouts of webpages are added to the criminal record to secure the evidence. To ensure results of our model can be used as evidence and to improve traceability a digital printout (PDF) of the source page is added to the snapshot. An open source library (e.g. HTMLDOC[1]) converts the source page to a PDF file. The PDF file is named after the ID of the snapshot and saved to the hard disk.

Timestamp

By adding a timestamp to a snapshot we are able to make multiple snapshots of the same profile over time and compare or aggregate those profiles. Most sources of personal information are highly dynamic; therefore a timestamp is valuable meta-data. It also enables a data analyst to capture multiple snapshots of the same source over time and analyse changes in the user profile. The model is able to automatically monitor suspects over time. However, systematically monitoring users is considered controversial, not yet allowed by law and should thereby not be used.

[1] HTMLDOC - http://www.htmldoc.org/

Data

This element specifies the personal data found on a specific source. It is saved as a set of Attribute-Value pairs. To secure authenticity of the data, the attributes as well as the data are saved as found on the information source. The attribute, data format and source-type can be used as conditions in the investigation layer.

5.4 Intelligence data

The intelligence data that is derived by the investigation layer is saved in the intelligence data part of the model. The following data classes represent the intelligence profile for a certain suspect.

Profile

In our data-model a profile is represented by set of source snapshots (information) and a set of derived clues (intelligence). The related snapshots contain the source information that was found on the Internet and led to clues. A clue contains personal data and information about the derivation process that led to this data. This information will increase usability of the evidence in prosecution of the suspect.

Clue

A clue is considered a 5-tuple containing a log message, reliability factor, the name of the responsible data analyst and the Attribute-Value pair. A clue is a single result of the profiling process but can be derived from multiple sources. If a clue is supported by multiple sources the accumulated reliability factor improves.

Log message

The log message describes in text how the derived Attribute-Value pair is derived. It contains information about which rule is applied on which source data. This attribute improves traceability of the clue and provides transparency about the forensic rule-engine that is explained in the next paragraph.

Reliability

The reliability factor in a clue-tuple describes reliability of the derived Attribute-Value pair on a scale from 0 to 1. A clue that is supported by multiple sources/rules is considered more reliable. Through this we designed the following algorithm to determine a gain factor and the reliability factor that can be applied to single- and multi-source clues. The average reliability of the applied rules is amplified by gain G which is determined by the reliability factor R_i of the applied rule *i* and the total number of applied rules N as follows:

$$G = \frac{1 + (1 - \frac{1}{N})}{2}$$

The accumulated reliability factor is formulated as follows:

$$Reliability = Min(1, G * \sum_{i=0}^{i=N} \frac{Ri}{N})$$

With this approach a clue supported by a single rule is amplified by factor 1.0 and remains unchanged, if a clue is supported by 10 sources the average reliability factor will be amplified by factor 1.45.

Data analyst (Name)

The data analyst element contains the name of the responsible data analyst in the specific investigation; this benefits the traceability of the evidence.

Attribute

The attributes used in the intelligence part of the model are generalised to a static list of attributes described in the following list and derived from (Zainudin, Merabti, & Llewellyn-Jones, 2011).

- Name
- Profile picture
- User ID
- Gender
- Birthday
- Religion
- Political view
- Education
- Work
- Hometown
- Location
- Friend request
- Relationship
- Friend
- Music
- TV
- Movie
- Book
- Activity
- Group
- Website
- Status update
- Link
- Note
- Event
- (Tagged) Photo
- (Tagged) Video
- Message
- Status update

The initial rule-base should at least ensure source attributes are translated to this generic list of attributes.

Value

The value element contains the string representation of the actual clue. If the value is an URL that points to an image, the image is downloaded and saved to the hard drive. The filename will be the ID of the value element. The value string in the data model will be set to the full path of the image. To determine whether or not an URL points to an image we validate the file extension with a list of image file extensions.

5.5 Investigation layer and engine

The investigation layer is a framework to implement the intelligence derivation from open source profile information. This is realised by formulating and applying rules to transform personal data in the open information part of the model to forensic clues in the intelligence part of the model.

Rules can be developed to perform on a low abstraction level where different source-specific attributes are translated to generalised attributes. On a higher abstraction level more advanced clues can be extracted by rules as well, for instance a social activity timeline from different information sources.

To implement this functionality we use a business rules approach that has been proven in enterprise software systems to derive business intelligence. In a data warehouse architecture, as described before, business rules can be applied to extract business intelligence from various business processes. Parallel to our model forensic rules can be applied to extract Open Source Intelligence from various open information sources.

Many commercial and open source rule engines are available to implement this technology; we will prove our concept by formalizing forensic rules in Jess[2], a rule engine and scripting environment. Jess has the capacity to "reason" using knowledge supplied in the form of rules (Friendman-Hill, 2003). This "reasoning" process is realised by evaluating conditions and priorities of individual rules. Our concept is proved by the following Jess implementation example.

In Jess, templates define data structures for in- and output objects. To integrate the rule-engine in our data model we define the in- and output objects in two templates. The input object, personal data, is loaded from the snapshots in our model, consisting of attribute-value pairs and extended with the snapshot source type (e.g. Twitter). Our output object is saved in the data model as a clue, a 5-tuple containing an attribute, a value, a reliability factor, a log message and the name of the data analyst. This structure corresponds with the data model in which the clues are saved. The resulting implementation is presented in Table 2.

```
;;Template declaration
(deftemplate personal_data
    (slot attribute)
    (slot value)
    (slot source_type)
)

(deftemplate clue
    (slot attribute)
    (slot value)
    (slot reliability)
    (slot log)
    (slot data-analyst)
)
```

TABLE 2 DECLARATION OF DATA TEMPLATES

To be able to test the implemented rules, the (input) templates have to be populated. An instance of a template is considered a fact. The test data is chosen in a way all rules apply at least once. To be able to construct a clue data structure the name of the data analyst is globally defined. The current year is also statically defined as a global variable. In a final application those attributes can be requested on the fly.

```
;;Input facts declaration
(deffacts snapshot_a
    (personal_data (attribute screen_name) (value willemp54))
    (personal_data (attribute age) (value 59))
    (personal_data (attribute first_name) (value Willem))
    (personal_data (attribute last_name) (value Peters))
```

[2] Jess, the Rule Engine for the Java Platform - http://herzberg.ca.sandia.gov/

```
)

;;Global variable definition
(defglobal ?*data_analyst* = "Henk de Vries")
(defglobal ?*year_now* = 2013)
```

TABLE 3 POPULATION OF INITIAL FACTS AND GLOBAL VARIABLES

A forensic rule is defined by a rule name, a set of conditions and a set of actions. The rule engine will evaluate input data by applying the actions of those rules where the conditions apply. Typical conditions are the presence of certain attributes or data in a certain format. Typical actions are: asserting a resulting fact (output data) or reformatting/altering data. Because data transforming can be inaccurate a reliability factor is added to every rule. To ensure the most reliable clues are derived a salience factor (priority) can be added to rules that have similar conditions. For the model to work properly a complete rule-base should at least contain rules to translate all source specific attributes to general attributes. In Table 4, several rule examples are given.

```
;; Rules declaration

;; Rule to convert first- and last name attributes to full name attribute
(defrule firstlast_name
    (personal_data (attribute first_name) (value ?fn))
    (personal_data (attribute last_name) (value ?ln))
    =>
    (assert (clue (attribute name) (value (str-cat ?fn " " ?ln))
(reliability .75) (log "Applied rule: firstlast_name to concatenate first
and last name to full name") (data-analyst ?*data_analyst*) ))
)

;; Rule to convert age to birthdate with low reliability
(defrule age_birthdate
    (personal_data (attribute age) (value ?x))
    =>
    (assert (clue (attribute birthdate) (value (format nil 00-00-%d (-
?*year_now* ?x))) (reliability .5) (log "Applied rule: age_birthdate to
convert age to birthdate") (data-analyst ?*data_analyst*) ))
)

;; Rule to translate screen_name attribute to username attribute
(defrule screen_name
    (personal_data (attribute screen_name) (value ?x))
    =>
    (assert (clue (attribute username) (value ?x) (reliability 1.0) (log
"Applied rule: screen_name to translate screen_name to username") (data-
analyst ?*data_analyst*)))
)
```

TABLE 4 RULE DEFINITIONS

To start the rule evaluation the system has to be 'reset' to clear existing facts. The 'run' command will start the evaluation and the 'facts' command will show all initial and resulting facts.

```
;;Rule evaluation
(reset)
(run)
```

```
(facts)
```

TABLE 5 INITIALISATION OF RULE EVALUATION

The output of the Jess rule evaluation is showed in Table 6 Jess output: Initial facts (personal_data) and resulting facts (clues). By integrating Jess in the resulting profiling application the resulting facts can be saved in the data model.

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008

f-0    (MAIN::initial-fact)
f-1    (MAIN::personal_data (attribute screen_name) (value willemp54))
f-2    (MAIN::personal_data (attribute age) (value 59))
f-3    (MAIN::personal_data (attribute first_name) (value Willem))
f-4    (MAIN::personal_data (attribute last_name) (value Peters))
f-5     (MAIN::clue (attribute name) (value "Willem Peters") (reliability
0.75) (log "Applied rule: firstlast_name to concatenate first and last
name to full name") (data-analyst "Henk de Vries"))
f-6    (MAIN::clue (attribute birthdate) (value "00-00-1954") (reliability
0.5) (log "Applied rule: age_birthdate to convert age to birthdate")
(data-analyst "Henk de Vries"))
f-7    (MAIN::clue (attribute username) (value willemp54) (reliability 1.0)
(log "Applied rule: screen_name to translate screen_name to username")
(data-analyst "Henk de Vries"))
For a total of 8 facts in module MAIN.
```

TABLE 6 JESS OUTPUT: INITIAL FACTS (PERSONAL_DATA) AND RESULTING FACTS (CLUES)

Rule engines like Jess supply features to prioritise rules and features to specifically define whether or not rules apply to the fact set. By separating the source data from the intelligence data we are able to re-evaluate previously gathered data if intelligence-rules change due gained insight and experiences. This might eventually lead to new clues in old cases as it often happens in the field of criminal investigation due to new technology.

5.6 Organisational implementation

To implement the final model in an intelligence organisation we will propose roles for the actors on the system. The different roles ensure separation of responsibilities and ensure that the different fields of knowledge contribute to the right aspects of digital profiling.

Domain experts

Because the quality of forensic rules will determine quality of the resulting clues a proper rule-base is required. Domain experts in the field of intelligence should be trained to define and implement intelligence rules in the model. Intelligence experts have the knowledge and experience to reason about rules, their importance and reliability. They should thereby be responsible for those aspects of the model. We presume domain-experts do not necessary have the proper knowledge about business rules, they should be trained to accurately translate their knowledge to rules in the model. This will result in a more qualitative output of the model.

Data analysts

The entire system is designed to assist data analysts in their job. They should be trained to use the application and to understand the process of digital profiling and how this is implemented in the application. The data analysts will be responsible for the output of the system: the digital user profile. Thereby, their name is related to the profile in the data-model.

Computer programmers

The computer programmers are responsible for implementing the model in the application and generally maintaining the application. The input for the model is provided by many different sources and requires maintenance. Online social networks are constantly evolving; the specification of APIs and their data structures change over time the system should be well maintained. It is the programmers responsibility to keep the system working properly.

6 VALIDATION

This chapter describes the validation of the model by explaining and substantiating every step in the research process.

6.1 Literature study

The first step in the process of developing a model to assist in digital profiling of a certain individual using online open source information is to study the state of the art. We did so by studying scientific publications in the field of open source intelligence, digital profiling and social media analysis.

To identify which sources can be used and found on the Internet we researched open information. We described the deep and shallow web and concluded that both the shallow web and parts of the deep web contain personal information. Online Social Networks (OSNs) are a valuable source for personal information and are partly considered deep web. To be able to use these sources OSNs provide Application Programming Interfaces, these interfaces are provided to support developers to build applications that use information from social networks. APIs are used in our model to search for user profiles and extract profile information from an OSN.

To ensure the model is able to contribute to the process of a criminal investigation and prosecution we analysed course judgements and law publications. We found out (partially) open information sources are already being used in prosecutions and are included as a printout in the file of the condemned individual. Justice is interested in extending the possibilities to use online information in prosecutions but realises such could cause privacy implications. Current laws do not make a distinction between the digital and physical world in terms of observing a suspect. This is still a topic of discussion.

To embed our research in the state of the art we studied scientific literature in the field of Open Source Intelligence. Research has been done in the field of OSINT and digital profiling but specifically in offline profiling where electronic devices belonging to a suspect are analysed. Models to compare profiles from different devices are also applicable in online user profiling. A lot of analysis of social networks has been done that contributes to our research to better understand what to typically find on certain social networks.

6.2 Requirements analysis

To be able to develop an (prototype) application to support data analysts in their work we performed a requirements analysis. We mainly used input from domain experts at TNO and used investigation requirements described in scientific literature.

6.3 Feasibility study

The specified requirements led to an initial model. As a feasibility study we built a prototype application based on this model. By evaluating this application we determined shortcomings of our initial model. We implemented the three largest social networks: Facebook, Twitter and LinkedIn in the prototype application. As described in chapter 4, a source specific crawler had to be built for each source. This approach proved to be fairly devious because the translation of attributes and different

data types on the various sources had to be statically implemented. In this initial model attributes from various sources are generalised before saving in the database. Therefore the attributes are no longer directly traceable to their original source.

By implementing our initial model we validated the process of extracting profile attributes from various sources in an automated manner. We indicated a lack of flexibility in the model, a more systematic approach to the process of analysing and saving profile data is desirable.

6.4 Improved model

To bring a solution to the shortcomings described in the previous paragraph we introduced the abstraction layer. This led to a clear separation between the data extracting, intelligence derivation and the resulting profile information. This is realised by implementing a rule-engine as a layer between open information and intelligence derived from this information.

By introducing a rule engine in our model we improved traceability of clues. It enhances possibilities to alter and append derivation rules. It also secured the profiling process by clearly logging the process from snapshot attributes to a set of clues.

To validate this concept we constructed a data set to test the rule-engine, the results are presented in Table 6 Jess output: Initial facts (personal_data) and resulting facts (clues). By expanding the rule-base more sources can be implemented in the model.

7 CONCLUSIONS

In this thesis we described the concepts of digital user profiling using open information sources to conduct a research on the possibilities of using open source information in the process of user profiling. The result is a model for a new investigation instrument that contributes to criminal investigations performed by intelligence agencies and organisations.

The Dutch police force is already creating online user profiles to use in criminal investigations. A data analyst performs this process of digital profiling manually, which is time consuming and prone to errors. Therefore, a more automated and less time consuming way of profiling is desired.

In order to satisfy the needs of a data analyst in the profiling process we performed a requirements analysis. We defined the workflow in the process of digital profiling and developed a model to support this process. After implementation of this model we concluded our initial model was not sufficient. There was a lack of agility in implementing and analysing different sources and the evidence, the set of clues, was not fully traceable to the information source.

To cope with those shortcomings we analysed techniques from another form of intelligence, business intelligence. To derive business intelligence from various information systems within an enterprise a data warehouse architecture is applied. This architecture aggregates data from various enterprise software systems and saves them to a general database. Business rules are applied on this centralised data to monitor various business processes and to derive business intelligence. We applied this architecture to our problem by translating it to the field of digital user profiling using open source information. Various open online information sources are saved to a centralised database. Applying "forensic rules" on this centralised data enables the model to derive intelligence in an automated, yet agile manner. Using this rule-based approach creates a clear distinction between the authentic source data and the intelligence derived from it. It also enables domain experts to change, enhance or add derivation rules that could lead to new forensic clues.

The proposed model is able to support online profiling using various sources. By designing an adaptive data model it should support the process of digital online profiling in the future. Still, society is adapting more quickly than justice and law can. Automated user profiling is still in its early days so more research and practical experience is needed.

7.1 Future works

More research should be done on implementing our proposed model in an application. Most rule engines are implemented in Java so a connection between the rule engine and web framework should be realised. Furthermore, the frontend of the application should be extended and user interaction should be improved.

To realise a usable application a rule-base foundation should be set up to cope with default attributes in various information sources. More research is desired on the end result of the user profiling process, an analysis of usable information that could be extracted from open sources.

We would recommend to keep track of new technologies like Facebook's Graph search and social media aggregation services like Pipl. Those can be an improvement to our model in terms of information sources and precision.

It is necessary to analyse how to implement this application in a police organisation. Due to an emerging rule-base and maintenance of the system it would be preferable to implement it on a national scale. This will reduce the overhead of the system.

BIBLIOGRAPHY

National Defense Authorization Act for Fiscal Year 2006. (2006, January 6).

(2011). *Verduurzaming iRN/iColumbo.* Nationaal Coördinator Terrorismebestrijding en Veiligheid.

Baldini, N., Neri, F., & Pettoni, M. (2007). A multilanguage platform for Open Source Intelligence. *Data Mining and Information Engineering*.

Balduzzi, M., Platzer, C., Holz, T., Kirda, E., Balzarotti, D., & Kruegel, C. (2010). Abusing social networks for automated user profiling. *Recent Advances in Intrusion Detection* (pp. 422-441). Ottawa: Springer.

Bergman, M. K. (2001, August). White Paper: The Deep Web: Surfacing Hidden. *Journal of Electronic Publishing, 7*(1).

Best, C. (2008). Open Source Intelligence. In F. Fogelman-Soulié, D. Perrotta, J. Piskorski, & R. Steinberger (Eds.), *Mining Massive Data Sets for Security* (pp. 331-344). IOS Press.

Brett Crawley, J., & Wagner, G. (2010). Desktop text mining for law enforcement. *2010 IEEE International Conference on Intelligence and Security Informatics (ISI)* (pp. 138-140). Vancouver: IEEE.

Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the Seventh International World Wide Web Conference. 30*, pp. 107-117. Brisbane, Australia: Elsevier.

Central Intelligence Agency. (2005, November 8). *Press Releases & Statements*. Retrieved February 12, 2013, from Central Intelligence Agency: https://www.cia.gov/news-information/press-releases-statements/press-release-archive-2005/pr11082005.html

Chen, T., Kaafar, D., Friedman, A., & Boreli, R. (2012). *Technical Report: An Analysis of Social Footprints Across Multiple Online Social Networks.* Sydney, Australia: National ICT Australia (NICTA).

Colombini, C., & Colella, A. (2012). Digital scene of crime: technique of profiling users. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 3*(3), 50-73.

Demmers, m. (2012, June 19). *LJN: BW8469*. Retrieved February 12, 2013, from De Rechtspraak: http://zoeken.rechtspraak.nl/detailpage.aspx?ljn=BW8469

Facebook. (n.d.). *Graph API*. Retrieved February 12, 2013, from Facebook Developers: https://developers.facebook.com/docs/reference/api/

Friendman-Hill, E. (2003). *Jess in Action: Rule-Based Systems in Java.* Greenwich, USA: Manning.

Geradts, Z., & Sommer, P. (2006). *D6.1: Forensic Implications of Identity Management.* Future of Identity in the Information Society. Future of Identity in the Information Society.

Häsel, M. (2011, January). Opensocial: an enabler for social applications on the web. *Commun. ACM, 54*, 139-144.

He, B., Patel, M., Zhang, Z., & Chang, K. C.-C. (2007). Accessing the deep web. *Communications of the ACM, 50*(5), 94-101.

Huber, M., Mulazzani, M., Leithner, M., Schrittwieser, S., Wondracek, G., & Weippl, E. (2011). Social snapshots: digital forensics for online social networks. *Proceedings of the 27th Annual Computer Security Applications Conference* (pp. 113-122). Orlando: ACM.

Kulak, D., & Guiney, E. (2003). *Use Cases: Requirements in Context (2nd Edition).* Addison-Wesley Professional.

Long, J. (2008). *Google Hacking* (Vol. 2). Elsevier.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.

Oerlemans, J., & Koops, B. (2012, September). Surveilleren en opsporen in een internetomgeving. *Justitiële verkenningen - Politie Anno 2012*, pp. 35-49.

Oosterveer, D. (2012, Augustus 14). *Social media in Nederland: de halfjaarcijfers van 2012*. Retrieved February 12, 2013, from Marketingfacts: http://www.marketingfacts.nl/berichten/social-media-in-nederland-de-halfjaarcijfers-van-2012/

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford InfoLab. Stanford InfoLab.

Perito, D., Castelluccia, C., Kaafar, M. A., & Manils, P. (2011). How Unique and Traceable Are Usernames? In S. Fischer-Hübner, & N. Hopper, *Privacy Enhancing Technologies* (pp. 1-17). Berlin: Springer Berlin Heidelberg.

Pouchard, L. C., Dobson, J. M., & Trien, J. P. (2009). *A Framework for the Systematic Collection of Open Source Intelligence.* Association for the Advancement of Artificial.

Schonfeld, U., & Shivakumar, N. (2009). Sitemaps: above and beyond the crawl of duty. *Proceedings of the 18th international conference on World wide web* (pp. 991-1000). Madrid, Spain: ACM.

Stocky, T., & Rasmussen, L. (2013, January 15). *Introducing Graph Search Beta*. Retrieved February 12, 2013, from Facebook - Newsroom: http://newsroom.fb.com/News/562/Introducing-Graph-Search-Beta

Zainudin, N. M., Merabti, M., & Llewellyn-Jones, D. (2011). A Digital Forensic Investigation Model and Tool for Online Social Networks. *12th Annual Postgraduate Symposium on Convergence of Telecommunications* (pp. 27-28). Liverpool, UK: Networking and Broadcasting (PGNet 2011).

APPENDIX I: DATA SOURCES

- Personal data
  - Spyderweb
  - Telefoonboek
- Kadaster
- Profile search engines
  - Pipl
  - Wie o wie
  - 123 people
- Social networks
  - Namechk
  - Hyves
  - Facebook
  - LinkedIn
  - Schoolbank
  - Myspace
  - Twitter
- Search engines
  - Bing
  - Yahoo
  - Google
  - Mama.com
- Online auction/shopping sites
  - Ebay
  - Marktplaats
  - Advertentiezoeker
- IP-address
  - Maxmind
  - Whatismyaddress
- Domain information
  - Whois
  - Central Ops
  - Domain Tools
  - SIDN
- Cache
  - Wayback machine
  - Warrick
- Forums

APPENDIX II: FACEBOOK, A CASE STUDY

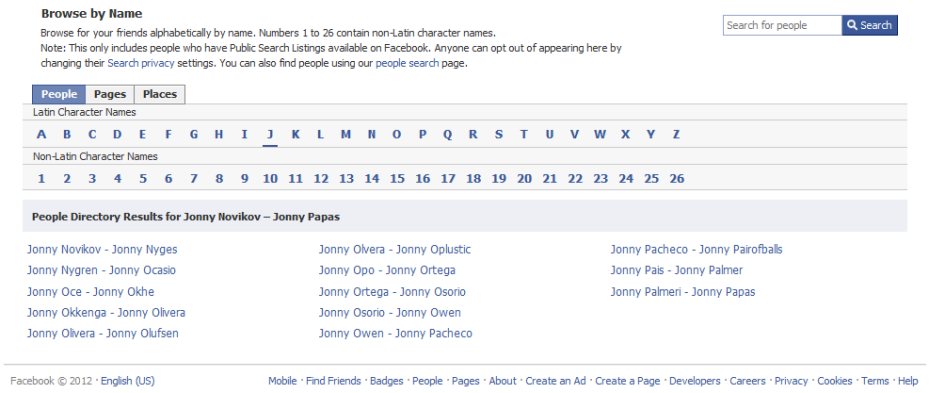
The network with the largest number of users and the most active users is Facebook, which is why we consider Facebook as an interesting candidate for a case study to determine the usefulness of social media networks as a source for criminal profiling.

Web search engine approach

The following different page categories are result of a web search engine query on Facebook.

Directory pages

A directory page is a phonebook like view to browse users; it gives a limited overview of people in a certain alphabetical name range. This might be useful in a crawler/indexer but the directory pages found on Google often do not contain names of users we are looking for. For instance, the screenshot below is the result of a Google/Facebook search on “Jan Jaap Hakvoort”, a known Facebook user that does not occur in any of the directory lists.



Directory pages are therefore not useful in our application if we do not create an index of all users, which we want to avoid for privacy reasons.

Overview pages

There are loads of users with the same name on Facebook, when searching Google for Facebook pages a result is an overview, as shown below. This overview does not only contain matching profiles, it also shows “Pages” and/or “Events” with the same name.

This overview does only show the first few results for public users, logged in users can see all results. In other words, it is not trivial to find all users using overview pages.

Event pages

Public event pages are also returned when performing a Google search on Facebook. The information on a public event page is: invited users (specified as invited/attending/maybe-attending), date(s) of the event and an event description. Public event pages are really useful in the process of finding Facebook usernames because public events do have public lists of invited users that include usernames of protected user profiles.

Stories

A story on Facebook is a post on a public Facebook page that could be about any subject (artists, television shows, football players etc.) that is being followed by users sharing this common interest. Facebook users can comment or like such stories, the usernames of those users are public and can be used to address a specific user profile.

Data extraction

It is possible to find Facebook usernames by searching specifically on public Facebook pages or Event pages. For instance the user “Bret Taylor” (Facebook’s CTO) has attended to a public event that can be found using the Google search engine by entering the following query:

site:facebook.com "Bret Taylor" instreamset:(url):"events"

This results in a list of public events Bret Taylor got invited to but does not provide an explicit verification it is the Bret Taylor we are targeting on. To verify that one should retrieve the list of invitees to retrieve the profile of each single invitee and match the name with the person under investigation.

Remarks

To avoid automated systems from crawling Facebook they use CAPTCHA-protection whenever the number of page requests gets too high. Google blocks automated search queries as well by generating HTTP exceptions. A workaround would be adding a random timer in between queries, which will dramatically slow down the process.

Extracting information from an OSN

The easiest way to get a user's profile information is to use Facebook's Graph API. An (OAuth2.0) authenticated call on https://graph.facebook.com/<username> will return a JSON object of the following form:

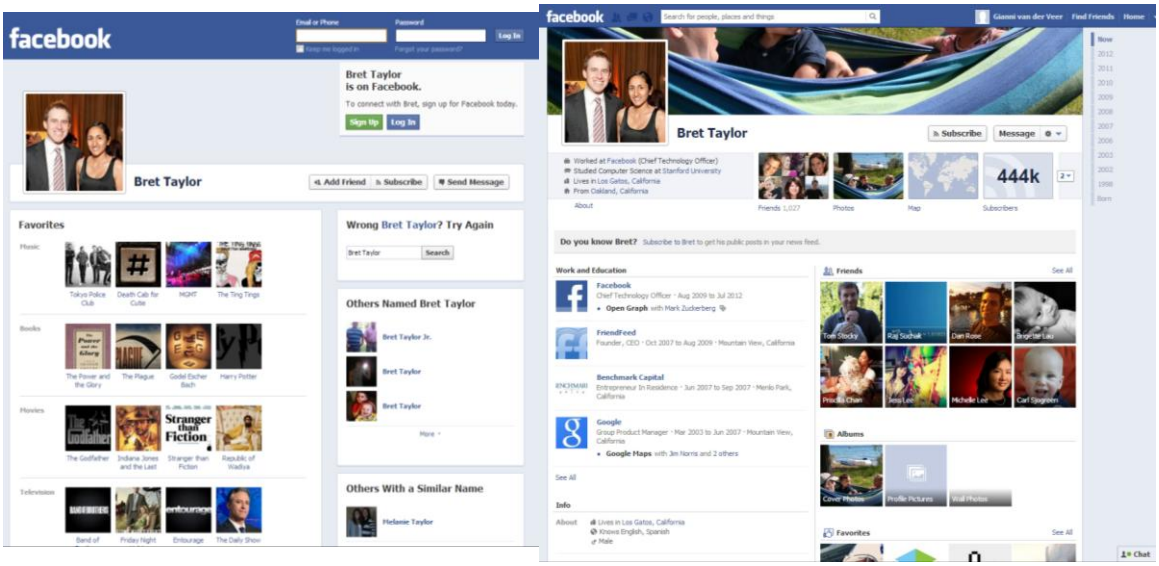
```
{
   "id": "220439",
   "name": "Bret Taylor",
   "first_name": "Bret",
   "last_name": "Taylor",
   "link": "http://www.facebook.com/btaylor",
   "username": "btaylor",
   "gender": "male",
   "locale": "en_US"
}
```

Profile pages

Profile pages show a version of the users profile with only the information that is either marked public by the user or is public for every user (Profile picture thumbnail, name and userid).

Note: profile pages do not contain ALL public information on a user. When logged in or using the Facebook Graph API the default public information (gender, locale, name, first-name, last-name, profile link and Facebook-ID) can be retrieved and additional data can be found.

The following example is showing the Facebook profile page of Brett Taylor, the screenshot on the left shows the public version, the screenshot on the right shows what a logged in user is able to see.



Mobile profile pages

Facebook also has a mobile view of each profile that is accessed through http://m.facebook.com. These pages in general contain less information than the full web profile but do have a much easier layout to parse because no AJAX requests are used.

Public information

For privacy/commercial reasons the public information is limited to:

Name: This helps your friends and family find you. If you are uncomfortable sharing your real name, you can always delete your account.

Profile Pictures and Cover Photos: These help your friends and family recognise you. If you are uncomfortable making any of these photos public, you can always delete it. Unless you delete them, when you add a new profile picture or cover photo, the previous photo will remain public in your profile picture or cover photo album.

Network: This helps you see whom you will be sharing information with before you choose "Friends and Networks" as a custom audience. If you are uncomfortable making your network public, you can leave the network.

Gender: This allows us to refer to you properly.

Username and User ID: These allow you to give out a custom link to your timeline or Page, receive email at your Facebook email address, and help make Facebook Platform possible.

Source: http://www.facebook.com/full_data_use_policy#publicinfo

Facebook's application programming interfaces (API's)

Facebook provides two API's that dig the same information source, the Graph API and the FQL API. The Graph API in general enables the Developer to retrieve data from Facebook without having to parse

HTML from Facebook's pages. In example: http://graph.facebook.com/btaylor will give you a JSON object containing more or less the same information you will see if you go to http://www.facebook.com/btaylor. However, this object does not include, for instance, a profile picture. This can then again be retrieved by the Graph API using another request: http://graph.facebook.com/btaylor/picture?type=large. The complete documentation of this API is available for developers. (Facebook)

Remarks

When performing a search on Facebook API's, protected profiles are not visible, even as an authenticated user. However, if the same user performs the same search query on the Facebook website, while logged in, those protected profiles are displayed, there is a slight gap here.

APPENDIX III: REQUIREMENTS SCENARIO

The police force observes a threatening Tweet from a Twitter user named '@PersonaA'. They let a data analyst use an application to search for other profiles on social media that might concern the same individual. The system will present the data analyst an overview of possible profile matches in an organised manner ordered by relevance. The data analyst validates the proposed profiles and/or profile attributes. the system will use selected attributes to find more data on the Internet generate search queries using the collected profile attributes to find additional information about the suspect and presents them to the user. The user again selects relevant information that is saved in the system. The system will then aggregate all relevant information and presents it to the data analyst. The aggregation of information coming from the same person will support the data analyst to give a better estimation of the seriousness of the threat and the suspect.

In this case the starting point is an online identity, in other cases the starting point can be a physical identity represented by a real name.

APPENDIX IV: PRIVACY IMPLICATION EXAMPLE

In the online report of case BZ0603 on Rechtspraak.nl the court convicts a suspect of threatening and a terrorist act, 'anonymous' Twitter messages are included in the report. However, by executing a search query on a web search engine containing one of the convict's Tweets we can easily find corresponding Twitter profile.