Unfriend your Boss
Mapping Organizations Social Networks for Red Team Engagement

Quentin Kaiser

contact@quentinkaiser.be

Abstract

In this paper, we explore the security implications of employees publicly exposing their employer through social media. Using Facebook social network as a data source, we go through the steps of building a reliable scrapper to generate an organization social network. We then apply social network analysis algorithms to explore our dataset and identify high value targets, gate keepers, and communities to use that information against the targeted organization during red team engagements. Finally, we propose some recommendations to online social network designers, end users, and organizations.

Contents

I Introduction

The current trend in offensive security is to put a lot of effort researching the latest steps of an attack, that is vectors of intrusion, exploits, keeping access, lateral movement, and hiding. We observe the same trends in reports documenting what is now defined as Advance Persistent Threats that focus on attack vectors.[1]

At the opposite end of the spectrum, research focusing on passive reconnaissance and information gathering is almost stagnating. Even if some toolkits [8][5] are coming up with new and interesting ways of doing OSINT collection such as process automation and integration of social networks as data sources, the current state is that most pentesting teams - or phishing awareness companies - information gathering processes can be summed up in two steps: email addresses acquisition by scrapping data off the Internet, and generating email addresses by combining observed email patterns and acquired data for those who goes the extra mile.

We think that this might lead penetration testers to perform engagement that do not reflect the current level of sophistication of advanced attackers. We therefore decided to fill this gap by showing what applied and extended analysis of acquired data can bring, with the hope that it will inspire those working in offensive security.

II Research Questions

This research is built around three main questions: First, we want to find out how much data is exposed by organizations members through social media by building a tool that can effectively and efficiently retrieve it. Second, we want to discover what information can be derived from such data by means of social network analysis. Third, how information derived by those means can be exploited during an attack to make it stealthier or improve the probability of successful exploitation. When answering that last question, we consider two main applications: finding the most interesting targets within the organization, and how to better target those individuals through social engineering.

III Related Work

This paper does not claim to have designed new and more efficient ways to do social network analysis. We, however, relied on previous research done in the field to inspire us.

First of all, the problems tackled by *The Power of Local Information in Social Networks* by Borgs et al.[2] did not directly helped us in our methodology. However, their exploration of the implications of attributes that preferential attachment networks expose is an eye-opener for anyone that wants to explore privacy implications of online social networks. Their coverage of problem sets related to restricted network visibility helped us in characterizing the networks we analyzed during this research. Furthermore, it greatly summarizes what are the implications of such networks for its designers. This helped us in providing sound recommendations for mitigating the risks we exposed.

[1] This can be explained by the lack of information related to the initial steps performed by the attacker and the need for IOCs.

One of the goal of this research was to see if we could reconstruct an organization internal hierarchy based on network information we retrieved from Facebook. *Inferring the Maximum Likelihood Hierarchy in Social Networks* by Maiya et al.[6] greatly helped us in this challenge. First of all, their extended coverage of prior work done in social networks hierarchy reconstruction provided us with a clear view of the current state of the art. Second, the way they consider hierarchy reconstruction as a generative problem is a real breakthrough compared to much of the previous works that rely on graph theoretic centrality measures to reconstruct hierarchies. In the end, Maiya et al. provided insight in how weight and direction are central to hierarchy reconstruction. A bittersweet realization as our graphs are undirected and unweighted.

In the subject of hierarchy reconstruction, *Finding Hierarchy in Directed Online Social Networks* by Gupte et al.[3] was inspiring but not applicable to our research as we are dealing with undirected graphs.

Eight Friends Are Enough: Social Graph Approximation via Public Listings by Bonneau et al.[1] is the most relevant paper for our research. Indeed, not only it focus on Facebook as a target for social graph approximation but it also rely on scrapping tools to gather data from that website. What is interesting with this paper is that we can look at how Facebook is doing in terms of *social graph privacy* - a term coind by Bonneau - by comparing their results with ours. As we will see, the expansion of Facebook does not necessarily means improvements in term of social graph privacy. We will also see that the limitations related to the *coupon collector's problem* that Bonneau et al. encountered while fetching data from facebook.com does no longer apply when data is queried from Graph Search.

Finally, we would like to acknowledge two code projects: recon-ng[2] and linkedin-neo4j[3]. Recon-ng is a full-featured Web Reconnaissance framework written in Python. Complete with independent modules, database interaction, built in convenience functions, interactive help, and command completion, Recon-ng provides a powerful environment in which open source web-based reconnaissance can be conducted quickly and thoroughly. Linkedin-neo4j is a set of python scripts that loads a linkedin network into neo4j using the linkedin developer API. Tinkering with it convinced us that Neo4J was a valid tool for our purpose.

IV Background

Please note that our metrics fits into a qualitative analysis of 20 companies and organization we managed to analyse. It is important to keep in mind that not all members of an organization possess a Facebook account, yet publicly disclosing their organization's membership. Our objective is to analyse networks to derive information then manually verify it (e.g. a member has a high centrality but its job titles mention "Employee" so we execute a Google Search to find out its real role in the organization). Such process can't be performed automatically nor can it be performed at scale.

[2]https://bitbucket.org/LaNMaSteR53/recon-ng/
[3]https://github.com/rjbriody/linkedin-neo4j

I Facebook Graph Search

Facebook Graph Search is a semantic search engine introduced by Facebook back in March 2013. It is designed so that users can send queries using natural language to search for entities such as pages, people, places, events, check-ins, and status updates.

When this feature got available to Facebook users, some of them started to demonstrate its power by, for example, demonstrating the chilling effect of such a tool when using the right words such as obtaining sets of people that liked LGBT related pages living in countries where LGBT groups are persecuted.

Facebook thwarted all privacy concerns by explaining that entities are indexed based on privacy settings that Facebook users define.[4]

V Methodology

I Data Acquisition

Data acquisition was performed by a scraper written in Python, based on the recon-ng's facebook module we initially wrote. In the next sections, we explain how we built our scraper and the challenges we faced while developing it.

I.1 Building our scraper

The purpose of this scraper is to acquire a list of employees working for a specific company, then obtain relationships linking all those employees together.

Employees retrieval Facebook allows the retrieval of an organization current employees with the following URI: *https://www.facebook.com/search/COMPANY_ID/employees/present*. It is also possible to retrieve former employees of an organization by using the following URI: *https://www.facebook.com/search/COMPANY_ID/e*
The screenshot below shows an example of a successful request.

Relationships retrieval Once our scraper acquired the list of current employees, we request Graph Search for relationships existing between them. To obtain those relationships, we rely on a powerful feature of Graph Search: intersects. Quite undocumented, that feature allows the retrieval of members belonging to an intersection applied to two - or more - search results. Using the following URI: *https://www.facebook.com/search/COMPANY_ID/employees/present/USER_ID/friends/intersect* allows us to obtain the results present in the intersection between a set A, containing all members of an organization, and a set B, containing a specific member's friends.

Note that the way Graph Search works remove the need for optimization (e.g. by using dominant set theory to get higher probability of covering the graph in the least amount of time) as we already know all the nodes that are part of the graph.

[4]The author personally consider that using the privacy settings as an argument for privacy protection is a fallacy. The fact is that if *some* members of a graph under scrutiny are sharing their relationships publicly, each person linked to that overly open person will be included in the graph. Even if they don't want to.

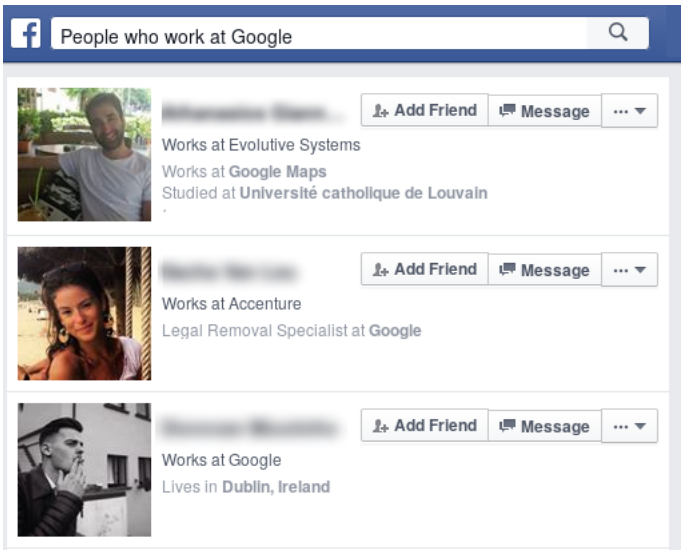

Figure 1: *Employees retrieval request results*

I.2 Challenges

User-Agent Blacklisting It appears that Facebook blocks requests sent with unknown or exotic User Agent values. Overcoming this challenge was as simple as setting our tool's User Agent to mimick one of a browser such as Firefox.

Session Token Facebook previously set the session token by returning it in the login HTTP response Cookie header. Starting in 2015, it changed to setting that cookie on the client side by using JavaScript. We could have digged into the minimized and obfuscated code to retrieve that value but an easier fix was to login on `m.facebook.com` - which still returns it within HTTP headers - to obtain it. This cookie is bound to the *.facebook.com wildcard domain, allowing us to send authenticated requests to Graph Search which is only available on `www.facebook.com`.

Lazy Loading Facebook implements a mechanism that pipeline web pages in order to increase performances. This mechanism, called BigPipe[4], is used by the Graph Search interface to do lazy loading of the search results. Each time the user reach the bottom of the page, a new request is sent to the server to obtain the next batch of search results. We reverse engineered the client code so our scrapper also fetch those data that a traditional crawler would miss.

Parsing Another challenge is the actual parsing of data. Content returned by the Graph Search interface are presented within HTML comments. We still haven't found a nice way of overcoming it and are still relying on regexes[5].

Rate Limiting Rate limiting is not effectively measurable as we do not directly interact with the backend system nor are we sending request to a Facebook API. The "human measurement" of that limit told us that it is around 50 search requests per day. If the targeted company is employing

[5]We know, this is the best way to break software.

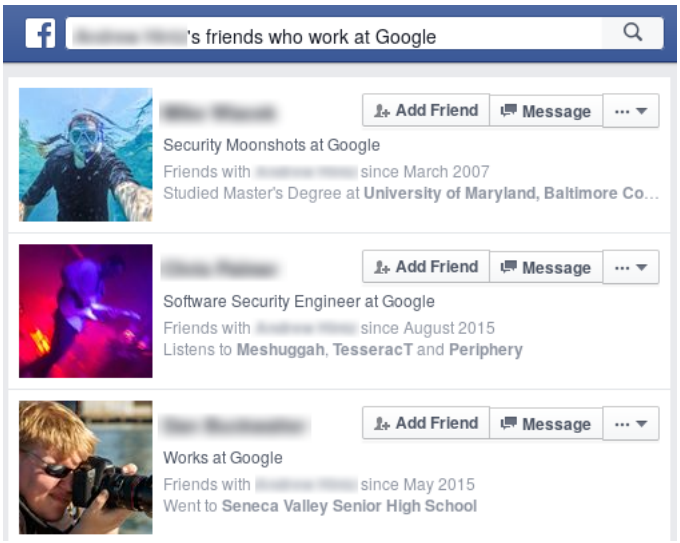

Figure 2: *Friendships retrieval request results*

many people and the alloted time frame for our tool to run is limited, it is up to the user to run scrapping jobs in parallel, each of them connected with a different account.

CAPTCHA When Facebook receives too much requests within a limited time window, it will pop up a modular window with JavaScript, asking the user to click on a button to prove that she is not a bot. We chose to implement an incremental sleep time when Facebook triggers this mechanism and it worked well experimentally. We also discovered that we were not subjected to CAPTCHA protections when requesting employees listing using intersecting requests (e.g. requesting male employees, then female employees).

II Data Storage

We rely on the Neo4J graph database to store acquired information. Initially, direction of edge does not really make sense as the relationship is shared - not like Twitter follower/followed model. However, we could consider it as a directed graph by taking into account the fact that a person is sharing its friendships, compared to the ones that do not. What is the real impact of using such direction is yet to be discovered.

VI Social Network Analysis

To understand the internal organization of a company, we need to first define our network model that serves as the base for our analysis framework.

We consider social networks comprising of employees and their relationships. The social network is represented by a graph G(V,E), where employees are nodes, while relationships are represented as edges. Sets V,E, are the set of all nodes and edges, respectively. Each node possess the following attributes: facebook ID, last name, middle name, first name, and job title. We chose to explore three common graph metrics: community detection, degree analysis, and centrality analysis.

Before we dive into the subject, we would like to point out that our objective is not to come up with exact rules, or even a perfect understanding of the data under scrutiny. Due to the complex nature of human behaviour and the incomplete nature of the acquired data, the relevant analytical methodologies are somewhat flawed and imprecise.

However, this should not forbid us to try if we are intellectually honest from the beginning. Our objective is more about starting a discussion on the security implications of exposing information through social media rather than devise exact analytical rules to do so.

I Centrality Analysis

Degree Centrality Degree centrality is a measure of influence within large and complex networks. In our assessments, the three different profiles that were coming up the most were employees working in the human resource department, union representatives, and so-called "social beasts" who adds everyone as friend but doesn't really influence the network.

Betweenness Centrality Betweenness centrality can be seen as "a measure for quantifying the control of a human on the communication between other humans in a social network". In other terms, betweenness centrality can be used to identify the gate keepers within an organization. During our different assessments, we found out that betweenness centrality can identify high value targets by returning employees holding coordinators profiles such as executive-level employees or directors coordinating communication between different departments or different geographic locations (e.g. working in headquarters but managing everything related to a branch located in a foreign country).

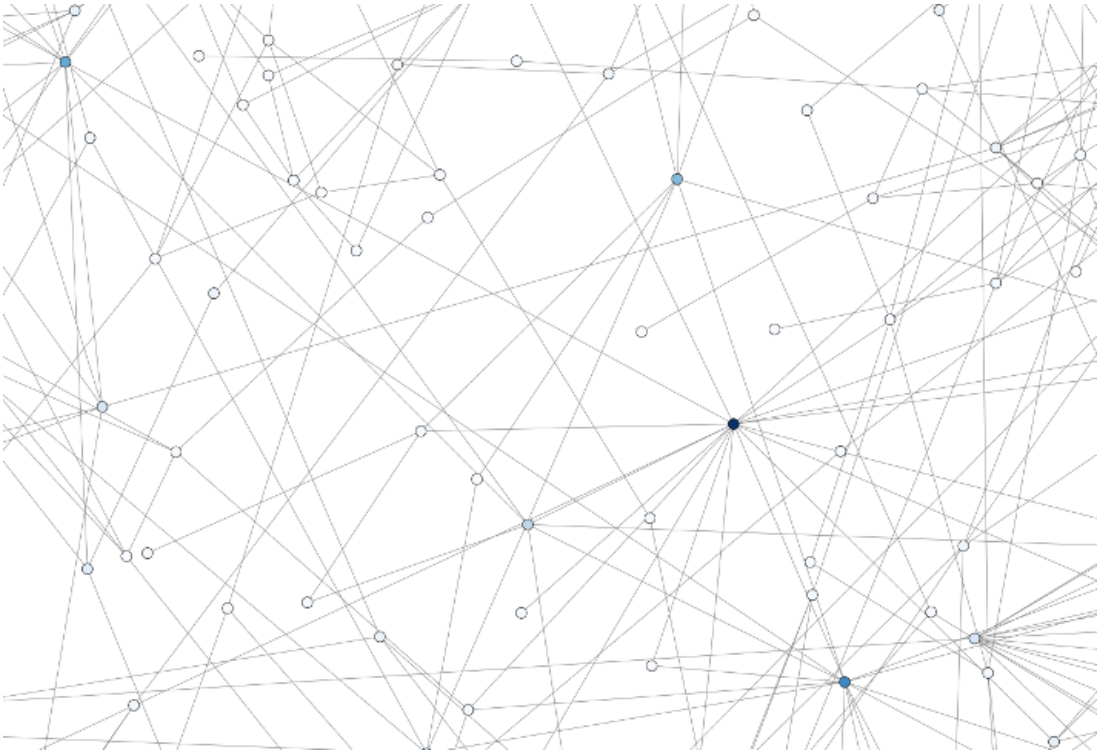

Figure 3: *Betweenness centrality analysis output (darker blue means higher centrality ratio)*

II Community Detection

Humans have a tendency to homophily, mainly caused by geographical and organizational foci. Such a tendency can be visualized when applying community detection algorithms to the acquird social networks. In our research, we relied on Louvain algorithms to detect such communities. We can see a company social network as the main community to which members bond due to organizational foci.

On all of the organizations that we observed, the same trend appeared: communities are clustered around geographical locations. The granularity of those locations highly depends on the organization's structure. Multinational companies had communities showing the different countries on which it operates while organizations operating on a single country had communities that reflects the different towns on which it operates. Even though geolocation information can already be scrapped off an employee's profile, we see community detection as a potential tool to infer location of employees that do not disclose it publicly. A naive approach would be to set the location of such employee to the top location observed within its own community.

When applying community detection to the sub-graphs identified during the initial pass, we observed a second trend: communities reflected internal departments. Facebook allowing its users to mention their job title, it allowed us to observe that trend fairly easily. However, when a large portion of the community did not mention it we had to rely on online searches to enrich our dataset to see if we could confirm our hypothesis.

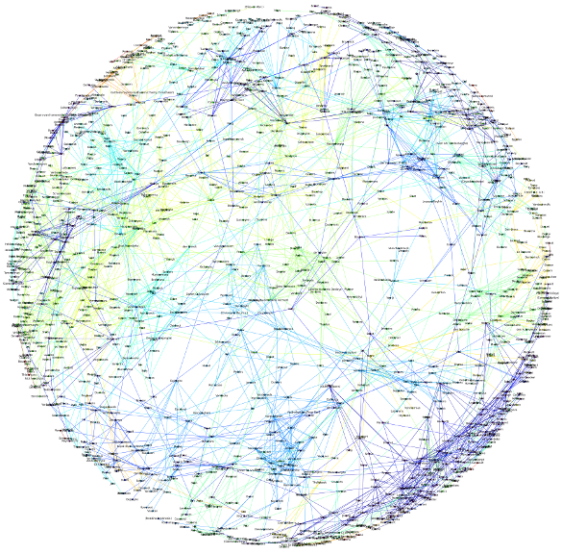

Figure 4: *Community detection output*

An hypothesis that we were not able to prove with our experimental settings due to the lack of insider information is that community detection could be used to detect hidden communities. That is, communities that should not be visible to outsiders. We can think of internal commissions with members coming from different departments and working on projects that the organization would like to keep secret. Another possibility would be members of the targeted organization that shares common interest (sports club, political affiliation).

III Hierarchy Generation

Applying hierarchy generation algorithms did not yield any interesting results. This is due to the fact that our graph edges does not possess a weight value and are undirected. It is highly unlikely that an organization hierarchy could ever be derived from Graph Search results.

VII Discussion

I Leveraging Results for Red Team Engagements

Identifying High Value Targets for Stealthier Attacks Identifying high value targets and their role in the company can help attackers that exactly know what they are after. Let's say that the attacker objective is intellectual property designed in head quarters but sent to a remote branch oversea for manufacturing. By identifying the gatekeeper between those two locations within the graph using betweenness centrality, we most likely identified an employee with access to such information. Without even touching the network.

Instead of targeting the whole company and trying to get domain admin privileges once inside to then search through vast amounts of Sharepoint documents, a stealthier attack would target that specific user and rely on that user privileges only to gain access to the final objective.

Using Community Attributes to Design Phishing Emails To lower the probability of detection when sending phishing emails, attacker can leverage community detection to identify attributes that will help in narrowing down the list of targets and writing believable pretexts. When it comes to narrowing down the list of targets, we can think of phishing campaigns that are aimed at specific departments within the targeted organization. As of writing believable pretext, it is known that writing phishing emails using the recipient's native language is more convincing. An information that can be derived from their geographic location[6].

II Applicability to Other Online Social Netwoks

Twitter We're currently in the process of expanding our tool to Twitter. We leverage the Twitter API to obtain an organization profile first and then obtain a list of employees that mention that page in their Twitter bio. Once a list of employees is built, we generate the graph by looking for following and followers relationships. An important challenge when identifying employees is that mention of a profile within a bio does not always mean the user is a current employee (e.g. "ex-editor @NYT"), or an employee at all (e.g. "I love @Twitter !"). Natural language processing might be of some help during the identification process.

[6]This has been proven more than useful in countries where three different languages are spoken such as Belgium.

III Possible Developments

Profile wise, we only considered a limited set of information that are publicly available (ie. last name, first name, job title) in comparison to the large amount of other informations that organization members can share publicly.

Geolocation Facebook users can share different kind of geolocation information such as their home town, current town, work address, geolocated posts, and checkins. Considering this, we think it is possible to correlate detected communities and geolocation information to identify the different locations in wich an organization is operating (head quarter, contractors offices, ...). An interesting information for red teams executing physical penetration tests.

Time In addition to users potentially sharing their birth date, Graph Search can return two organization related time based information: employmentship duration and friendship duration. This could help an analyst in identifying older employees, new hires, but also potential promotions and mutation within the company if we consider constant monitoring of the graph.

IV Mitigating Exposure

As Bonneau et al. explains it in [1], "It is difficult to safely reveal limited information about a social network". However, we would like to consider the following possibilities.

Facebook We think that implementing limitations in Graph Search such that information exposure limitation is a function of node distance would be a good thing. However, this does not appear to be the direction that Facebook is currently heading to. Another possibility for Facebook could be the implementation of anti-scrapping measures such as better limits on the amount of requests that a single user can send. Our experience is that if a scrapper crawl content by acting like a an authenticated user and does not start to mess with Facebook core business (e.g. generating fake pages with huge amount of likes, bots used for content promotion, spreading malware through messenger), it will stay under the radar.

Organizations Organizations must find the right balance between the threat of exposing itself too much and the PR benefit of exposing itself. After all, having employees mentionning their employer publicly is free advertisement. And even if those organizations choose to implement policies that, for example, require employees not to mention their employer in their profile, the solution is not ultimate. The problem being that the legislation under which the company is operating can consider those policies as an intrusion of employees privacy. Furthermore, those policies have to be strictly enforced to be effective as the inherent characteristics of online social networks showed us that even limited exposure can provide deep insights into an organization.

Employees Employees can act for themselves and remove as much information as possible from their public profiles. This recommendation can be integrated in organization's security awareness programs.

V Mandatory Note on Ethics

Anonymizing Data We took precautionary measures to anonymize all data included in this research paper. This could be seen as an issue given that we can't prove that our analysis actually

yielded the results we report without compromising employees information, such as proving that betweenness centrality returned the list of executives of company X. We do not think that providing that information would have helped in discussing the issue. Verifying our claims with our published tools is left as an exercise to the reader.

Facebook Disclosure We initially thought about getting in contact with Facebook but an guest post on Facebook blog[7] convinced us not to do so. In that article, the author quotes Facebook Security Team on discoverability of friendships:

> This is a case where privacy can get complicated, but we think the way we've chosen to operate is a good balance of the competing priorities involved. We've also chosen to focus more on privacy controls around your content and personal information, since trying to maintain privacy by limiting discoverability is often an illusion. Since Facebook is a network designed for social participation, it's nearly impossible for it to work properly and let people stay completely hidden - there are many ways to discover a profile or friendship beyond friend lists or searches. But even if someone discovers your profile, you have a great degree of control about what they can then access.

VIII Conclusion

To conclude, we have discovered new ways to leverage open source information extracted from online social networks. We proved that social network analysis can be applied to identify the best targets within an organization and provided different scenarios where that knowledge can be applied. We demonstrated that in comparison to Bonneau et al. tools published in 2008, it got easier to scrape data off Facebook since Graph Search release.

Finally, we hope that we have attained our objective by inspiring those who work in offensive security, but also by providing sound recommendations to end users and defenders.

References

[1] Joseph Bonneau, Jonathan Anderson, Ross Anderson, and Frank Stajano. Eight friends are enough: Social graph approximation via public listings. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, SNS '09, pages 13–18, New York, NY, USA, 2009. ACM.

[2] Christian Borgs, Michael Brautbar, Jennifer Chayes, Sanjeev Khanna, and Brendan Lucier. The power of local information in social networks. In PaulW. Goldberg, editor, *Internet and Network Economics*, volume 7695 of *Lecture Notes in Computer Science*, pages 406–419. Springer Berlin Heidelberg, 2012.

[3] Mangesh Gupte, Pravin Shankar, Jing Li, S. Muthukrishnan, and Liviu Iftode. Finding hierarchy in directed online social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 557–566, New York, NY, USA, 2011. ACM.

[4] Changhao Jiang. Bigpipe: Pipelining web pages for high performance, June 2010. Available at https://www.facebook.com/notes/facebook-engineering/bigpipe-pipelining-web-pages-for-high-performance/389414033919.

[5] Darry Lane. Bluto. Available at https://github.com/darryllane/Bluto.

[6] Arun S. Maiya and Tanya Y. Berger-Wolf. Inferring the maximum likelihood hierarchy in social networks. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, CSE '09, pages 245–250, Washington, DC, USA, 2009. IEEE Computer Society.

[7] PHWD. Facebook bug bounties - the unofficial treasure map, June 2016. Available at https://www.facebook.com/notes/phwd/facebook-bug-bounties-the-unofficial-treasure-map/1020506894706001.

[8] Tim Tomes. Recon-ng. Available at https://bitbucket.com/recon-ng.