Enabling Open Source Intelligence (OSINT) in private social networks

by

Benjamin Robert Holland

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Co-majors: Computer Engineering, Information Assurance

Program of Study Committee:

Yong Guan, Major Professor

Doug Jacobson

David Weiss

Iowa State University

Ames, Iowa

2012

DEDICATION

I would like to dedicate this thesis to my family, without their support this work would not have been possible. I would also like to thank my friends, advisors, and mentors that have given valuable feedback, peer reviewed, and supported me throughout this work. Finally, I would like to thank everyone that pushed me at times when I needed to be pushed. You have all inspired me to rise to a higher expectation.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those that have helped me with various aspects of conducting research and the writing of this thesis. First and foremost, I would like to acknowledge my major professor Dr. Yong Guan for his guidance and support throughout my academic career and this research. His encouragement has helped to inspire me to undertake tasks of ever increasing difficulty. I would also like to acknowledge the many other faculty members at Iowa State University that have helped to shape my academic career. Their efforts have not gone unnoticed. To those that critically reviewed this work, I owe you a debt of gratitude. You have made this work better as a result. Finally, I would like to acknowledge the multiple Department of Defense Cyber Crime Center mentors that have helped to provide insight into the current state of Open Source Intelligence and social media efforts.

ABSTRACT

Open Source Intelligence (OSINT) has been widely acknowledged as a critical source of valuable and cost efficient intelligence that is derived from publicly available sources. With the rise of prominent social media platforms such as Facebook and Twitter that record and expose a multitude of different datasets, investigators are beginning to look at what social media has to offer the Intelligence Community (IC). Some major obstacles that OSINT analysts often face are privacy and platform restrictions that serve both to protect the privacy of individuals and to protect the economic livelihood of the social media platform. In this work we review existing social networking research to examine how it can be applied to OSINT. As our contribution, we propose a greedy search algorithm for enabling efficient discovery of private friends on social networking sites and evaluate its performance on multiple randomly generated graphs as well as a real-world social network collected by other researchers. In its breadth, this work aims to provide the reader with a broader understanding of OSINT and key concepts in social network analysis.

CHAPTER 1. BACKGROUND

1.1 Open Source Intelligence

Open Source Intelligence (OSINT or OSCINT) is defined as intelligence "produced from publicly available information that is collected, exploited, and disseminated in a timely manner to an appropriate audience for the purpose of addressing a specific intelligence requirement" [36]. The roots of OSINT date back to the Foreign Broadcast Information Service (FBIS) created in 1941, and the field has only been growing since its official establishment by the Director of Central Intelligence Directive in 1994 [26]. Notably, OSINT includes "grey literature" that are unclassified materials that have a limited public distribution. Grey literature includes technical and economical reports, official and unofficial government documents, newsletters, subscription-based journals, and electronic documents that cross-cut political, socio-economic, military, and civilian boundaries.

Table 1.1 Advantages and Disadvantages of OSINT

Advantages	**Disadvantages**
Easy to share (source is unclassified)	Not a full-coverage solution
Does not compromise sensitive sources	Desired information may not be public
Passive activity (low risk)	OSINT often needs to be verified
Broad coverage	Large amount of noise

Table 1.1 shows a few key advantages and disadvantages of OSINT. One primary advantage of OSINT is that intelligence gathering is a passive activity that does not require interaction with a target. The result is that OSINT efforts pose very little risk of alerting an adversary to the presence and motives of the investigator. Another advantage is that public information is easier to share between agencies than classified information and can be pointed to as an alternative source of intelligence that does not compromise a sensitive source that may reveal

a technological or strategic advantage. For many of these reasons and more, in his book, *No More Secrets: Open Source Information and the Reshaping of U.S. Intelligence*, Hamilton Bean has dubbed Open Source Intelligence the "Source of First Resort" because OSINT is rapidly becoming a primary resource utilized by the Intelligence Community (IC) [10].

An important point to remember is that OSINT is not a full-coverage solution. OSINT should be regarded simply as another tool in the intelligence analyst's toolkit. The NATO OSINT Reader outlines the point by making the following metaphor:

> Open source intelligence provides the outer pieces of the jigsaw puzzle, without which one can neither begin nor complete the puzzle. But they are not sufficient of themselves. The precious inner pieces of the puzzle, often the most difficult and most expensive to obtain come from the traditional intelligence disciplines. Open source intelligence is the critical foundation for the all-source intelligence product, but it cannot ever replace the totality of the all-source effort [26].

Finally, perhaps the largest criticism of OSINT is that the information available to the public domain tends to contain a very large amount of noise. A major concern in the Intelligence Community is the increasing amount of difficulty and time required to filter the noise and discover the valuable nuggets of intelligence from the continually growing pool of public information. The amount of information being generated each day is growing at such a fast rate that the NSA has asked congress for funding to build new power plants to power data centers capable of processing the massive amounts of information on the Internet [32]. As pointed out by Eric Schmidt, Google's former CEO, the amount of space it would require to store all of recorded human communications prior to 2003 would be about 5 billion gigabytes, but today that much information is generated every two days, which includes user generated content such as pictures, tweets, instant messages, etc. [37]. In its proposal to congress, the NSA had to invent new units of measurement just to describe the sheer amount of information they anticipate seeing by 2014 [32].

1.2 Online Social Networks

The everyday use of online social networks (OSN) such as Facebook, LinkedIn, and Twitter have seen a steady rise in adoption by since 2005 [19]. Online social networks mirror a subset of our everyday social interactions and contain information that crosses geographic borders. Figure 1.1 shows the most popular social networking sites for each country. Examining the social graph of multiple social networks can help to reveal our everyday habits and current social contacts.

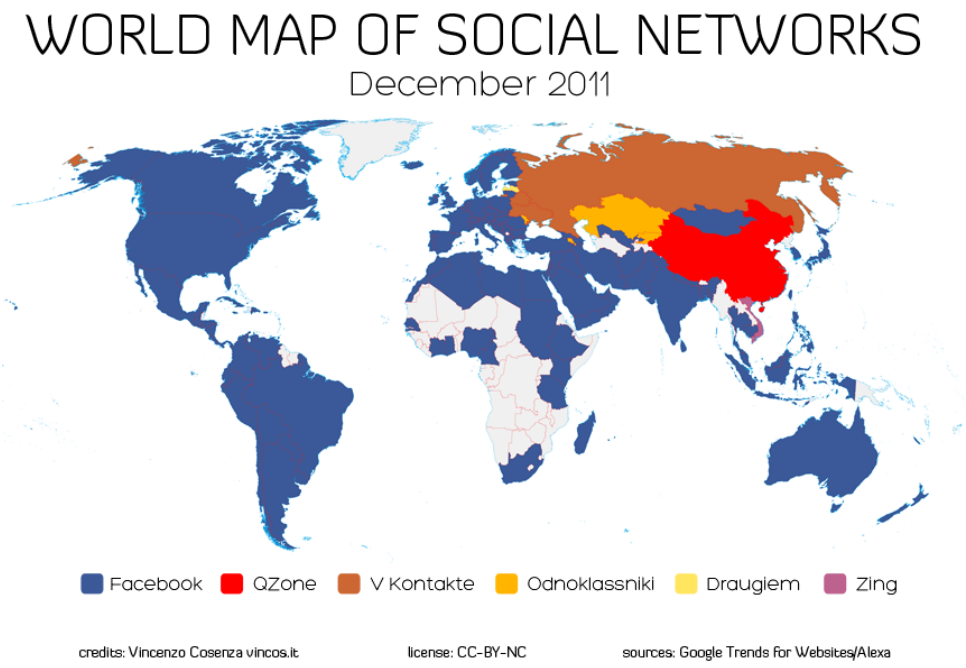

Figure 1.1 World Map of Social Networks

As consumers of social media, we are starting to see an emergence of specialized social networking sites such as Twitter and LinkedIn that provide sets of data that are both overlapping and disjoint from datasets of more general purpose social networking sites such as Facebook . While Facebook and Twitter tend to have casual social interactions, LinkedIn has specialized in professional networking. A survey by the PEW Internet and American Life project, a non-profit think-tank dedicated to uncovering trends in American life, found that more than 50% of online social network users have two or more online profiles [19]. The study revealed that

of those users approximately 80% had profiles on different social networking accounts. The primary uses for the multiple social network accounts was to allow the user to interact with friends on a different social network site or to separate personal and professional contacts.

1.2.1 Case Studies

A 2010 case study by the U.S. Department of Homeland Security [28] found that Jihad and Mujahideen terrorist groups increasingly use Facebook as a medium to disseminate propaganda and to a lesser extent tactical information such as AK-47 maintenance, IED recipes, and remote reconnaissance targeting. It was found that this information was being spread in a variety of languages including Arabic, English, and Indonesian. Extremist groups primarily use Facebook as a means of furthering ideological messages and providing a gateway to other extremist content outside of the Facebook platform. The case study reveals the intentions of the terrorist Facebook accounts after exploring links to outside radical forums. The forums give instructions to would be recruiters to use anonymizing services such as Tor to mask true identities and to use artifice by not revealing sympathy for terrorist groups such as al-Qaeda when interacting with other Facebook users. The case study theorizes that the increased amount of propaganda that appears unmoderated in Arabic is likely due to the lack of resources dedicated to overseeing the language when compared to more popular languages such as English and Spanish. According to the study, while full-fledged terrorist plots may not be revealed on Facebook itself, information leading to and concerning the plot may be partially revealed by close examination of Facebook activity. At the very least sentiment can be detected and measured by Facebook accounts that respond to publicly posted propaganda messages.

In another case study by Northeastern University and the Massachusetts Executive Office of Public Safety and Security [41], law enforcement crime units used online social media such as Facebook, Twitter, and Myspace to track gang related activity. The study revealed that one in three youth gang members would promote their gang on a social medium. The law enforcement agencies surveyed in the study report that they have found online social media services useful for tracking gang related activity and for engaging in public outreach programs for reducing gang related violence. Aside from multiplayer games, social media surveillance

of social networking sites, video sharing sites, photo sharing sites, as well as blogging and microbogging sites have been incorporated into most law enforcement agencies at least at some level.

1.2.2 Obstacles and Limitations

Online social networks have a lot of information to offer OSINT such as social contacts, activities, and personal details of an individual of interest, but contrary to what many might believe, not all information on the web is easily accessible. Investigators are met with several obstacles including privacy and platform restrictions as well as data availability and longevity.

1.2.2.1 Privacy Restrictions

With growing privacy concerns, many social networking platforms have continued to add privacy control mechanisms to restrict access to private information. As of May 2010, Facebook offered an excess of 170 privacy settings and maintained a privacy policy longer than the U.S. Constitution [27]. Despite growing privacy concerns and public calls for increased legislation to enforce the protection of individual privacy, one of the most successful methods of collecting information from users is simply asking for it. A common approach by third party application developers is to create applications that ask the user for unneeded permissions in the hope of gaining additional information. It was found that approximately 50% of adults and more than 75% of teens thought it would be difficult or impossible to find out who they were based on the information available in the restricted profile [19], which may indicate a false sense of privacy when dealing with online social networks. On the other hand, a more recent survey conducted in early 2012, shows that profile pruning and unfriending contacts in the interest of privacy is on the rise [22].

1.2.2.2 Platform Restrictions

Information flowing into online social networks is collected on a massive scale, but is tightly controlled by the social media platform regarding how it flows out. Social networking platforms generally control information flowing out based on social relationships, user-based privacy set-

tings, as well as rate limiting, activity monitoring, and IP address based restrictions. While many access control mechanisms are often put in place to protect the privacy of the user, other mechanisms are often added or intentionally handicapped to protect the economic livelihood of the service platform.

1.2.2.3 Data Availability

As discussed earlier, OSINT by definition has no ability to discover information that does not exist in the public domain. In the worst case scenario, the desired information may have never been gathered by platform, such as when a user chooses not to provide information on a social networking profile. If the desired information was never recorded and is not in the public domain, OSINT has no hope of discovering the desired information. In a more typical case, the target information exists, but privacy and platform restrictions introduce a fog that masks large portions of the desired information. As a result, investigators are forced to attempt to extract the desired information from the digital footprint found outside of the fog that is left behind by user activities.

1.2.2.4 Data Longevity

The social graph is far from static, relationship dynamics change frequently and profiles are updated constantly. Facebook claims that of its 800 million users, over half its users log in at least once a day [34]. Previous studies of the Facebook social graph have limited collection periods to a maximum of one to two weeks (depending on the type of data) [42] [12] to limit the corruption of data due to changes in the social graph during collection time. Monitoring the changes of social content is also an important source of information for understanding the dynamics of the social graph. We can think of each data access as a snapshot in time that is capturing the state of the social graph at collection time.

1.2.3 Legal Issues

While many social networking platforms such as Facebook disallow the use of screen scrapers and other data mining tools through their terms of service agreements, the legal enforceability

of these terms remains unclear. U.S. courts have recognized that in some cases automated web spiders and screen scrapers may be held liable for digital trespassing. Perhaps the best known case is eBay vs. Bidder's Edge that resulted in an injunction against Bidders Edge to stop data mining activities of the eBay platform [16]. Similar court battles have been fought by American Airlines and Southwest Airlines against the online site FareChase, which allows users to compare ticket fare rates after data mining multiple airline websites [5] [39].

A wide range of deep web indexing tools such as Spokeo [4], Pipl [3], Maltego [33], and the Facebook Visualizer [2] continue to exist and data mining cases continue to be fought on a per case basis leaving the matter far from resolved.

Various legal mechanisms such as subpoenas or the U.S. Patriot Act [30] exist so that law enforcement officials can directly access information when needed, but it is important to note that this tactic would not constitute OSINT intelligence gathering because subpoenaed information is outside the realm of public information.

Executive Order 12333, the legal guidance for intelligence oversight, paragraph 2.3 section A, notes that agencies are allowed to collect, retain, and report on information that is publicly available [1]. The order is a bit dated and is subject to interpretation, but some interpretations consider information acquired from authenticated services to be outside the realm of the public. For example, information gathered from the Facebook platform is considered public information, unless it is necessary to log into the Facebook service to obtain the same information. This restriction may be enforced in the United States but is most likely not enforced by adversaries and is of course subject to change in the future. For the purposes of this work we will not consider this restriction.

1.2.4 Ethical Issues

Crawling social networks for personal information is an ethically sensitive area. To justify our work we cite other works that have conducted live crawling experiments [12] [11] [42] [24] [23] and works that have examined ethically sensitive online experiments [20] [21] with human subjects. Furthermore, in our own crawling experiments we have limited crawls to specific social networking accounts that make up a sample toy network that was created for testing this

work in order to prevent our crawler from incurring a major resource overhead to the social networking platform.

In other cases when it was necessary to test our methods on a larger dataset we used randomly generated graphs of relationships between randomly generated personas that do not represent actual identities. We also leveraged anonymized social network datasets that have been made available by other researchers.

In situations when we were unsure of the best ethical action to take, we consulted our University's Institution Review Board (IRB) for guidance. With respect to the data mining functionality of our test framework and the utilities described in the appendix, we would like to stress that our framework does not do any sort of "hacking," it simply deduces information from already public information in an automated fashion.

CHAPTER 2. OBJECTIVE

2.1 Motivation

In the FBI Request For Information made in early 2012, one of the desired capabilities of OSINT was to interface with social media to create "pattern-of-life matrices" [29]. The behavioral analysis of online users would then be used support law enforcement planning and operations, likely including gathering information for a search warrant or otherwise collecting actionable intelligence such as a target's daily routines and contacts.

A common approach to determining behavioral patterns of an individual is to look at the individual's daily contacts, which are likely to be mirrored at least partially in online social networks. However, with the recent increased utilization of privacy mechanisms on social media platforms, it becomes more likely that the target's contact information is obscured from the public domain through privacy restrictions put in place by the user. In a random sampling study of Facebook it was found that about one in four accounts have privacy settings enabled [12]. Many social networks, such as Facebook, represent relationship information as an undirected edge between two nodes in a graph. We can discover a target's set of private friends by looking for accounts that list the target as a friend, meaning we only need to discover one of the two accounts in the friendship relationship to determine the connection exists (assuming that one account is public).

Searching a social graph for friend nodes has a cost associated in terms of the number of Application Program Interface (API) calls to the social media platform. Making an excessive number of API calls in a short amount of time will exceed rate-limiting thresholds, which denies further interaction. Making matters worse, social media platforms often attempt to prevent this type of activity through IP address and account banning, so it is important to use API

calls efficiently.

To overcome these problems, we propose a greedy search algorithm to minimize the number of API calls needed to return the maximum number of friends for a given private profile. We evaluate the performance of our algorithm by comparing the percentage of friends discovered after each newly discovered friend and the ratio of the number of API calls made to the number of total nodes in the graph.

We compare our algorithm to the optimal case as well as Breadth-First Search (BFS) and Depth-First Search (DFS) on sets of randomly generated graphs and sets of real-world social networks. We then randomly privatize nodes in the graph to replicate the privacy situation found in online social networks and compare the performance of our algorithm to BFS and DFS.

In short the following goals are set for this thesis:

- Propose and implement an algorithm for efficient discovery of private friends
- Evaluate the performance of the proposed searching algorithm against the optimal case and existing crawling methods BFS and DFS

The breadth of our work has lead us to investigate several issues specific to social networking analysis and OSINT. During our investigations we created a framework to efficiently manage multiple social networks and interact with online social media. We believe that the lessons we have learned from this activity may be useful to others looking to do future work in this area so we have included several sections in the appendix of this work dealing with cross-correlation of online social networking accounts, context preserving graph databases, and data collection in online social networks.

In short, the breadth of our work aims to:

- Develop a set of utilities for collecting data from online social media for operations that are not supported by an official API but are required for OSINT gathering
- Develop a framework for storing multiple social graphs while preserving the context of the network from which the graph was collected

- Implement a system to create cross-correlation identity mappings between a set of social graphs

The means to achieve the the breadth and depth of our goals listed in this section are described in the following chapters, as outlined in the Thesis Overview section.

2.2 Thesis Overview

The remainder of this thesis is structured as follows. Chapter 3 discusses related work in social network analysis leveraged by our work. Chapter 4 presents our proposed algorithm and defines and evaluates its performance. Chapter 5 discusses the results of our algorithm on a single network created from a random graph model and on a real world social network. Chapter 7 concludes this thesis and provides possible directions for future work.

CHAPTER 3. RELATED WORK

In this chapter we review related work that is needed to understand the work presented in the rest of this thesis. First we discuss how researchers access the underlying social graphs of online social networking sites and then we discuss some common metrics used to describe the properties of social graphs. We then review three random graph models that are commonly referenced when dealing with social graphs.

3.1 Accessing Social Networks

When accessing social networks researchers are at a disadvantage because only small portions of the social graph are visible at one time. The rest of the graph is shrouded in a fog that masks network nodes until crawling algorithms discover them. Platform restrictions add a new set of challenges to accessing a social graph. In the case of Facebook, extended exploration of friend-of-friend relationships is removed from the platform API, so for this work we created a headless browser to simulate the actions normally taken by a user. We have placed a sequence diagram of this function in the appendix for the reference of the reader. The next sections discuss random sampling and properties of well-known search algorithms Breadth-First Search (BFS) and Depth-First Search (DFS).

3.1.1 Breadth-First Search

A Breadth-First Search visits all successors of a visited node before visiting any successor of a child node. This functionality is implemented as a queue (first-in first-out) where newly discovered nodes are placed at the end of the queue and search order is determined by removing the first item from the queue. A BFS is guaranteed to return an optimal (shortest) path if a path exists to a given target node from the starting node. According to Gjoka [18], a BFS

is biased towards nodes with higher degrees because higher degree nodes have more incoming links than lower degree nodes when considering an undirected graph.

3.1.2 Depth-First Search

A Depth-First Search is similar to a Breadth-First Search except it uses a stack (first-in last-out) in place of a queue. DFS is less commonly used to crawl online social media because it tends to crawl in and out of social clusters (by following a friend of a friend of a friend and so on) whereas a BFS will explore all friends of a given node systematically before moving on to another node's friends. Neither BFS or DFS make use of any prior knowledge about the structure of the graph, leaving plenty of room for optimizations to improve on search times and the amount of resources consumed by the crawler to find the target.

3.1.3 Random Sampling

Another graph exploration approach is to randomly sample nodes in the social graph by guessing the identifier of the node. In most online social networks there is no way to knowingly generate valid account identifiers without some form of rejection sampling. Before Facebook expanded the identifier key space to 64 bits from 32 bits a birthday attack was feasible by rejection sampling guesses of random integers between 1 and 2^{32}. After a random integer is generated, it is used to attempt to access the corresponding account. If the account does not exist the sample is discarded. In 2010, given an identifier range of 2^{32} and the number of subscribed users at approximately 2^{29}, a birthday attack succeeds every 1 in 8 attempts ($\frac{2^{29}}{2^{32}} = \frac{1}{8}$) [12]. Unfortunately for researchers, this technique is unfeasible on Facebook for the new 64-bit identifier key range because the number of Facebook accounts is still not dense enough at this time. We can make the attack feasible again by exploiting some additional known facts about Facebook identifiers. Knowing previous identifier ranges and that identifiers are permanently tied to an account allows us to restrict the random generator in a manner that increases the odds of guessing a valid account identifier but still maintains a uniform sampling [18].

3.2 Properties of Social Networks

There are many metrics used to described graph structures, but when dealing with social networks researchers tend to describe social graphs in terms of size, average shortest path, clustering, and degree distributions. This section briefly reviews each metric in order to build a foundation for the rest of this work.

3.2.1 Size

The number of nodes in a graph usually defines the size of a social network. Consequently, size is an exemplary metric for expressing the scope of information contained in a graph. At the time of this writing, Facebook hosts over 800 million active social network profiles [34]. Managing large social graph relationships resource intensive. A 2010 estimate of the overhead required to crawl the Facebook social graph was roughly 44 Terabytes of data [12]. Another useful metric is the number of edges in the graph. With the set of edges E and the set of vertices V we can calculate graph density (a measure that represents how close a graph is to containing the maximum number of edges) for an undirected graph as:

$$density = \frac{2|E|}{|V|(|V|-1)}$$

3.2.2 Degree Distribution

Aside from knowing the number of nodes in a network, we often want to describe the distribution of friends (node degrees) among all nodes in the network. A degree is defined as the number of adjacent neighbors connected to a given node. Many social networks have a degree distribution that asymptotically follows a power-law distribution as in the following relation (where k is the degree and λ is a variable parameter usually $2 < \lambda < 3$) [13]:

$$P(k) \sim k^{-\lambda}$$

Graphs with degree distributions that follow a power-law are called scale-free networks and are commonly explained using a preferential attachment model. In a preferential attachment

model the rich (nodes with high degrees) get richer (gain additional degrees faster) than the poor (lesser degree) nodes. This property leaves a signature linear plot on a log-log scale as shown in Figure 3.1.

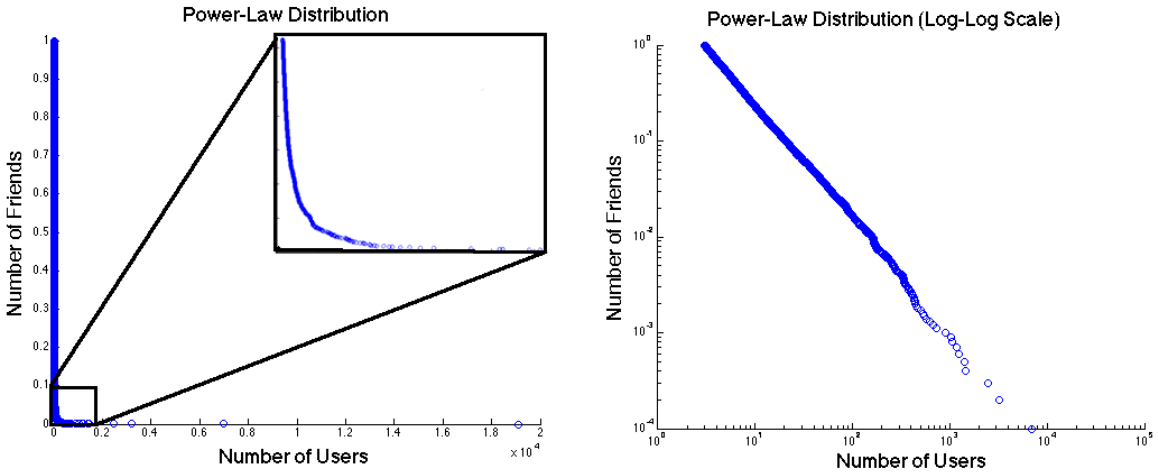

Figure 3.1 Power-Law Distribution

3.2.3 Average Shortest Path

The average shortest path is the average of the minimum length path (also know as a geodesic) required to connect a given pair of nodes for each node pair in the graph. When N is the number of nodes in the graph and $d(v_s, v_t)$ is defined as the shortest distance from vertices v_s to v_t (distance is 0 if $v_s = v_t$ or if a path does not exists between v_s and v_t) the average shortest path is defined as:

$$L = average\ shortest\ path\ length = \frac{1}{N(N-1)} \sum_{\forall v_s, v_t \in V} d(v_s, v_t)$$

In social networks, a small-world graph is a graph that has an average shortest path length that scales proportionally with the logarithm of the number of nodes in the graph [40].

$$L \propto \log N$$

This small-world phenomenon is well observed in social networks and is sometimes referred to by the six degrees of separation concept that is associated with the work done in the Stanley

Milgram small-world experiment [38]. A trivia game called the six degrees of Kevin Bacon [7] is based on linking the actor Kevin Bacon to another actor through no more than six connections where each connection is a movie where two actors appeared together.

3.2.4 Clustering

Social networks tend to form definitive clusters of nodes that are made up of tight knit groups of highly connected nodes [40]. Following the work of Duncan Watts and Steven Strogatz, define an undirected graph $G = (V, E)$ to be a graph of the set of vertices V and the set of edges E. Supposing that a vertex i has k_i neighbors, it follows that a node can have at most $\frac{k(k-1)}{2}$ edges. The local clustering coefficient C_i for a vertex i is defined as the fraction of edges that actually exist out of the set of possible edges a vertex can have. The average clustering coefficient for the entire graph is defined as:

$$C = \frac{1}{N} \sum_{i=1}^{N} C_i$$

In the context of friendship networks, the clustering coefficient C_i reflects the measure of neighbors connected to vertex i where 1 is a connection to all neighbors in the neighborhood and 0 is a connection to none of the neighbors.

3.3 Random Graph Models

Due to the increased computing power and storage capabilities of modern computers it is now possible to create large data sets for modeling real networks. In this work we use randomly generated graph models to produce test data with varying properties that represent characteristics of online social networks. One pitfall of using randomly generated graphs is that current graph models only partially represent observed graph characteristics and fail to accurately portray social graph structures. In the following sections we will review three of the most common random graph models examined by social network analysis research.

3.3.1 Erdős-Rényi Model

The Erdős-Rényi model, first proposed in previous works [15] [14], is defined as the graph $G(n, p)$, where n is the number of nodes and p is the probability that an edge exists between each possible pair of vertices (independent from every other edge). The model is expected to produced $\binom{n}{2}p$ edges. For each vertex in the graph there exists an edge with probability p between the remaining $n-1$ other vertices creating the binomial degree distribution for the probability that a vertex has a degree k:

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

While easy to generate, the Erdős-Rényi model does not produce the strong clustering relationships between groups of nodes that is commonly found in online social networks, and the model's binomial degree distribution is unlike most real-world networks. In social network analysis, many researchers tend to use models such as the Watts-Strogatz model or Barabási-Albert model discussed in the next sections, which better represent the characteristics commonly observed in social networks.

3.3.2 Watts-Strogatz Model

The Watts-Strogatz model was created to produce graphs with strong clustering coefficients and average shortest path lengths similar to what is found in social networks. The Watts-Strogatz model is defined a graph $G(n, k, p)$, where n nodes are arranged in a lattice ring with $\frac{k}{2}$ neighbors connected on each side. The parameter p is a probability used to randomly rewire the end of each edge (where the start of the edge is the node i and the end is some other node) for each node in the graph. A sample graph at each stage of this process is shown in Figure 3.2. Note that the graph in the left side of Figure 3.2 is before the rewiring step but is equivalent to the resulting graph of parameters $G(6, 4, 0)$.

The Watts-Strogatz model is expected to create a mean degree k for each node and $\frac{nk}{2}$ edges between the total set of n nodes in the graph. As probability p is varied from $p = 0$ to $p = 1$ the graph approaches an Erdős-Rényi random graph. From the *Collective dynamics*

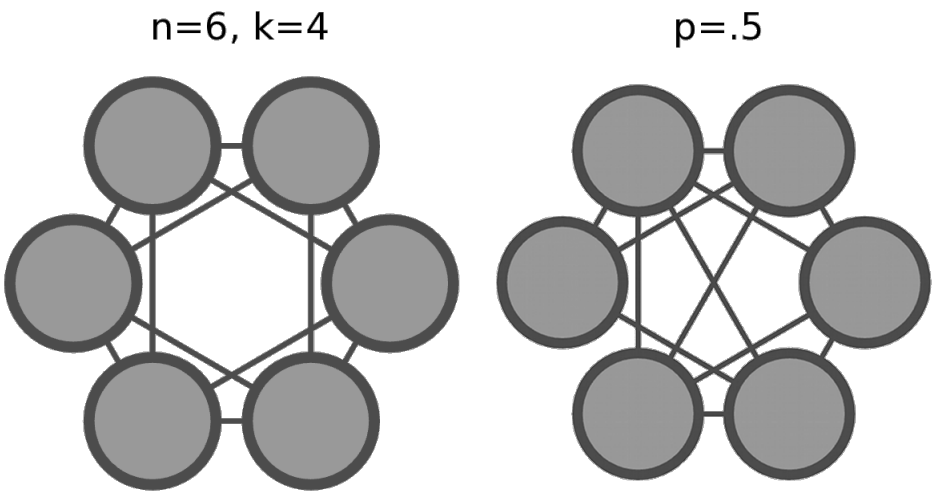

Figure 3.2 Watts-Strogatz Random Graph Model

of 'small-world' networks paper [40] that first proposed the model, we find that the average path length scales linearly from $\frac{n}{2k}$ when $p = 0$ to $\frac{ln(n)}{ln(k)}$ when $p = 1$. Furthermore, it was found that the average clustering coefficient is: $C = \frac{3}{4}$ when $p = 0$, $C = \frac{k}{n}$ when $p = 1$ and $C(p) \sim C(0)(1-p)^3$ for values of p $0 < p < 1$. Finally, degree distribution is found to be a sharply peaked curve centered on k when $p = 0$ and a Poisson distribution function when $p = 1$ [9]. It should be noted that the Watts-Strogatz model does not produce the desired power-law distribution described earlier, which brings us to look at our final model, the Barabási-Albert model, described in the next section.

3.3.3 Barabási-Albert Model

The Barabási-Albert model was designed to generate random scale-free (power-law distributed) networks. The variation of the model that we consider in this work is defined as a graph $G(n, m_0, m)$ where:

- n is the number of desired nodes
- m_0 is a set of initial nodes
- m is the number of new edges to consider for each additional node added to the graph

At run time, the algorithm repeatedly adds a new node and connects the new nodes to m existing nodes with a probability proportional to the number of edges each existing node has until the graph contains n nodes. As a result, the existing nodes get preferential treatment over nodes added later in the process creating a rich get richer effect. The result of a run with $G(n = 50, m_0 = 1, m = 1)$ (shown in Figure 3.3) illustrates how early nodes tend to become hubs of activity for neighboring nodes.

According to the model's authors, the degree distribution is a power-law distribution of the form $P(k) \sim k^{-3}$ [8] and the average path length is $L \sim \frac{ln(N)}{ln(ln(N))}$ [6]. An analytical prediction for the coefficient properties of the model has yet to be determined, but empirical results indicate that the clustering coefficient is stronger than a Erdős-Rényi random graph but decays with the increase of n, making the clustering distinct from small-world networks [6].

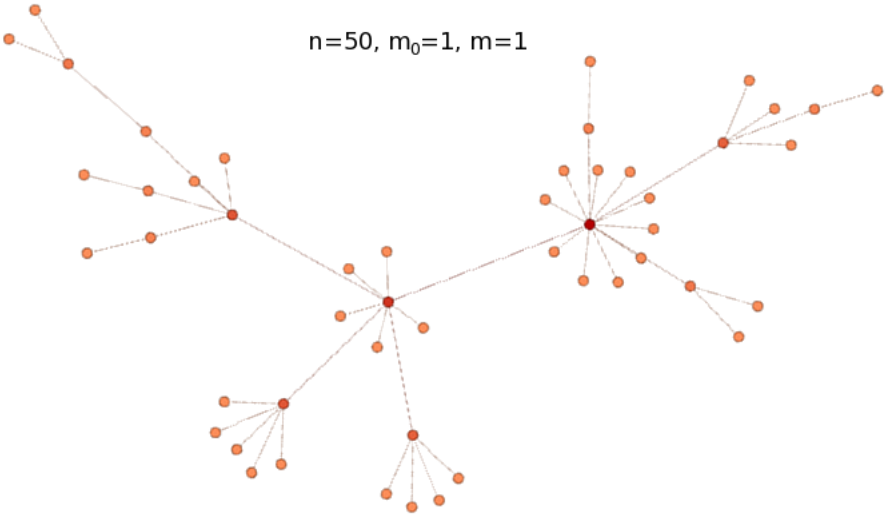

Figure 3.3 Barabási-Albert Random Graph Model

3.4 Summary

In this chapter we discussed existing graph exploration techniques and key concepts in social network analysis. We examined BFS, DFS, and random sampling and laid the foundation to provide a search optimization to take advantage of known graph properties such as size, degree distribution, average shortest path, and clustering to reduce search overhead. We discussed

three random graph models, Erdős-Rényi, Watts-Strogatz, and the Barabási-Albert model for to generate known graph properties.

CHAPTER 4. ALGORITHM

4.1 Goals

Investigators commonly need to verify and expand on the known associates of a target for vetting a security clearance, verifying an alibi, or general target profiling. Online social networks provide a cost effective means to gather contacts of an individual, but with a target's privacy protections enabled many valuable contacts may be obscured from an investigator. By examining the digital footprints of activity left behind online it is still possible to deduce obscured friend information by interrogating a target's neighbors and checking for friendship connections to the target declared by the neighbor.

Basic search strategies such as BFS and DFS ignore graph topologies that can be exploited to reduce the amount of nodes required to unmask the majority of a target's friends. We define an evaluation metric of one Application Program Interface (API) call to be the cost of expanding a single node. Traditionally, an API call is the cost of interacting with an online social network to query information for a given node in the social graph. In many online social networking platforms requests can be batched together to count as a single interaction with the service, but because of the fog that the surrounds the graph at runtime many nodes will not be discovered in time to batch requests efficiently. Investigators are prevented from trying to brute force guess the friends of a target through rate limiting, access request quotas, and the sheer size of the social graph. Thus, the goal of this algorithm seeks to use the least number of API calls necessary to gather the largest return of private friends belonging to a target. Therefore improving search return efficiency while maintaining accuracy.

4.2 Model Assumptions

We assume that online social networks such as Facebook have properties similar to the Watts-Strogatz small-world random graph model. Previous works [40] [12] have show that the average shortest path length and clustering coefficients of real-world social graphs can be reasonably well modeled by the Watts-Strogatz small-world graph. It is known that the Watts-Strogatz model does not reproduce the power-law degree distributions found in most observations [13] of social networks, but at the time of this writing we could not find a random graph model that could accurately represent the set of all properties observed in real world social networks. Our algorithm leverages mutual friend relationships and group clique formation that we find present in online social networks such as Facebook [18], which of the models we evaluated, the Watts-Strogatz model demonstrated best. To evaluate our algorithm's performance we run it on datasets generated using the Watts-Strogatz random graph model.

4.3 Intuitions

Consider this scenario. Bob, a high school student, has ten friends at school but won't tell us who his friends are. Even though Bob won't say who his friends are, we can still find the same information by walking around the school and asking students if they are friends of Bob. Assuming every student at the school either tells the truth or refuses to answer the question, we can exploit known social structures of the school to create an efficient search strategy to reveal all of Bob's friends.

Social graphs tend form large numbers of triadic closures of mutual friends [31], which means that if we ask Bob's best friend Jim (or any of Bob's friends for that matter) who their friends are, we are likely to find a set of friends that are mutually shared between both Bob and Jim. Likewise, it is more likely that a friend of Bob's friend Jim is also a friend of Bob, as opposed to another randomly selected individual. We can exploit the mutual friendship property of social graphs by asking newly discovered friends who their friends are and checking to see if Bob and the newly discovered friend share mutual friends that we don't already know about. We use the number of times an individual has been observed as a friend of a friend of

the target as a metric to describe the priority in which we should interrogate friends of friends.

Social graphs also have strong clustering properties [40], so let's now consider that the school is made up of various clusters of students (e.g. the math team, the chess club, the football team, the band, etc.) and that we know Bob is the quarterback of the football team. It makes sense that we would start by asking members of the football team if they are friends of Bob because the members of the football team are more likely to be friends with Bob. The members of the football team are more likely to be friends with Bob because of the tight clustering of mutual friends between the football team clique. For the members of the math team, which share most of their mutual friend connections with other members of math team, a member is less likely than a member of the football team to make connections to members in the football team cluster. That's not to say that Bob and Mike (a member of the math team) can't be friends, it just that Bob is more likely to have more friends on the football team than the math team.

Let's pretend that we don't know that Bob is the on the football team. It makes less sense to interrogate the entire math team to see if they are friends with Bob than if we just randomly picked students from the school to ask if they are friends of Bob because the chance that a random student belongs to the same clique as Bob (which has members highly connected to Bob) is better than the chance that one individual in the math team is a friend of Bob. When we don't know what cluster Bob belongs to, it's better to cast out a wide net and interrogate individuals from each cluster (the math team, the football team, the chess club, the band, etc.) until we know what cluster Bob belongs to than just systematically asking everyone on the school's roster if they are friends of Bob or by systematically exploring each clique until we find Bob's clique.

To explain it another way, imagine that we have the graph of nodes A, B, C, and D as seen in Figure 4.1. In a Breadth-First Search node A is expanded and discovers nodes B, C, and D and then node B is expanded to discover node C (for the second time). If we know an average number of friends that each node has we can calculate the number of new nodes we expect to discover for each node as the average node degree minus the number of times we have discovered the node. We subtract the number of times we have seen a node because

each observation is made from a node that would be returned in the overall set of expanded nodes. In order to maximize the number of new nodes discovered with each node expansion we should choose to expand nodes that we have seen the fewest times first. In the context of mutual clusters of friends this has an effect crawling away from clusters towards other clusters. Reversing the priority biases the crawl order towards staying within a cluster until the entire cluster has been explored. In our proposed algorithm we use this as a secondary heuristic to maximize search potential when searching for the target.

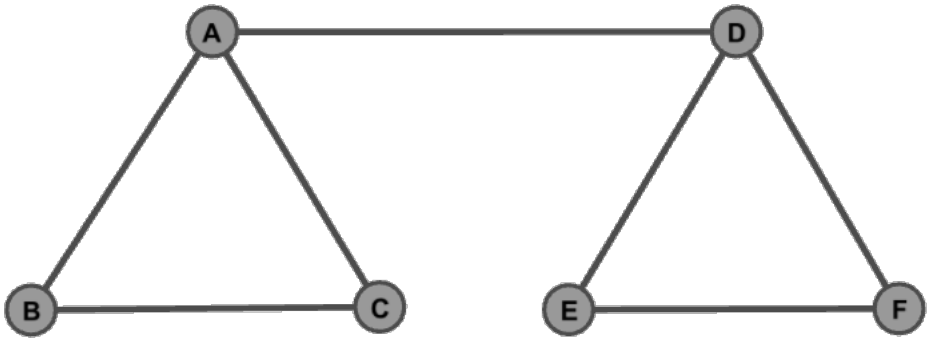

Figure 4.1 Example Graph

In other words, our algorithm does the following:

(a) Searches friends of friends for mutual friends of target

(b) Maximizes search potential to discover target

4.4 Algorithm

Given the information available to the algorithm at runtime, the algorithm behaves in one of two modes: hunter or seeker. In hunter mode, the algorithm has expanded a friend of the target and searches the friend's friends discovered for mutual friends of the target (giving the most discovered nodes of a known target friend search priority). In seeker mode, the algorithm has no friends of friends left to explore and attempts to maximize its potential of finding a new friend of the target by prioritizing the exploration of nodes expected to reveal the highest number of new nodes.

The private friend discovery algorithm shown in Listing 4.1 uses a priority queue with the properties listed below to manage the order of node exploration.

- Items in the queue are sorted by their priority rankings
- Priority ranking is determined by two scores (a primary score and a secondary score)
- A primary score for an item increases each time the item is added to the queue by a value defined on each add
- A secondary score for an item increases by one each time the item is added to the queue
- A higher primary score always takes precedence over a secondary score
- A lower secondary score is used to break ties between the primary scores
- A tie between secondary scores is broken by order of arrival in the queue

Listing 4.1 Private Friend Discovery Algorithm

```
friends = map(origin, target){
  if(origin == target)
    return error // origin is target and target is private
  pq.enqueue(origin, PRIORITY_0)
  neighborhood = {}
  searched = {}
  while(pq.size > 0){
    node = pq.dequeue()
    if(searched.contains(node)){
      continue // skip node
    } else {
      neighbors = node.expand()
      if(neighbors.contains(target)){
        neighborhood.add(node)
        pq.enqueue(neighbors, PRIORITY_1)
      } else {
        pq.enqueue(neighbors, PRIORITY_0)
      }
```

```
        searched.add(node)
      }
    }
    return neighborhood
  }
```

CHAPTER 5. RESULTS

5.1 Test Framework

To test our algorithm's performance we generate Watts-Strogatz random graphs using the advanced graph model generator plugins available for the Gephi graph visualization utility (*http://www.gephi.org*). We then import the models into our database framework for storing multiple social graphs (described further in the appendix). Our test platform simulates the role of an online social network by restricting the set of node neighbors visible for a given node based on a single privacy restriction that can be enabled or disabled for each node in the database. At run time we import a new random graph, enable privacy for the target node, and run a test harness that chooses two random start locations (one in the set of the target's friends, and one outside the set of the target's friends) and records the results of a BFS, DFS, and our algorithm for both types of starting locations. The test harness is run over twenty or more iterations where each iteration records a pair of data points for each new friend discovered. The pair of data points corresponds to the ratio of the number of API calls used to the total number of nodes in the graph and the ratio of the number of friends discovered to the actual number of friends connected to the target node. After each iteration, the start location privacy setting is disabled to prevent metric bias in the algorithm. The results are then serialized to a file and exported to Matlab to be analyzed as histograms and scatter plots.

5.2 Expectations

Ideally, in the optimal case, a perfect algorithm would discover every private friend using the absolute minimum required API calls in the process. If we say that the target has k friends, then we find that for each node in the friends list we must make one API call to confirm the

friend is indeed a friend. We must then add to k the minimum number of nodes required to reach each friend from a given start point to cover the optimal case. The minimum number of nodes is defined as the minimum spanning tree that includes the start node and each of the required friend nodes. The rest of the nodes may optionally be included in the minimum spanning tree if they are required to complete a path between two nodes in the tree. This problem is classified as the node-weighted Steiner tree problem [17] and is computationally expensive to compute. For simplicity purposes we will assume the optimal case is defined by an algorithm that is either extremely lucky or has access to an all knowing oracle that reveals the friend list to the search agent. Upon learning the friends list, the search agent simply needs to confirm each friend at a rate of one friend per API call. On a scatter plot like Figure 5.1 the optimal case would be a nearly vertical sloped line with the x coordinate starting at $1/|n|$ API calls and moving to $|k|/|n|$ API calls and the y coordinate starting at $1/|k|$ friends ending at $|k|/|k| = 1$ friends.

Our expectation of our algorithm is that we approach the optimal case once a target friend is discovered and that we on average do better than BFS, which is the standard search method for crawling social networks at this time.

5.3 Comparison to Existing Algorithms

To verify that our algorithm indeed works in this suggested hypothetical best case scenario (i.e. when all nodes are public except for the target), we created a small test network of 1000 nodes using Watts-Strogatz model parameters of $G(1000, 130, .2)$, which can easily be completely explored by BFS and DFS. We justify our average degree $k = 4$ by equating it to the average number of friends a user has on Facebook (a statistic shared by Facebook during their F8 Developer Conference). We justify a value of $p = .2$ by intuitively reasoning that less than half of friendship connections on Facebook are comprised of random connections. Also, $p = .2$ generates a model with an expected average clustering coefficient of $C_i = .384$, which is within the range observed by previous work [18].

Examining the scatter plot results in Figure 5.1 and Figure 5.2 shows that our algorithm using the hunter-seeker strategy consistently performs better than both BFS and DFS regardless

of the starting location. When the starting location is moved outside of the target neighborhood both BFS and DFS are negatively impacted in terms of API calls per friends discovered, but the hunter-seeker algorithm remains relatively unaffected.

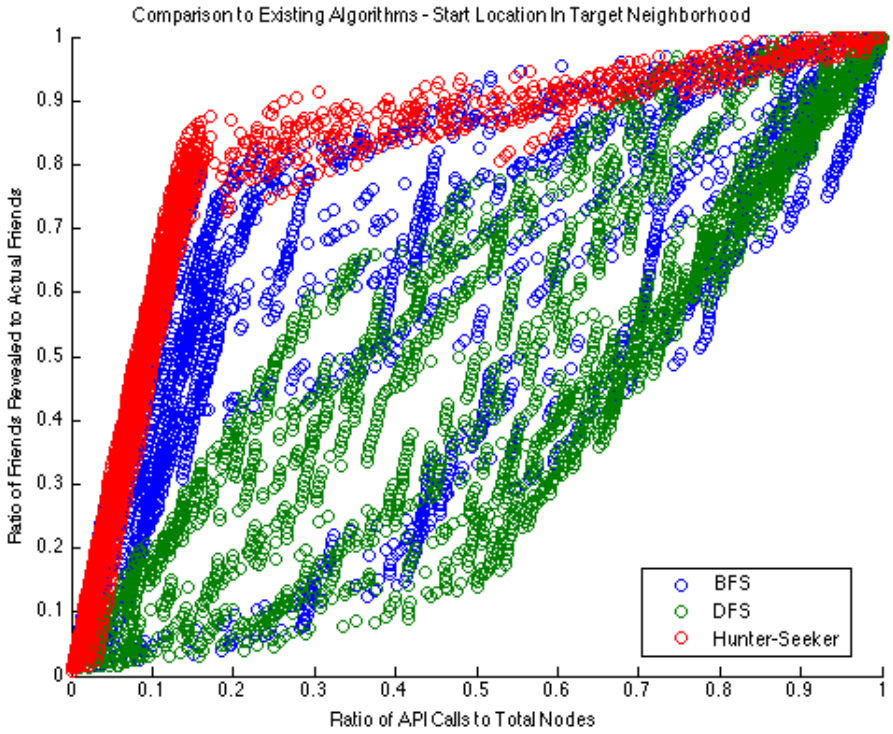

Figure 5.1 Comparison to Existing Algorithms - Start Inside of Target Neighborhood

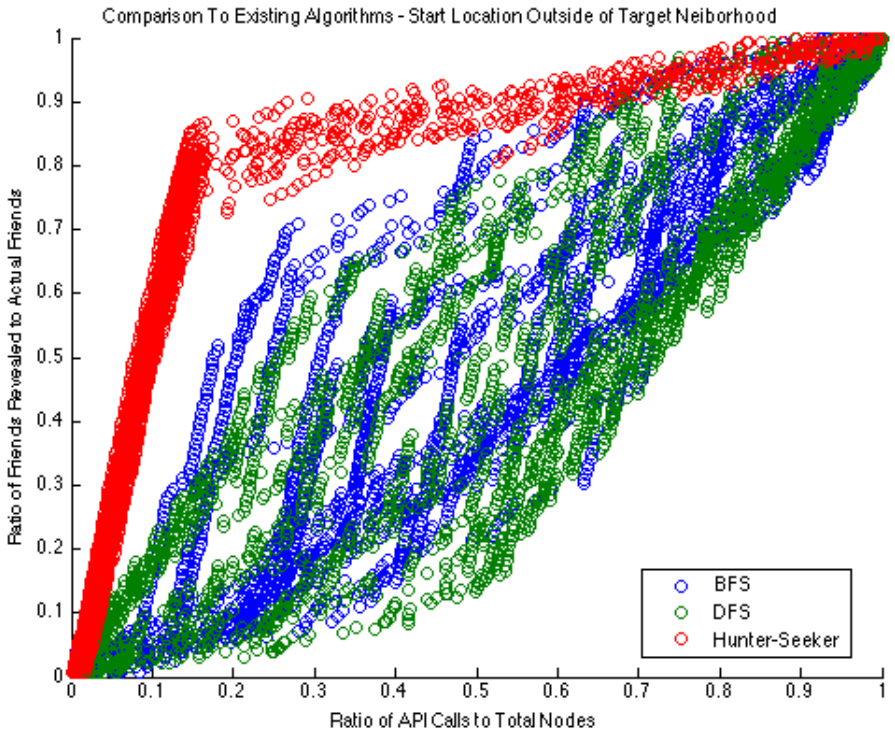

Figure 5.2 Comparison to Existing Algorithms - Start Outside of Target Neighborhood

Histogram plots of the same graphs (shown in Figure 5.3 and Figure 5.4) for the hunter-seeker algorithm show the number of friends revealed in the first percentages of the number of API calls to the total number of nodes includes the majority of all friends. We find that the algorithm finds most of its target's friends in the first 10-20% of the total nodes and then spends the rest of the algorithm searching for the edge nodes that were not near the target and most likely located in other clusters. At this point in the algorithm, there is a surplus of bad guesses the algorithm could make and very few right answers. A comparison of both histograms confirms that the algorithm will produce similar results for a random location in the graph versus a start location inside the target neighborhood.

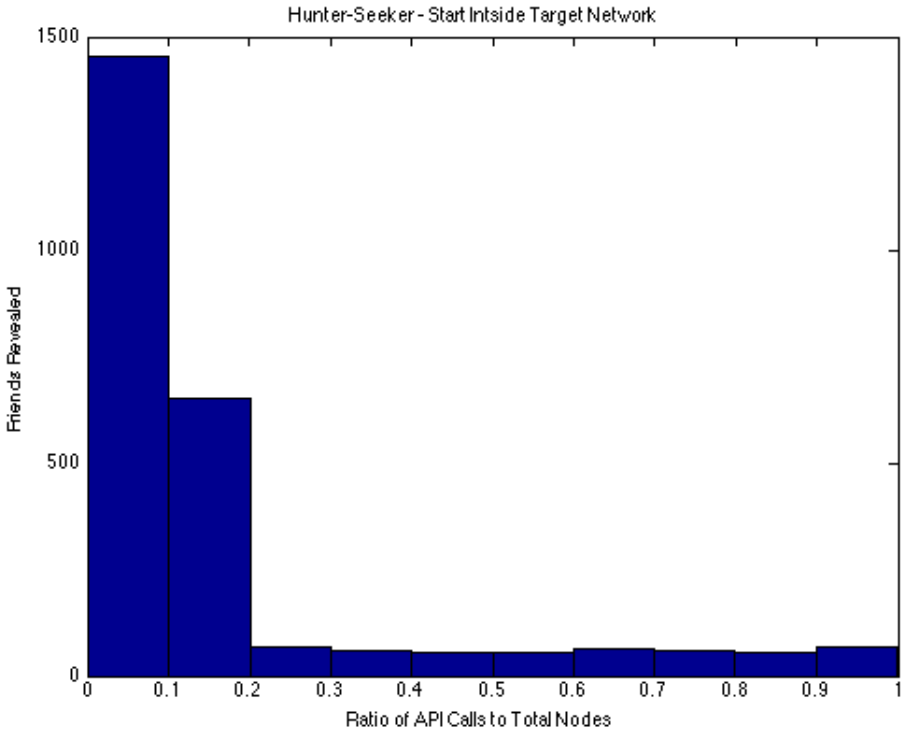

Figure 5.3 Hunter-Seeker Histogram for Start Location Inside of Target Neighborhood

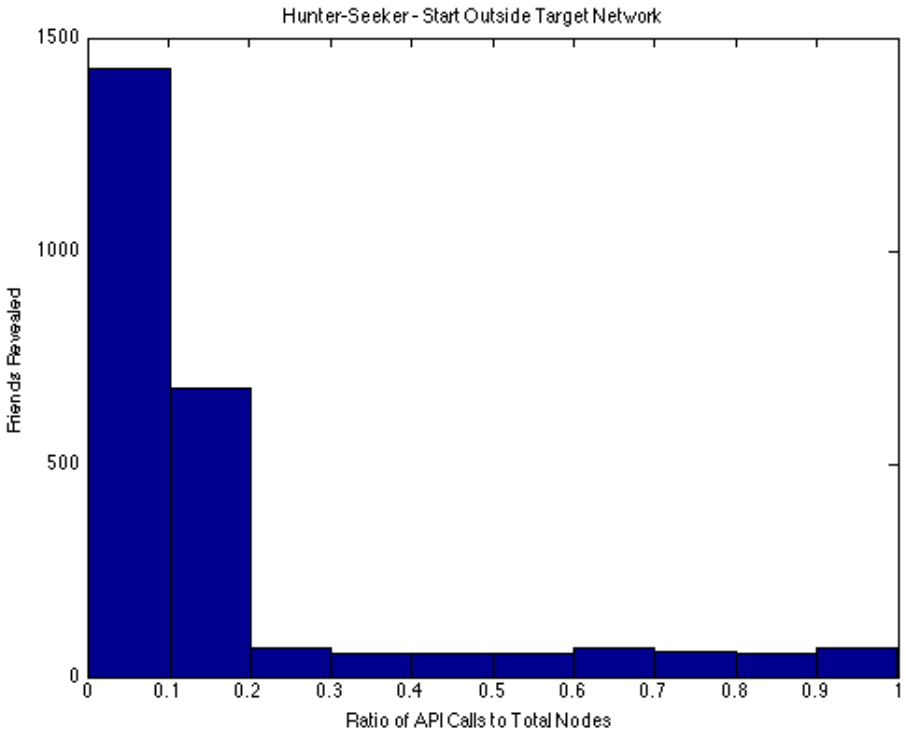

Figure 5.4 Hunter-Seeker Histogram for Start Location Outside of Target Neighborhood

To compare our results to large graphs we scaled the graph by creating a Watts-Strogatz model with parameters $G(10000, 130, .2)$. The resulting graph took about 24 hours to complete the test harness iternations of each algorithm on a modern i7 laptop processor. The graph contained $\frac{nk}{2} = 650000$ edges. Figure 5.5 shows that as n increases the number of nodes ignored by the hunter-seeker algorithms also increases, widening the performance difference between BFS and DFS and the hunter-seeker algorithm.

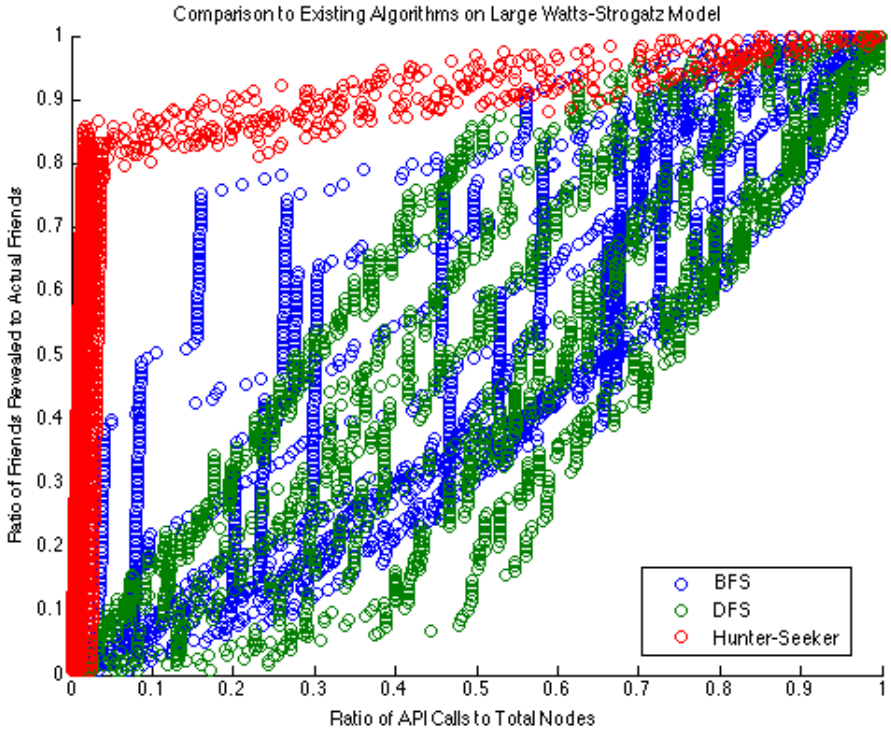

Figure 5.5 Comparison to Existing Algorithms on a Large Watts-Strogatz Model

5.4 Performance on Real World Networks

After speaking with our University Institutional Review Board (IRB) advising board we were advised not to proceed with a large-scale experiment that would break the terms of service agreements defined by Facebook (and most other online social networks). Instead we looked for existing social network research that made public similiar social network datasets. We found a dataset [35] collected as part of study on face-to-face interactions in primary schools. The

dataset contained strong mutual friend relationships and clustering that we intuitively designed our algorithm around.

Figure 5.6 shows that in this real world dataset our algorithm continues to outperform existing search methods of BFS and DFS.

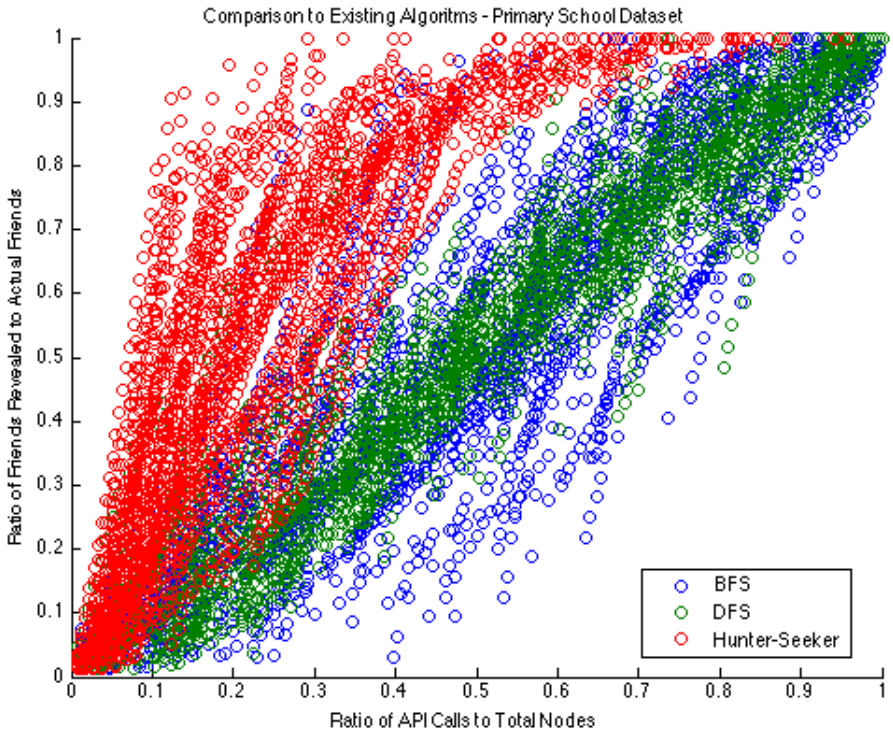

Figure 5.6 Comparison to Existing Algorithms on Primary School Dataset

5.5 Performance on Private Networks

To test our algorithm on private social networks we modified our test harness to randomly privatize a percentage of the total nodes (including the target) before selecting a start location. The test harness then picked two start locations, one that was inside the target network and one that was outside the target network to begin the test data collection process like before. We privatized nodes at 25%, 50%, and 75% on a 1000 node Watts-Strogatz random graph with parameters described previously and ran the test harness. In the best case our algorithm can discover n minus the number of privatized friends (which is on average $k*25\%$, $k*50\%$, and

$k*75\%$ respectively) because there is no way to verify that a private friend and the target are friends.

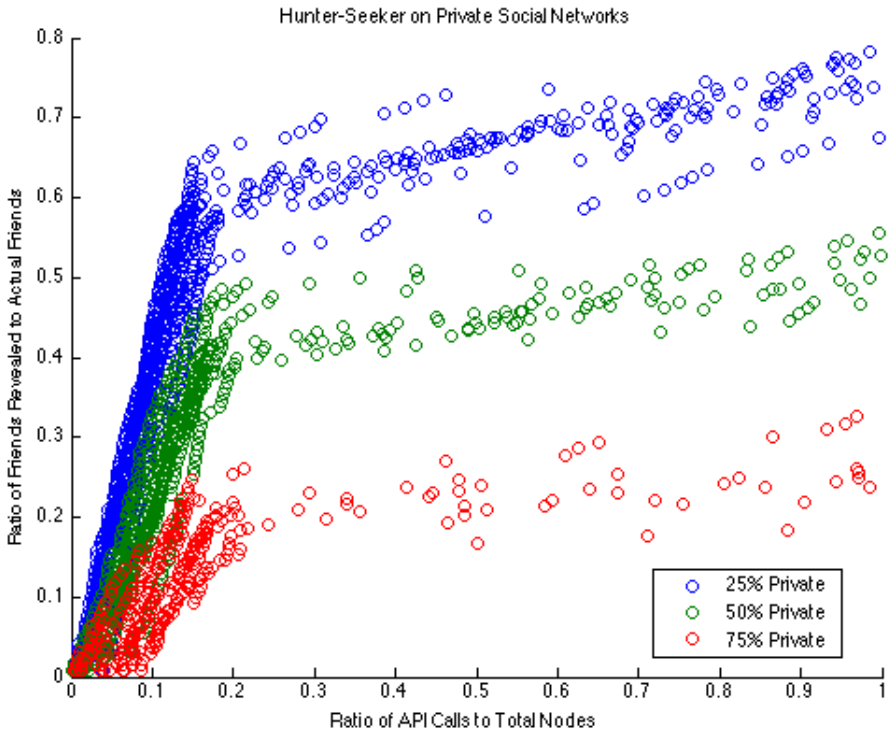

Figure 5.7 Hunter-Seeker on Private Social Networks

CHAPTER 6. CONCLUSION

6.1 Summary

In summary, we examined and compared the needs of the Open Source Intelligence community with what social media has to offer investigators. We observed that a friends list of a given individual is a useful starting point for launching an investigation but found that several technical limitations (privacy and platform restrictions and data availability and longevity) may prevent investigators from accessing friend list information of a target account. We address privacy restrictions for the particular case of friends by creating a private friend discovery algorithm with hunter-seeker behaviors. To address platform restrictions we defined a platform based metric of API calls to measure the algorithms performance, which motivated us to optimizing the algorithm to perform efficiently under practical constraints. While there is little we can do to change the availability of information, our algorithm does address data longevity issues by serving as a mechanism to enable efficient and automatic crawls of the social graph at times defined by the operator.

Our evaluation of the algorithm showed that our algorithm is practical for several reasons. With previous search techniques such as BFS and DFS a large portion of the graph must be crawled to be confident of discovering the majority of the target's private friends. Our hunter-seeker algorithm depends less on the size of the graph making it practical for large social networks. Considering that Facebook has approximately 800 million [34] profiles, a BFS would quickly exceed API rate limiting and request quotas and would most likely not collect the information in a reasonable time (information on online social networks degrades quickly [12] [42]). Furthermore our algorithm performs consistently regardless of its startling location (the same cannot be said of BFS and DFS), meaning an investigator does not need to know

additional information such as a known friend or associate to begin crawling to the target.

While previously investigators would have found it infeasible to search for a target's friends, we have proposed a practical, robust, and cost efficient algorithm to selectively search for private friends. By doing so we have enabled investigators and other researchers to examine new data that was previously unavailable for future study. Through this work we have incorporated other works into a common framework (the Social Media Toolkit), which consists of several utilities ranging from basic interactions with Facebook, to storing and de-anonymizing nodes between multiple graphs, to the implementation of the hunter-seeker algorithm proposed in this paper. By open sourcing the framework we believe we are providing the community with a foundation to conduct further research into OSINT and social media.

6.2 Discussions

Perhaps the largest criticism of our results is that our random graph models do not account for free-scale degree distributions that are commonly found in online social networks. The lack of random models that demonstrate free-scale distributions and the small-world and clustering coefficient properties as well as our inability to gather real-world results of our own (for legal reasons) has left us ill-equipped to address the concern.

We believe the implications of this work will impact the OSINT community and individuals the most. In the FBI Request For Information discussed earlier, interactions with social media were desired to be mostly automatic processes aside from some guidance from the operator. By enabling the discovery of private friends in an automatic process, we have provided a mechanism to enhance the capabilities of future tools. It is unclear as to what a reasonable expectation of privacy is and is not on online social networking sites, but we can safely assume that users that enable privacy protections are expecting at least some level of privacy protection from the social networking platform. By creating a function to lift the privacy restrictions of investigators the operator may be breaching user's reasonable expectations of privacy by using the tool. This was one topic central to a debate in our ethics committee review board that eventually decided not to endorse a large-scale crawl of Facebook.

6.3 Future Work

As future work we aim to expand our framework to study cross-correlations between social networks. We believe that it may be possible to utilize a second reference network to infer information that is not present in a single network through the use of graph de-anonymization techniques. To address the concerns of our algorithms performance on free-scale networks we are seeking to find more public datasets with properties similar to online social friendship networks as well as other random graph models that produce the desired characteristics.

APPENDIX A. ADDITIONAL MATERIAL

This appendix contains outlines of solutions to problems that we encountered or explored throughout this work. We are including it for the sake of posterity in that others might find it to be useful information. Each section is more or less unrelated from another section. The topics discussed in this appendix have been implemented as part of our framework we have dubbed the Social Media Toolkit (SMT) for dealing with social media in the context of OSINT.

A.1 Friend-of-Friend Relationships on Facebook

Both the old Facebook REST API and the newer Facebook Graph API do not support accessing friendship information of an account that is not associated with an authorized account (i.e. friend of friend relationships). This is an intentional handicap of the API to prevent crawling of friendship relationships. The information is available only through the standard web-based interface at Facebook.com. This restriction does not prevent crawling but it does make it significantly more difficult. Any automated solution will have to replicate the actions that a user would take to manually spider friendship relationships.

In our proof of concept crawler we use the Apache Commons HttpClient library released under the Apache Source license to make individual HTTP requests to the Facebook platform that automate manual user actions. To interpret and parse the response we use the open source Jericho HTML Parser released under the Eclipse Public License (EPL). Friendship information is only available once a user has logged into a Facebook account (including information that is declared public in the Facebook user's privacy settings), so a Facebook crawler requires authentication credentials in the form of a username and password that must first be passed as an HTTP POST request to the Facebook login form to authenticate the user before any

crawling can be accomplished. The primary Facebook interface makes extensive use of AJAX requests to asynchronously load information in the background. The use of asynchronous scripts makes it very difficult to determine the proper HTTP requests to simulate on behalf of the user, so after authenticating the account our implementation reverts to using the mobile Facebook interface, which for device compatibility reasons only uses simple HTML. Through manual inspection we have verified that the results returned by the mobile interface and the primary web interface are identical (the same cannot be said for most of the features in the official API). Given the unique Facebook account identifier of a target account, the utility uses the mobile interface to navigate to the friend's page of the target account. Using the Jericho HTML Parser the utility searches for hyperlinks resolving to each friend of the target account and returns a set of unique Facebook account identifiers. A sequence diagram of the crawler process for getting friend identifiers is shown in Figure A.1.

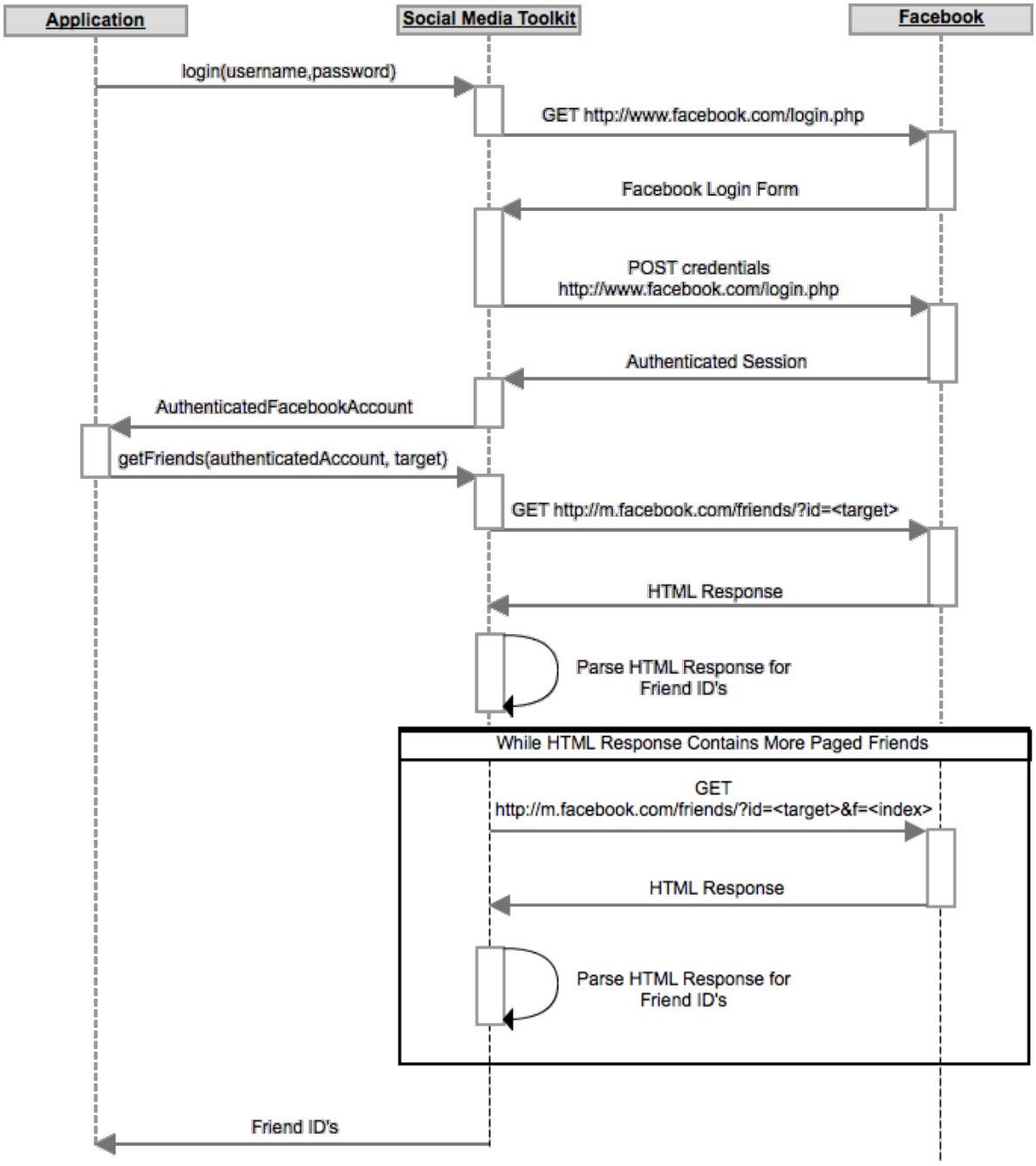

Figure A.1 Facebook Crawler Sequence Diagram

A.2 Storing Social Graphs

A.2.1 Graph Coloring

To store multiple social graphs we must be able to store each social network in the context in which it was collected. In graph theory, edge coloring is a way to allow different edge types within the same graph by assigning a color to an edge. Similarly, node coloring adds a type assignment to the node. For example, the LinkedIn social graph is intended to record undirected "professional" connections, whereas Facebook social graph connections represent undirected "friendship" relationships. The Twitter social graph contains directed "follower" and "retweet" relationships. To distinguish edge types we should assign each edge type a color and each node a respective color based on its graph membership.

A.2.2 Hypergraphs

Storing multiple graph layers can be described formally as a generalization of a hypergraph. We will define a hypergraph as a graph $H = (V_h, E_h)$, where V_h is the set of vertices, and E_h is the set of hyperedges. A hyperedge is a subset of vertices in $P(V_h)$, where $P(V_h)$ is the power set of V_h. For any given set of vertices included in a hyperedge we define a single graph $G_n = (V_n, E_n)$, where V_n is the set of vertices, and E_n is the set of edges between each vertex. A cross-section of the hypergraph can be imagined if we consider each data source (i.e. Facebook, Twitter, LinkedIn, etc.) to be a hyperedge vertices set representing the graph $G_n = (V_n, E_n)$ (a single graph layer) as shown left in Figure A.2. A collapsed view (an overhead view) where each vertically intersecting node (linked by an identity relationship) represents the hypergraph $H = (V_h, E_h)$ shown right in Figure A.2.

A.2.3 Graph Transformations

When comparing two graphs, it is important to compare apples to apples, so we use graph transformations to normalize graphs to a common context. Graph transformations, also known as graph rewriting, are changes applied globally to an entire graph by rewriting an input graph to a corresponding output graph through an automatic machine. A transformation function

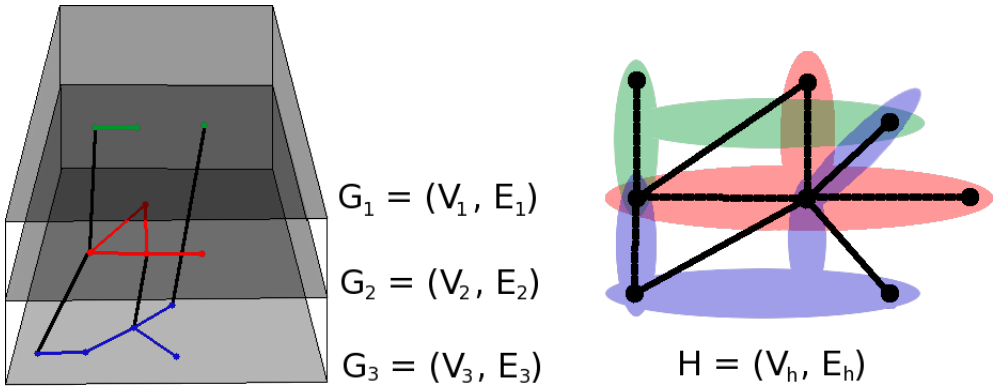

Figure A.2 Hypergraph

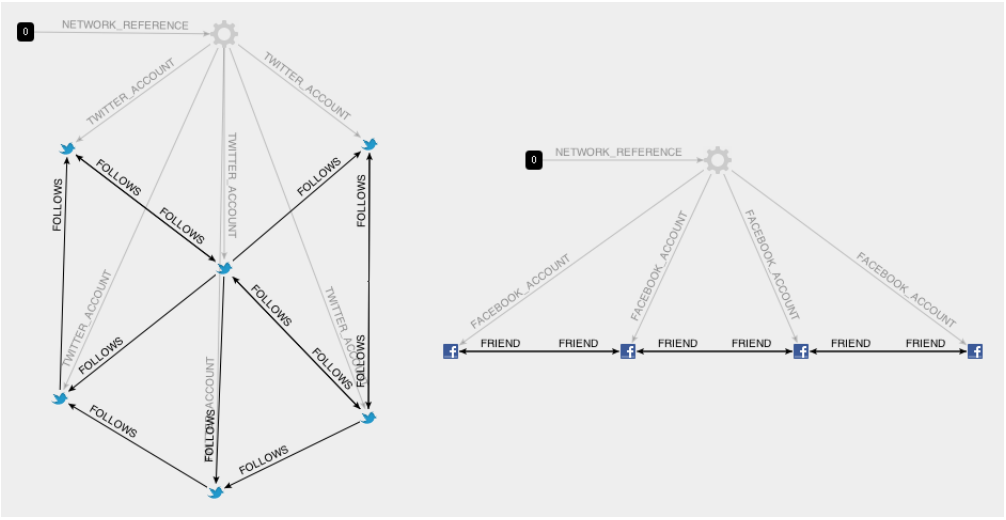

Figure A.3 Sample Normalization of Twitter to Facebook

$f(lgraph) = rgraph$ defines how a graph will be rewritten. One example of a transformation (shown in Figure A.3) could be to normalize a Twitter social graph to an equivilent Facebook social graph using the assumption that two Twitter users that mutually follow each other are equivilent to a Facebook friendship relationship in the Facebook social graph.

A.3 Cross-correlation Identity Mapping

Storing multiple networks has limited uses unless we are able to correlate node identities between network layers. Formally, we define identity mapping as finding a partial one-to-one mapping between nodes in a graph $G_A = (V_A, E_A)$ and a graph $G_B = (V_B, E_B)$. Each mapping

represents an identity relationship between nodes in the two graphs. The task is to determine whether the two graphs are partially isomorphic, with the added difficulty of dealing with background noise present in each graph.

For n graphs $G_1 = (V_1, E_1)$ to $G_n = (V_n, E_n)$ there exists a partial mapping (including the empty set $\emptyset = \{\}$) between every other graph such that a perfect mapping creates an identity relationship between a pair of nodes that are owned by the same user and are both present in the overlap of the two graphs.

We use the Jaccard index of two sets of nodes (each set from a seperate graph) A and B, defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ as a metric to measure the overlap between two sets. Furthermore, we calculate that for n graphs, there exists $\frac{n(n-1)}{2}$ partial mappings between each graph layer in the hypergraph.

A.3.1 Heuristic-Based Identity Mapping

Deep-web search engines such as Spokeo [4], Pipl [3], and Maltego [33] all utilize various features of social networking profiles such as email addresses, usernames, and other details to correlate user identities. In this work we represent our heuristic based identity mapping as a heuristic score that is computed by examing a set of common profile features between a pair of nodes and summing the result of each feature comparison (as illustrated in Figure A.4). The results of each node pair score between the two graphs are then either accepted or rejected by comparing the heuristic score to a threshold value determined to provide an optimal yield of identity mappings with the fewest false-positive identity relationships.

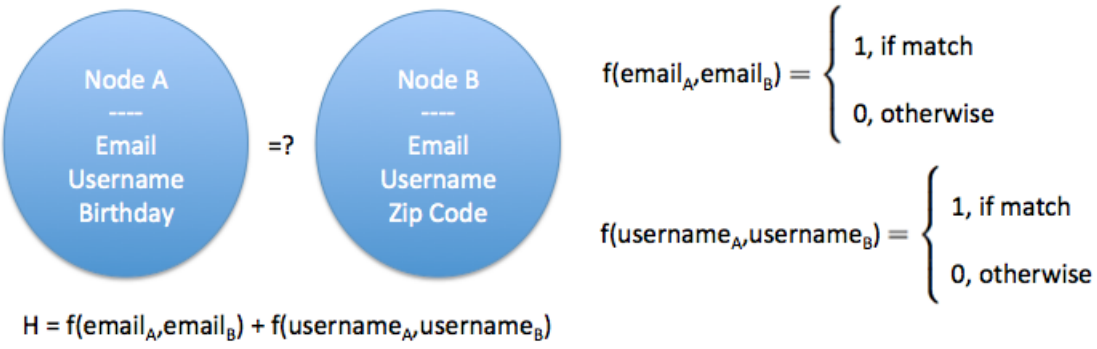

Figure A.4 Heuristic-Based Identity Mapping

A.3.2 Structural-Based Identity Mapping

When heuristic mappings fail due to a lack of identifying information a secondary structural based identity mapping method using algorithms created for node de-anonymization of sanitized graph datasets can be utilized to discover identity relationships. The de-anonymization technique we adopt in this work has proven to be a robust means of node identification in anonymized dataset of social networks [23] and the de-anonymization of the sanitized Netflix movie recommendation dataset [25].

The de-anonymization algorithm makes use of two metrics, cosine similarity and eccentricity. Eccentricity (the measure of how much an item X stands outs from the rest of its parent set) is defined as $eccentricity(X) = \frac{max(X) - max_2(X)}{\sigma(X)}$, where $max(X)$ and $max_2(X)$ are the highest and second highest values in the set respectively, and $\sigma(X)$ is the standard deviation of the set. Eccentricity is 0 if $max(X) = max_2(X)$. The cosine similarity of two sets of vectors X and Y is defined as $cosine(X, Y) = \frac{|X \cap Y|}{\sqrt{|X||Y|}}$.

The algorithm takes two normalized graphs with directed edges and returns an identity mapping between the two sets. Note that we can transform an undirected graph to a directed graph and back again without losing any information. A scoring function takes a node to compare, the two graphs, and a current mapping and computes a metric comparable to the cosine similarity for the vectors of incoming and outgoing node degrees respectively to return an updated node candidate score mapping. To remove bias from nodes with high degrees, the score is divided by the square root of the node's degree. The algorithm iteratively matches

nodes in one graph to nodes in the opposing graph if the opposing graph contains a reverse match to the same node. Matches between nodes with a low eccentricity are rejected from the final mapping because node pairs with a higher eccentricity have a higher confidence interval. The algorithm is run iteratively until it converges on a final mapping by replacing early bad guesses with better guesses in subsequent passes.

The algorithm does require a relatively small initial seed mapping, which can be obtained through the heuristic-based identity mapping methods in the previous section. According to previous work [24], the number of initial seed mappings required is based heavily on the properties and overlap of the two graphs as well as the accuracy of the initial seed mapping. One strategy is to simply continue collecting the number of seeds until enough seeds are found to carry out large-scale de-anonymization. A surplus of accurate seed mappings does not hinder the final result, but a lack of seeds will result in poor identity mapping results.

BIBLIOGRAPHY

[1] Executive order 12333–united states intelligence activities. http://www.archives.gov/federal-register/codification/executive-order/12333.html#2.3, December 1981.

[2] Lococitato facebook visualizer. http://www.lococitato.com/, March 2012.

[3] Pipl. http://pipl.com, March 2012.

[4] Spokeo. http://www.spokeo.com/, March 2012.

[5] 67th District Court Tarrant County Texas. American airlines v. farechase - cause no. 067-194022-02, March 2003.

[6] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, January 2002.

[7] Kevin Bacon. Sixdegrees.org - it's a small world. you can make a difference. http://wwww.sixdegrees.org/about, April 2012.

[8] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, pages 509–512, September 1999.

[9] A. Barrat and M. Weight. On the properties of small-world network models. *The European Physical Journal B*, 13(3):547–560, May 2000.

[10] Hamilton Bean. *No More Secrets: Open Source Information and the Reshaping of U.S. Intelligence*. Praeger, Santa Barbara, CA, 2011.

[11] Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. The socialbot network: The socialbot network: When bots socialize for fame and money. Technical report, University of British Columbia, December 2011.

[12] Salvatore Cantanese, Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Extraction and analysis of facebook friendship relations. Technical report, University of Messina, Italy and University of Oxford, UK, 2010.

[13] A. Clauset, C. Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *ArXiv e-prints*, June 2007.

[14] Erdős and Rényi. On the evolution of random graphs. *Institute of Mathematics, Hungarian Academy of Sciences*, pages 343–346, November 1958.

[15] Erdős and Rényi. On random graphs i. In *Publicationes Mathematicae*, 1959.

[16] United States District Court for the Northern District of California. ebay v. bidder's edge - 100 f.supp.2d 1058 (n.d. cal. 2000), May 2000.

[17] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, January 1986.

[18] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. A walk in facebook: Uniform sampling of users in online social networks. *CoRR*, abs/0906.0060, 2009.

[19] PEW Internet and American Life Project. Adults and social network websites. http://www.pewinternet.org/Reports/2009/Adults-and-Social-Network-Websites.aspx, January 2009.

[20] Markus Jakobsson, Peter Finn, and Nathaniel Johnson. Why and how to perform fraud experiments. *IEEE Security and Privacy*, March and April 2008.

[21] Markus Jakobsson and Jacob Ratkiewicz. Designing ethical phishing experiments: A study of (rot13) ronl query features. *ACM*, pages 513–522, May 2006.

[22] Mary Madden. Privacy management on social media sites. http://pewinternet.org/Reports/2012/Privacy-management-on-social-media.aspx, February 2012.

[23] Arvind Narayanan, Elaine Shi, and Benjamin I. P. Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. *CoRR*, abs/1102.4374, 2011.

[24] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. Technical report, University of Texas, Austin.

[25] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. Technical report, University of Texas, Austin.

[26] NATO. *NATO Open Source Intelligence Reader*, February 2002.

[27] NYTimes. Facebook privacy: A bewildering tangle of options - graphic. http://www.nytimes.com/interactive/2010/05/12/business/facebook-privacy.html?ref=personaltech, December 2011.

[28] Department of Homeland Security. Dhs terrorist use of social networking facebook case study. http://publicintelligence.net/ufouoles-dhs-terrorist-use-of-social-networking-facebook-case-study, March 2012.

[29] Department of Justice. Social media application - fbi request for information. https://www.fbo.gov/index?s=opportunity&mode=form&tab=core&id=c65777356334dab8685984fa74bfd636&_cview=1, March 2012.

[30] United States Department of Justice. What is the usa patriot act. http://www.justice.gov/archive/ll/highlights.htm, December 2011.

[31] Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, December 2011.

[32] Eli Pariser. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Press, New York, 2011.

[33] Paterva. Maltego. http://www.paterva.com/web5, March 2012.

[34] Facebook Press. Statistics — facebook. http://www.facebook.com/press/info.php?statistics, December 2011.

[35] SocioPatterns. rimary school – cumulative networks. http://www.sociopatterns.org/datasets/primary-school-cumulative-networks, April 2012.

[36] Robert Steele. Open source intelligence: What is it? why is it important to the military? *Open Source Intelligence: READER Proceedings*, Volume II 6th International Conference and Exhibit Global Security and Global Comp:329–341, 1997.

[37] TechCrunch. Eric schmidt: Every 2 days we create as much information as we did up to 2003. http://techcrunch.com/2010/08/04/schmidt-data, December 2011.

[38] Jeffrey Travers and Stanley Milgram. An experiment study of the small world problem. *Sociometry*, Volume 32(4):425–443, 1969.

[39] Dallas Division United States District Court, N.D. Texas. Southwest airlines v. farechase - 318 f. supp. 2d 435, March 2004.

[40] Duncan Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[41] Russell Wolff and Jack McDevitt. Using social media to prevent gang violence and engage youth. Technical report, Northeastern University and Massachusetts Executive Office of Public Safety and Security, March 2011.

[42] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *2010 IEEE Symposium on Security and Privacy (SP)*, volume 1081-6011, pages 223–238. IEEE, May 2010.