PhishAri: Automatic Realtime Phishing Detection on Twitter

Anupama Aggarwal

IIIT-D-MTech-CS-IS-MT-002
Nov 30, 2012

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Ponnurangam Kumaraguru (Chair)
Srikanta Bedathur
Fabrcio Benevenuto

Submitted in partial fulfilment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Information Security

Keywords: Social engineering, semantic attack, phishing, learning science, human computer interaction, trust, design and implementation, and real-world studies

Certificate

This is to certify that the thesis titled **"PhishAri: Automatic Realtime Phishing Detection on Twitter"** submitted by **Anupama Aggarwal** for the partial fulfilment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* with specialisation in *Information Security* is a record of the bonafide work carried out by her / him under my / our guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Professor Ponnurangam Kumaraguru
Indraprastha Institute of Information Technology, New Delhi

Abstract

With the advent of online social media, phishers have started using social networks like Twitter, Facebook, and Foursquare to spread phishing scams. Twitter is an immensely popular micro-blogging network where people post short messages of 140 characters called tweets. It has over 100 million active users who post about 200 million tweets everyday. Phishers have started using Twitter as a medium to spread phishing because of this vast information dissemination. Due to constraints of limited text space in social systems like Twitter, phishers have begun to use URL shortener services. In this study, we first provide an overview of phishing attacks for this new scenario. One of our main conclusions was that phishers use URL shorteners not only for reducing space but also to hide their identity. We also observed that social media websites like Facebook, Habbo, Orkut are competing with e-commerce services like PayPal, eBay in terms of traffic and focus of phishers. [1] Further, it is difficult to detect phishing on Twitter unlike emails because of the quick spread of phishing links in the network, short size of the content, and use of URL obfuscation to shorten the URL. We developed a technique, *PhishAri*, [2] which detects phishing on Twitter in realtime. We use Twitter specific features along with URL features to detect whether a tweet posted with a URL is phishing or not. Some of the Twitter specific features we used are tweet content and its characteristics like length, hashtags, and mentions. Other Twitter features used are the characteristics of the Twitter user posting the tweet such as age of the account, number of tweets, and the follower-followee ratio. These TTwitterwitter specific features coupled with URL based features proved to be a strong mechanism to detect phishing tweets. We used machine learning classification techniques and detected phishing tweets with an accuracy of 92.52%. We deployed our system for end-users by providing an easy to use Chrome browser extension. The extension works in realtime and classifies a tweet as phishing or safe. In this research, we showed that we were able to detect phishing tweets at zero hour with high accuracy which is much faster than public blacklists and as well as Twitter's own defense mechanism to detect malicious content. We also performed a quick user evaluation of PhishAri in a laboratory study to evaluate the usability and effectiveness of PhishAri and showed that users like and find it convenient to use PhishAri in real-world. Currently, there are 74 active users of PhishAri chrome extension. To the best of our knowledge, this is the first realtime, comprehensive and usable system to detect phishing on Twitter.

[1] Phi.sh/$oCiaL: the phishing landscape through short URLs, Best Paper Award at CEAS 2011

[2] PhishAri: Automatic Realtime Phishing Detection on Twitter, Best Paper Award at APWG, eCRS 2012

Acknowledgments

I thank all members of PreCog research group at IIIT- Delhi for their valuable feedback and suggestions. I would also like to thank Aditi Gupta for her feedback on initial drafts of this paper. Special thanks to Ashwin Rajadesingan who worked in developing some modules of *PhishAri*. And also to Sidharth Chhabra and Fabricio Benevenuto for being a great collaborator during the early start of my MTech research work. Last and not the least, I really appreciate and thank "PK" for being an awesome advisor and mentor.

Contents

List of Figures

List of Tables

Chapter 1

Research Motivation and Aim

Phishing is an online fraudulent technique used to acquire personal and confidential credentials. Phishing attacks lead to theft of sensitive information such as e-commerce accounts, confidential bank account details and other personally identifiable information of an Internet user. Such attacks have disastrous consequences as they result in identity theft and often result in huge monetary loss [14].

It is estimated that $520 million were lost worldwide from phishing attacks in 2011 alone [24]. Traditionally, phishing attacks target email users, however, with the unprecedented explosion in popularity of Online Social Media (OSM) like Facebook, Twitter, YouTube and Foursquare, adversaries also use these media to spam and phish. In 2010, 43% of all the OSM users were targets of phishing attacks [34]. In 2012, around 20% of all phishing attacks targeted Facebook [25]. Another report in 2012 suggests that social network phishing has jumped 221% to 9,974 attacks during Q1 of 2012 when compared to such phishing instances in the previous quarter [20]. There has been an increase in phishing attacks through social media due to ease and spread of information on social networks. Multiple instances of phishing attacks have been reported on Facebook [3], Twitter [21] and other OSMs [7].

Such a rise in phishing attacks on social media presents a dire need for technological solutions to deter these attacks and protect users from phishing scams. Detecting phishing on social media is a challenge because of (i) large volume of data – social media allow users to easily share their opinions and interests which results into large volumes of data and hence, make it difficult to mine and analyze; (ii) limited space – social media often impose character limitation (such as Twitter's 140 character limit) on the content due to which users use shorthand notations. Such shorthand notation is difficult to parse since the text is usually not well-formed; (iii) fast change – content on social media changes very rapidly making phishing detection difficult; and (iv) Shortened URLs – researchers have observed that more than half of the phishing URLs are shortened to obfuscate the target URL and to hide malignant intentions rather than to gain character space [7]. Short URLs not only hide the target URL but also help in evading blacklists.

Twitter is an online social networking website which allows its users to, among other things, micro-blog their daily activity and talk about their interests by posting short 140 character messages called tweets. Twitter is immensely popular with more than 100 million active users who post about 200 million tweets everyday. [1] Ease of information dissemination on Twitter and a large audience, makes it a popular medium to spread external content like articles, videos, and photographs by embedding URLs in tweets. However, these URLs may link to low quality content like malware, phishing websites or spam websites. Recent statistics show that on an

[1] `http://blog.twitter.com/2011/09/one-hundred-million-voices.html`

average, 8% tweets contain spam and other malicious content [12]. Figure 1.1 shows an example of a malicious phishing tweet.

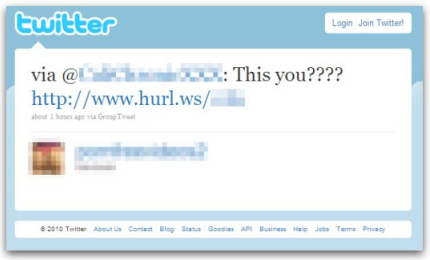

Figure 1.1: An example of a phishing tweet. The URL which appears in the tweet redirects the user to a fake Twitter login page.

In our research, we propose PhishAri [2] – a tool to automatically detect phishing tweets in realtime. PhishAri uses various features such as the properties of the suspicious URL, content of the tweet, attributes of the Twitter user posting the tweet and details about the phishing domains to effectively detect phishing tweets. PhishAri decides whether a tweet is "phishing" or "safe" by employing machine learning techniques using a combination of the aforementioned features. Also, we have built a Chrome browser extension to provide realtime phishing detection to Twitter users. The browser extension protects the user from falling prey to phishing attacks by appending a red indicator to phishing tweets. Further, PhishAri is time efficient, taking an average of only 0.425 (more details later in the paper) seconds to detect phishing tweets with high accuracy of 92.52%. Such low computation times make it ideal for real world use.

[2] 'Ari' in Sanskrit means Enemy, since we were building a tool to curb phishing, our system is christened PhishAri.

Chapter 2

Related Work

Phishing is an online fraudulent technique to acquire personal and confidential credentials of Internet users [14]. Adversaries use phishing for various malicious activities like stealing login credentials of bank accounts, e-commerce accounts and other sensitive information of an Internet user. This section gives an overview of studies which describe how and why phishing attacks are successful and techniques used to detect phishing scams.

2.1 Detection of Phishing Emails and Websites

Traditionally, phishing attacks target email users. Usually, such emails are sent through fake SMTP messages [6] or by impersonating the sending authority [22, 23]. There are powerful email spam filters which effectively filter out spam and phishing emails [6, 10]. Fette et al. used machine learning technique to classify an email as phishing or not by using features such as age of URL, number of dots in URL and HTML content of email while obtaining a high accuracy of 99.5% [10].

Other techniques have also been extensively used to detect phishing websites. Zhang et al. proposed CANTINA, an approach to detect phishing websites by examining the content of the website. CANTINA tries to find out whether the website has been indexed by popular search engines (e.g. Google) or not, which is considered as a measure of a legitimate website [33]. CANTINA+ is another technique proposed by Xiang et al. which extracts features of a website like URL properties, webpage properties and then uses machine learning technique to classify the websites as phishing or legitimate [30]. Blacklist is another popular method in which a record of phishing websites on the Internet is maintained. These blacklists (like APWG blacklist and Google Safebrowsing) are used by many web-based toolbars and web browsers as an early warning mechanism to stop users from visiting the malicious websites. However, blacklisting technique is ineffective as most blacklists catch less than 20% phishing websites at zero-hour [27]. Other methodologies to deter phishing by spreading awareness amongst Internet users have also been developed, which include games [26] and educational technologies [16].

2.2 Phishing and spamming on Online Social Media

With the unprecedented explosion in popularity of Online Social Media (OSM) like Facebook [11], Twitter [18] and Youtube [5], adversaries have started using these media to spread spam and phishing scams. In 2010, 1% of the total Facebook users have been victims of phishing

attacks, which amounts to 5 million Facebook users. [1] Further, Twitter receives a high spam URL clickthrough rate of 0.13%, which is much more than that of email spam [12] as spammers take advantage of the trust network of the social media user. The ease of sharing information on OSM and the larger reach to Internet users makes it a vulnerable target to spread scams [13].

Spam detection studies on Twitter usually involve machine learning classification techniques. These studies highlight important twitter specific features used for spam detection, such as follower-followee ratio, tweet count and age of account. These features can be used to detect spam tweets [8, 29] and spammer [5] with high accuracy. The use of URL shorteners on Twitter to share links makes automatic detection an even more arduous task [2, 7].

Most of the spam spread on Twitter is auto-generated by spammers [8]. In 2011, Zhang and Paxson proposed a method to detect automated activity on Twitter accounts using publicly available timeline of Twitter users. The fact that an *organic* user usually has a uniform distribution of tweets with respect to time, helps in detection of automated accounts. Automated accounts however exhibit a non-uniform behavior because of a fixed timing-distribution. Such non-uniform (referred as anomaly detection in classical security literature) behavior makes the automated activities detectable. The keywords associated with spam tweets have a much higher automation rate. An analysis of source of automated tweets reveals that automated tweets are sent using services like Twitterfeed and Twitter's REST API which provide automation and scheduling, as opposed to organic users who use the usual Twitter's web interface [31].

However, most of this work relates to spam detection on Twitter and very little research work has been done on phishing detection on Twitter or other OSMs, and in particular, on realtime detection.

2.3 Real Time Detection

Phishing is a harmful form of spam. Phishing attacks not only cause the leakage of personal information but also results in huge monetary loss. Hence it is important to build effective realtime phishing detection mechanisms for every OSM to protect its users. There exist browser based toolbars to detect phishing websites [32], but these toolbars require the user to click on suspected and possibly malicious URL. Thomas et al. proposed Monarch, a realtime malware and phishing detection system which crawls URLs submitted to a web service and assesses them in realtime to classify them as spam or legitimate [28]. Monarch relies on features of the landing page which sometime may not be available. However, these solutions are not specific to Twitter. We believe that phishing detection in Twitter hosts a wide range of challenges specific to Twitter itself such as quick spread of information and the limitation of 140 characters in tweets. A dedicated solution proposed exclusively for Twitter by Lee et al. is the WarningBird system which does not focus on detecting phishing but on suspicious URLs in general [19]. It uses correlated redirect chains of URLs on Twitter to detect phishing URLs. However, WarningBird may fail if the spammers use short redirect chain or multiple page-level redirects. Though WarningBird finds suspicious URLs on Twitter in realtime, unlike PhishAri, it does not provide an end-user mechanism for users to use and protect themselves from malicious URLs.

[1] `www.antiphishing.org/reports/apwg_report_Q1_2010.pdf`

2.4 Real Time Phishing Detection on Twitter

After reviewing the above techniques, it was evident that there was very little work done to detect phishing on Twitter in realtime. To fill this gap, we designed and developed PhishAri; it leverages the power of blacklisting as well as other Twitter based, URL based and WHOIS based features. Apart from a robust API which performs realtime phishing detection, we also developed a browser-based extension to protect users from phishing attacks.

Chapter 3

Research Contribution

In the first part of our research, we landscape the spread of phishing URLs on social media through the use of URL shorteners. We do an intense analytical study to find the following -

1. Phishers use URL shorteners not only for reducing space but also to hide their identity.
2. Online social media brands account for more than 70% clicks amongst the top 10 brands. Online social media brands like Facebook, Habbo, and Twitter are targeted by phishers more than traditional brands like eBay and HSBC.
3. Phishing URLs which are referred from Twitter have an edge over the others with respect to attracting victims.
4. Around 30% of the Twitter users spreading phishing URLs turn from *organic* (manual) to *inorganic* (automatic) users in a short span of time which is an indicator of spread of phishing (spam in general) campaigns.

After analyzing the spread of phishing on online social media, we propose PhishAri - an automatic and realtime phishing detection method for Twitter. The major contribution of this work are -

1. *Automatic realtime phishing detection mechanism for Twitter:* There have been studies on phishing detection in emails and spam detection on Twitter, but, to the best of our knowledge, this is the first comprehensive focused study on realtime detection (with a focus on building usable system) of phishing on Twitter.
2. *More efficient than plain blacklisting method:* Our technique proves to be better than plain blacklist lookup which is the most common technique used for phishing detection.
3. *Better than Twitter's own phishing detection mechanism:* Twitter has its own phishing and malware detection mechanism but it is often thwarted by the use of URL shorteners and multiple redirections. PhishAri is able to detect more phishing tweets than Twitter's own detection mechanism.
4. *Real-world implementation of the system:* To the best of our knowledge, PhishAri browser extension and API are the first ever deployed systems for phishing detection which can be (are being) used by real world Twitter users. PhishAri browser extension is freely available on Chrome Web Store for download. [1]

[1] `https://chrome.google.com/webstore/detail/pheokmlohglcpigbnbenbimcombeoolm`

Chapter 4

Solution Approach

In this study, we follow a two step approach.

Analysis of phishing URLs spread through short URLs: URL shorteners emerged as a bridge between needs of users to make their links shorter and terser, leading to obfuscating long URLs. URL shortening services were launched as early as 2001 and now there are hoards of them with bit.ly leading the pack. [1] Some popular URL shorteners like bit.ly, goo.gl, is.gd, ow.ly have shortened in total billions of links till now. Bit.ly alone accounts for about half of all URL shorteners on Twitter and beats the second best by a large difference [2]. The extensive usage of URL shorteners has aided spammers to spread spam. In July 2009, 6.2% of all detected spam messages on the Internet contained short URLs.[2] Many URL shorteners also offer its users, statistics for the click-stream on their shortened links and may pose a privacy risk. Sensing danger for their survival with increased roar over these risks, involved with URL shorteners, many such services have added a phish filter in the redirection process. As a first step to understand how phishing URLs propagate, in this research we track the evolution of phishing through the landscape of URL shorteners on online social media. Due to constraints of limited text space in social systems like Twitter, phishers have begun to use URL shortener services. In this study, we provide an overview of phishing attacks for this new scenario.

Phishing Detection on Twitter: Using few features from our analysis of short URLs used for phishing and delving deeper into understanding how phishing URLs percolate on online social media, specially Twitter, we build a phishing detection mechanism for Twitter which we name 'PhishAri'. We use various features such as the properties of the suspicious URL, content of the tweet, attributes of the Twitter user posting the tweet and details about the phishing domains to effectively detect phishing tweets. PhishAri decides whether a tweet is phishing or safe by employing machine learning techniques using a combination of the aforementioned features.

[1] `http://www.appappeal.com/the-most-popular-app-per-country/url-shortener`
[2] `http://www.certmag.com/read.php?in=3863`

Chapter 5

Phishing Landscape through Short URLs

Our objective in this research is to track the evolution of phishing through the landscape of URL shorteners on online social media and attempt to answer some of the unexplored questions like - Which are the brands (traditional vs. online social media) targeted by phishers? Where (on the web) do these shortened phishing URLs originate from and what is the spread (across the globe) of the victims clicking on these shortened phishing URLs?

Answering these questions is important for both technology developers and policy decision makers. However, to the best of our knowledge, there is no (very little) effort done in providing such overview of phishing related to online social media. In order to approach these questions, we used the blacklisted phishing URLs from PhishTank[1], linked these URLs to bit.ly and analyzed phishing shortened URLs originating from Twitter.

For our analysis, we collected data from PhishTank for the period Jan 1, 2010 to Dec 31, 2010. PhishTank *is a collaborative clearinghouse for data and information about phishing on the Internet*. Anyone can contribute to PhishTank by voting / submitting URLs. They use an adaptive cut-off for number of votes required for a submitted URL to be declared verified (referred as yes in our dataset). On average, number of unknown URLs decreased with time. The dataset had total 1,96,442 URLs out of which 1,18,119 were voted 'yes' by users as phishing URLs. Among the 1,18,119 URLs, there were 63,175 unique primary domain names.

Now we discuss the impact of using URL shorteners by analyzing the space gain, change in landscape of the target brands attacked by phishers, referral analysis for the bit.ly URLs pointing to phishing URLs and geographical spread of the victims clicking on the phishing URLs.

5.1 Space Gain

A bit.ly URL has two parts, the domain name, *bit.ly* and a hash $HaSh10$ (a case sensitive alphanumeric code). Bit.ly hashes can be of variable sizes; with time, the length of hashes have increased from 4 to 6 (to accommodate increasing demand). Our dataset (6,474 URLs) had 24 hashes of length 4,462 of length 5, and the rest 5,988 had length 6. Bit.ly started using length of 6 as default from mid of 2010. Majority of bit.ly hashes in our dataset are of length 6 which may imply that most of these shortened URLs were created in later half of 2010.

[1] `http://www.phishtank.com/`

To ascertain if bit.ly has really helped phishers, we calculate the *space gain* for each URL. By space gain, we mean the fraction of space saved by using bit.ly URL instead of the actual long URL. We find average space gain to be 39%; Figure 5.1 shows the cumulative *space gain* for PhishTank URLs found in bit.ly. For 50% of the phishing URLs, we observe a 37% or less space gain; for generic URLs, researchers have shown 91% space gain [2]. This implies that URLs shortened by phishers are not as long as generic URLs which favors the hypothesis that URL shorteners are not only used to make communication terse but to hide the real link behind bland hash so as to escape any scrutiny which is based on only URL text. Also Antoniades *et al.* found URLs with space gain ≤ 0 to be negligible [2], whereas we find 379 (5.9%) URLs which are shorter or equal to the shortened URL. We believe, phishers are taking up this loss in trade for new identity and an extra hop. Adding multiple hops, i.e., the use of URL redirection can break apart many spam filters and gain trust of users. URL redirection is a common technique used by phishers and it has been studied in past by analyzing vulnerabilities of open redirects.

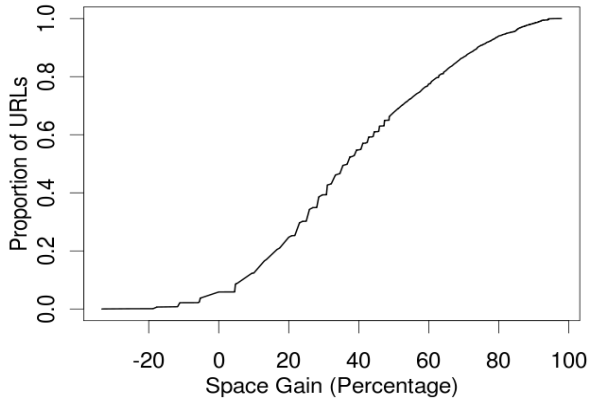

Figure 5.1: Shows the cumulative space gain in percentage against the proportion of URLs. We find that about 37% or less of space gain for half the URLs in our dataset.

5.2 Target Brands

In this section, we present the evolution of phishing targets from e-commerce services / financial institutions to online social media brands like Facebook, Orkut. To investigate the brands (or companies) targeted by phishers using URL shortener, we followed two steps: first, we used the brands listing created by PhishTank; the administrators of PhishTank maintain a list of popular 104 brands.[2] When tagging or annotating a URL as phishing, PhishTank suggests the contributor (one who adds or annotates the phishing link in PhishTank) to select the brand from the previously generated list. We compared the URLs that we had with the URLs from the 104 PhishTank brands, we found 50 popular brands amongst 2,349 URLs. We could not confirm the target for most of them as about 95% of these pages were down at the time of analysis.

Figure 5.2 shows the frequencies for top 10 brands with boxplot for number of clicks they received during the period of our analysis. The bottom and top of the box are 25^{th} and 75^{th} percentiles, and the band near the middle of the box is 50^{th} percentile. Four of the top 10 are online social media brands (Facebook, Orkut, Habbo and Zynga); five are e-commerce services (Bradesco, eBay, HSBC, IRS and PayPal) and one an email service provider (Live). Online social media brands account for more than 70% clicks amongst the top 10 brands. Though PayPal accounts

[2] `http://www.phishtank.com/target_search.php`

for a third of "branded" URLs but median number of clicks (50^{th} percentile) is 1, whereas Bradesco[3] has about 100 URLs but median number of clicks around 200. Habbo[4] is a recent entry to online social media and is a prime target amongst all online social media brands, in-fact, it has almost the same number of shortened phishing URLs as PayPal. Fast growth coupled with large user base supported by open architecture of online social media is making it more lucrative and easy to attack for phishers day-by-day. We see that there are many brands where the number of URLs are low, but the median clicks are high and some where the number of URLs are high and the median clicks are low which negates the silent assumption that large number of phishing URLs trap large number of victims.

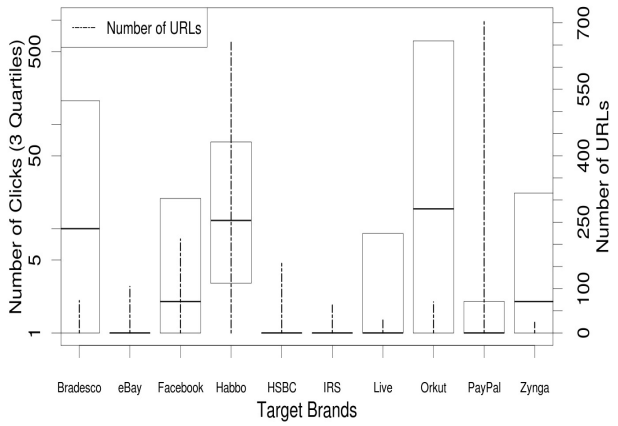

Figure 5.2: Shows the frequencies for top 10 brands with number of clicks they received during the period of our analysis. Four of the top 10 are online social media brands.

To observe the changing focus of phishers, we draw the temporal distribution of clicks for branded URLs. Figure 5.3 shows weekly average of clicks for top 5 brands. We look at only the top 5 brands because they constitute 78.37% clicks out of all branded URLs. We see that Habbo's average number of clicks increased heavily after Sept 2010; there seems to be a large difference between average clicks for Habbo and the next hit brand PayPal after Sept 2010. We also observe that on average PayPal's clicks are increasing with time and follows a cyclical pattern whereas Facebook achieved peak during July and August. HSBC received some traffic during February and eBay got spikes of clicks sparsely through the year. This indicates about the change in focus of phishers, from financial institutions / e-commerce websites to online social media. It brings to fore the need to shift focus of phish detectors to online social media.

5.3 Referral Analysis

Until now, we have been focusing on victims but now we examine the breeding zones, the places which are actively used to spread these phishing links. We query bit.ly API to fetch referral for clicks on every URL and get cumulative click counts for URLs which lead to the bit.ly URL. From the bit.ly statistics, we found that websites other than Twitter, which impose no limits on text length are also referrals for short phishing URLs. This affirms the belief that short URLs are used for the purpose of hiding suspicious URLs behind cover rather than shortening them. Antoniades et al. found that Facebook accounted for only 1.72% referrals (in general short URLs) [2]. However, here we found that Facebook accounted for 11.13% referrals of phishing

[3] `http://www.bradesco.com.br`
[4] `http://www.habbo.com.br`

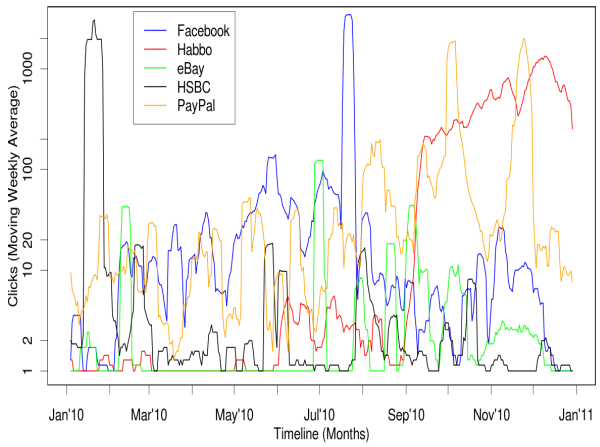

Figure 5.3: Shows the weekly average of clicks for top 5 brands. We see that Habbo has become primary target of phishers in last quarter (October, November, December).

short URLs and Orkut accounted for 31.48%. Twitter has risen from 12% in general URLs to sizeable 23% of all referrals in phishing URLs and has become a phisher's paradise. This significant increase clearly shows that online social media is the new target of phishers. Emails play a lesser role (18%) here which makes URL shorteners more suspicious for phishing in online social media.

To characterize more about the phishers, we queried the Twitter API for the referrals we got from the bit.ly. Orkut and Facebook don't allow crawling user profiles, so, we could not get more information from them. Though Twitter accounts for 23% of referrals, a significant portion of referral came from protected / private accounts.[5] We found 990 Twitter users who were public when they tweeted or re-tweeted one or many "phishing" status (tweet). At the time of analysis, only 864 among them were present, rest were suspended or deleted.[6] Table 5.1 reports some of the characteristics of these users who tweeted a phishing URL. Large number of followers, followees and statuses are indicators of automated accounts [11].

To study whether Twitter acts as a strong push for a phishing URL's publicity or not, we look at the clickthrough and geographical statistics from bit.ly. Table 5.2 compares summary statistics for number of clicks, geographical spread (number of countries), temporal spread (lifetime) and web popularity (number of referral) among URLs which were referred and not referred by Twitter. It clearly shows that phishing URLs which were referred from Twitter had an edge over the others. In-fact mean lifetime was seven times higher and mean number of clicks quadrupled. Therefore, building technologies around curbing phishing in Twitter (online social media) can largely reduce the effect of phishing on the Internet.

5.4 Behavior Analysis

Next, we classified user profiles into *organic* and *inorgranic*. An *organic* account is one of a legitimate Twitter user who posts her tweets manually. Hence, an organic user usually has a uniform distribution of tweets with respect to time. Whereas, an *inorganic* account would exhibit detectable non-uniformity in timing pattern [31]. In order to detect behavior, we collected recent status updates (max. 200) and used Pearson's χ^2 test to determine if the timing distribution

[5] `http://support.twitter.com/entries/14016-about-public-and-protected-accounts`
[6] `http://support.twitter.com/articles/15790-my-account-is-suspended`

Table 5.1: Some descriptive statistics of users who tweeted a phishing URL.

Statistic	**Friends**	**Followers**	**Status Messages**
Minimum	0	0	0
Median	597	302.5	2,138
Average	5,724.3	1,189	6,369.1
Maximum	12,49,000	1,27,291	3,40,113

Table 5.2: Summary statistics for URLs with / without Twitter referral. All values are mean with median in bracket. Shows that phishing URLs which were referred from Twitter has an edge over the others.

Twitter Referral	Clicks	Countries	Lifetime	Referrals
No	70.2(0.0)	2.7(0.0)	12.8(0.0)	2.5(1.0)
Yes	95.1(14.0)	7.2(4.0)	84.0(8.0)	7.2(4.0)

is uniform (human). We were able to obtain timeline (status updates) for 820 users as others were protected. Zhang and Paxson used this method and concluded that 16% of profiles were found to be using *inorganic* means [31]. We used same parameters i.e. $p < 0.001$ as threshold and if either of minutes or seconds distribution had p-value less than 0.001 then it was tagged *inorganic*. We found that 89% of the profiles were using their accounts inorganically. Inorganic accounts exhibit a robotic pattern of status updates as shown in Figure 5.4 (Present). A point in the plot is the time-stamp of tweets by the user. The user updates were concentrated around mid of the hour. With the advent of tools like tweetdeck, powertwitter etc., one can schedule tweets and thus create a mirage of a trend. We observed here two things, firstly, the probability of using inorganic means for wrong motive is significantly higher than for general purpose and secondly, phishers still employ the good old personalized message which is why a 11% of users update their timeline manually (*organically*).

To find if the users' behavior has changed over the time we went back 2000 status messages in their timeline (we were able to download tweets in the past only for 516 users), retrieved 200 tweets (from 2001 until 2200) and again did the same behavioral analysis. We found only 213 (41.3%) Twitter users were behaving inorganically. Around 153 (29.7%) users turned *inorganic* to *organic* in last 2000 tweets which is an indicator of spread of phishing (spam in general) campaigns. As an example of change in behavior, Figure 5.4 shows the message-posting pattern of a user in present and past (200 tweets back).

5.5 Network Analysis

In order to understand the network properties, we queried Twitter API for "Friends and Followers resources." We found that the friend-follower network is sparse with 254 nodes having 356 links among them as shown in Figure 5.5. Even though this is a small sample, we found $1/3^{rd}$ of nodes were connected. The network density is 0.01 and reciprocity is 56%, which is significantly higher than the 22% that has been observed in general population of Twitter [17]. Spammers increase their influence by following various strategies to increase the number of followers. One of the strategy is to follow others in hope of getting followees (return favor) and later discontinue

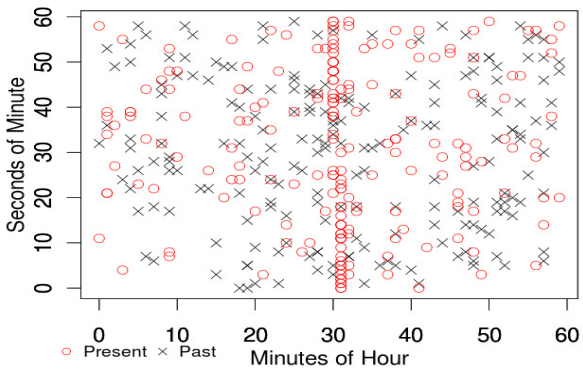

Figure 5.4: Temporal pattern for status updates of a user. A point on the plot is time-stamp for a tweet. Past (black X) is the posting pattern for user in the past (2000 tweets back) and Present (red circle) is the posting pattern in the present for 200 tweets. Shows change from organic (manual) to inorganic (automatic).

being follower.[7] A high reciprocity and sparseness points towards to a similar strategy being followed amongst phishers. Also $1/5^{th}$ of links are Simmellian ties [15] and there are 26 cliques of sizes 3, 4 and 5 which points to presence of strong ties amongst the nodes. Simmellian ties are measure of cooperation amongst nodes and longevity of the network. This implies that phishers communicate and network effectively to achieve higher gains (trap victims).

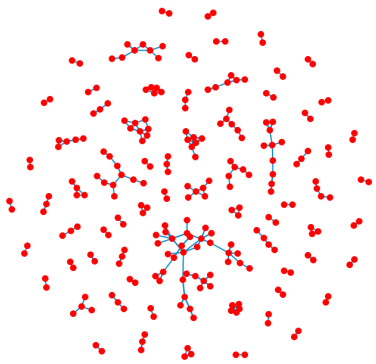

Figure 5.5: Network of the people tweeting the phishing URLs; shows sparse network with high reciprocity. We used ORA from CASOS Center at CMU to create this figure.

[7] `http://www.barracudalabs.com/downloads/2010EndyearSecurityReportFINAL.pdf`

Chapter 6

Data Collection and Labeled Dataset

Now, for phishing detection on Twitter, we start with data collection for our analysis to build a true positive dataset of phishing tweets. Data collection involves two steps as shown in Figure 6.1, (i) collecting data from Twitter, (ii) labeling the tweets as phishing or legitimate. [1]

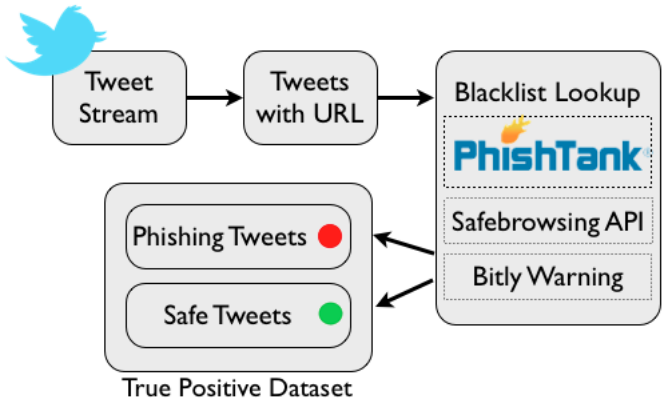

Figure 6.1: Architecture for data collection. We collected tweets with URLs from Twitter stream and compared the URLs against phishing blacklists to build a true positive dataset.

6.1 Crawling Twitter

For our study, we required only tweets containing URLs. We used the Twitter Streaming API [2] and the "Filter" function provided by the API to collect such tweets. As the Twitter Streaming API is rate limited, we can collect only a limited number of tweets per hour. In total, we collected 309,321 such tweets from 1 February 2012 to 19 April 2012.

6.2 Labeling Tweets as Phishing or Legitimate

To initially label tweets as phishing or legitimate in order to create an annotated dataset, we used two blacklists, PhishTank and Google Safebrowsing. For URL in every tweet, we queried both

[1] We use 'legitimate' and 'safe' interchangeably.

[2] `https://dev.twitter.com/docs/streaming-apis/streams/public`

PhishTank and Google Safebrowsing APIs. PhishTank [3] is a public crowdsourced database of phishing URLs. The suspicious URLs are submitted in the PhishTank database by contributors and marked as phishing or legitimate by volunteers. The PhishTank API accepts an HTTP POST request along with the query URL, and returns a JSON object in response which tells whether the query URL is phishing or not. Google Safebrowsing [4] is a database of malware and phishing links maintained by Google Inc. The Google Safebrowsing API uses an HTTP POST request to query the URL and matches the hash of the URL in its database of phishing and malware URL hashes. The response from the API is a JSON string describing whether the URL is "phishing," "malware" or "safe." In case the URL in a tweet is phishing according to PhishTank or Google Safebrowsing API, we mark the Tweet as "phishing." However, the inherent problem of blacklists is that they are slow to capture malicious URLs [27]. We observed that the phishing URLs did not get caught by blacklists on the same day they were posted on Twitter. Even after one day very small number of URLs were detected as phishing. Therefore, we waited for 3 days and checked all the URLs in the tweets we had collected 3 days earlier. We then repeated the same process for entire period of the data collection to build the true positive dataset of phishing tweets.

Apart from using PhishTank and Google Safebrowsing API, we also mark tweets as "phishing" which are declared 'phishing' by Twitter itself. Twitter opens a warning page when one clicks a malicious URL. Also, many URLs posted on Twitter are shortened using Bitly URL shortening service and have the domain name "http://bit.ly/." Bitly uses blacklisting services from various resources and also throws a warning page if it detects a phishing URL. We mark any such URL as "phishing." After applying the above technique to 3,09,321 tweets, we obtained 1,589 phishing tweets (with 903 unique URLs) in our labeled dataset.

[3] `http://www.phishtank.com/`

[4] `https://developers.google.com/safe-browsing/`

Chapter 7

Feature Selection for Phishing Detection

Phishing detection on emails has been studied in the past which shows that phishing websites can be detected using a thorough analysis of the URL and the website content. However, it has been observed that phishers constantly keep changing the techniques they use for phishing, making detection more difficult. Therefore, in this study we combine a variety of features to provide a more robust, water-tight and efficient detection methodology. This section explains the various features we identify for phishing detection on Twitter. Table 7.1 gives a list of all features which we used for our analysis.

7.1 URL based Features

URL features are based on the analysis of the URL of the suspicious website. The length of the URL, number of dots and subdomains, and the length of the domain are some of the most significant features that aid in phishing detection. In phishing websites, the length of the URL tends to be much longer than legitimate websites. However, the phishing domains (without TLD portion) are shorter than the regular domains. Also, phishing URLs often contain more number of dots and subdomains than legitimate URLs [10]. We also observe that many phishing URLs (using "robots.txt") automatically redirect bots (not browsers) to a legitimate domain instead of redirecting to the original phishing domain. This is one of the most effective techniques used by phishers to evade bot-based automatic detection systems. We add such behaviour also as a feature in phishing detection. We also use number of redirections as one of the features since malicious URLs often have multiple URL redirects to escape detection by blacklists.

7.2 WHOIS based Features

WHOIS is a query and response protocol which provides information such as ownership details, dates of domain creation / updation of the queried URL. We can identify tweets containing phishing URLs by identifying WHOIS based features that are common to phishing links. Most phishing campaigns register domains of websites from the same registrar, hence tracking the registrar may aid in detecting phishing. Further, most phishing urls are bought for a short period of one year as offenders need to keep constantly changing the url domain names to evade blacklists. Also, the phishing domains are usually created / updated just before they are tweeted.

Table 7.1: Features used in PhishAri. Classified into URL based, Tweet based, Network based, and WHOIS based.

URL Based (F1)	Length of URL	Length of expanded URL in number of characters
	Number of dots	Number of dots (.) used
	Number of subdomains	Number of subdomains (marked by /) in the expanded URL
	Number of Redirections	Number of hops between the posted URL and the Landing page
	Levenshtein distance between redirected hops	Avg Levenshtein distance between length of redirected URLs between original & final URL
	Presence of conditional redirects	Whether the URL is redirected to different landing page for browser or an automated program
WHOIs Based (F2)	Registering domain name	Name of the domain provider
	Ownership period	Age of the domain
	Time taken to create Twitter account	How much time lapsed between creation of domain and the Twitter account
Tweet Based (F3)	Number of #tags	Number of topics mentioned in tweet
	Number of @tags	Number of Twitter users mentioned in tweet
	Presence of trending #tags	Number of topics mentioned which were trending at that time
	Number of RTs	Number of times the tweet was reposted
	Length of Tweet	Length of tweet in number of characters
	Position of #tags	Number of characters of tweets after which the #tag appears
Network Based (F4)	Number of Followers	Number of Twitter users who follow this Twitter user
	Number of Followees	Number of Twitter users who are being followed by this Twitter user
	Ratio of Followers-Followees	Number of Followers / Number of Followees
	Part of Lists	Whether the Twitter user is part of a public list
	Age of account	How old the Twitter account is
	Presence of description	Whether the Twitter account has a profile description
	Number of Tweets	Number of tweets posted by the Twitter user

Thus, phishing links generally have low time interval between the domain creation / updation date and the tweet creation date. Therefore, we use WHOIS based features such as registrar's name, ownership period, time interval between domain creation / updation and tweet creation date to further enhance our phishing detection methodology.

7.3 Tweet based Features

Malicious tweets are often tailored to gain more visibility in Twittersphere. Phishers achieve high visibility by carefully using tags in their tweets and by timing their tweets at appropriate intervals of time. Twitter provides two kinds of tags:

- Hashtags (#): Indicates a topic on Twitter. An example of hashtag is #Euro2012 which signifies the Euro Cup held in 2012. Users who post tweets about Euro Cup append #Euro2012 in the text of their tweet.
- Mention tags (@): The @tag is used to either mention a fellow Twitter user or reply to one of his tweets. The tweets with @tags are displayed in the mentioned user's timeline. For example, a tweet with @John will appear in John's profile where 'John' is a Twitter username.

Twitter facilitates searching tweets based on topics. One who is interested in the Euro Cup can search for #Euro2012 to obtain a list of Euro Cup related tweets posted on Twitter. When the topics are very popular, the hashtag or topic become a "trending" topic. Trending topics are always displayed on a user's Twitter homepage (depending on their settings for the location). Thus, malicious users hijack such trending topics by posting phishing tweets with popular trending hashtags irrespective of their relevance to increase their reach and visibility. Also, the @tag allows any user to direct tweets to any other user in Twittersphere irrespective of whether they are friends / followers. Malicious users take advantage of this feature and direct phishing tweets to random users through the @ tag. Thus, malicious tweets have higher number of hashtags and @tags so that the tweet is directly visible to the mentioned users and the users searching for a topic on Twitter using the mentioned hashtags. Hence, we include such tweet based features for phishing detection.

7.4 User Attributes and Network based Features

Friend relationships on Twitter are unidirectional and described by the following:

- Followers of a user X are those Twitter users who subscribe to X's tweets. Whenever X posts a tweet, it appears in his follower's timeline
- Followees of a user X are those Twitter users whom X has subscribed to. X gets all the tweets posted by his followees in his timeline.

Studies on Twitter spam show that spammers have different tweeting behavior when compared to legitimate users. For example, spammers often post automated tweets in large numbers usually at predefined intervals of time [4]. Also, it has been observed that malicious users have a large number of "followees" but a small number of "followers." Thus, we use features such as number of tweets posted, Follower-Followee ratio and other Twitter profile information like the description of the Twitter user and presence of profile image for phishing detection.

Chapter 8

PhishAri API and Browser Extension

Our goal in this research work is to provide realtime protection from phishing to Twitter users. To enable this, we built a browser extension for Twitter and a supporting API to indicate whether a tweet is phishing or not.

8.1 Browser Extension

A large fraction of Twitter users use web browser to access Twitter. [1] Users are usually hesitant to change the platforms they use. Therefore, we built a browser extension which seamlessly integrates phishing detection results into the user's Twitter pages. The extension once installed shows a green indicator next to tweets which are safe and a red indicator next to phishing tweets. The detection mechanism is designed such that it requires no extra clicks or key press. The extension works for any tweet which appears either in a user's timeline, Twitter search results or tweets on the homepage of other Twitter users. PhishAri browser extension also works for Direct Messages (DM) of a user if the URL in the DM has been detected as phishing by a blacklist. Figure 8.1 shows the red and green indicators at the end of the URL in each of the tweets.

The current version of PhishAri extension works for 'Chrome' browser and is written in Javascript. The browser extension extracts the tweet ID [2] of a tweet and then makes a request to the PhishAri API hosted on a separate server. The API takes the tweet ID as input and returns back a string indicating whether the tweet is 'phishing' or 'safe.' Accordingly, PhishAri extension displays either a red or a green indicator in front of the tweet. This whole process is very robust and it takes a maximum of 0.522 seconds for an indicator to appear for a tweet. However, this time is dependent on various factors such as the speed of feature extraction, Internet bandwidth and time to query Twitter API. We elaborate our system configuration which affects the feature extraction and classification time. Figure 8.1 shows a screenshot of the extension which is available on Chrome Web Store for free download.

[1] `http://blog.twitter.com/2010/09/evolving-ecosystem.html`

[2] tweet ID is the numeric unique identifier of a tweet

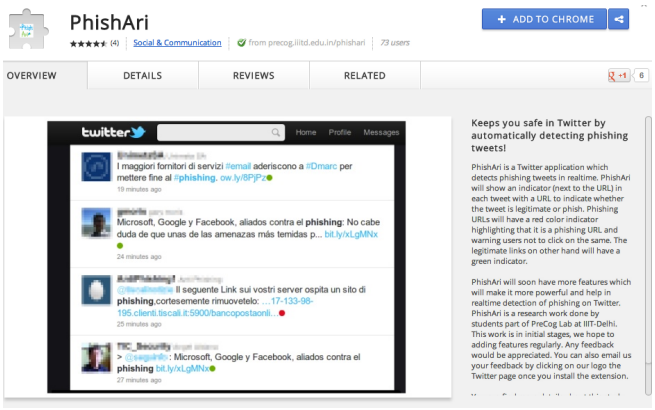

Figure 8.1: PhishAri on Chrome. Currently, there are more than 70 active users using the extension. The green indicator shows that the tweet is 'safe' whereas, a red indicator appears in front of 'phishing' tweets.

8.2 PhishAri API

PhishAri API is a RESTful API written in Python using mod_wsgi [3] framework. mod_wsgi framework enables the Apache server to host a Python application. The API is hosted on an Intel Xeon 16 core Ubuntu server with 2.67 GHz processor and 32 GB RAM.

The API provides a POST method to submit tweets for analysis. Once a tweet is submitted to the API, it classifies the URL as 'phishing' or 'safe' with the help of the set of features described in Chapter 7 using a trained classifier model pre-loaded on the server. Since our goal is to provide realtime indication to Twitter user, we require the time period for feature extraction and classification to be very less. To facilitate this, the API has multiprocessing modules which extract independent features simultaneously, hence saving a large amount of time in processing. Once the classification is done, the decision is output in form of a JSON string.

Figure 8.2 shows the integration of PhishAri browser extension with the PhishAri API. The extension sends a POST request to the API with the tweet ID. Once the API gets the tweet ID, it extracts all information about the tweet using the features mentioned in Chapter 7. These features include URL specific features, Twitter user information and details about the Twitter network of the user. Using these features, the API constructs a feature vector which is used for classification by comparing the feature vector to a pre-loaded classifier model for phishing tweet detection. Once the decision is made, the API returns back a JSON object indicating whether the tweet is phishing or not.

[3] `http://code.google.com/p/modwsgi/`

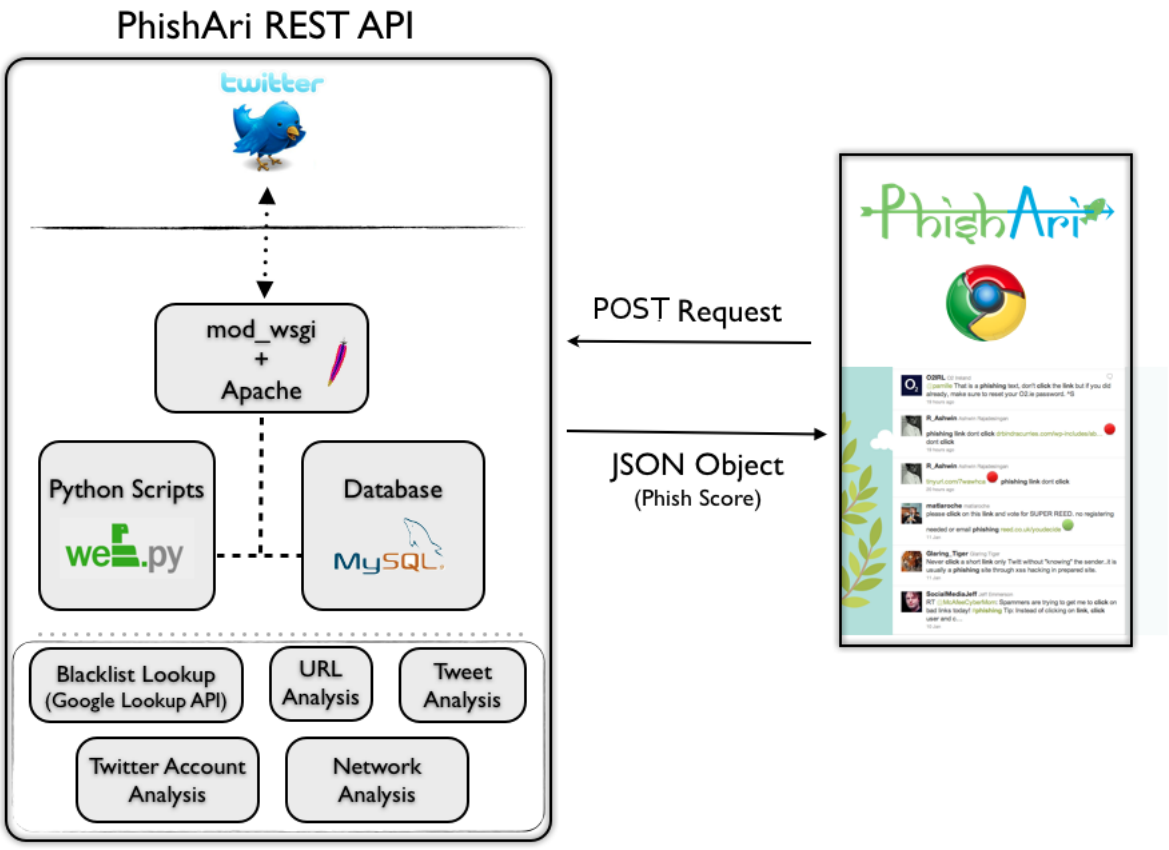

Figure 8.2: Integration of PhishAri API with the browser extension. The extension sends tweet ids and URLs to the API through a POST request. The API responds with the results based on which the red or green indicators are embedded to the corresponding tweets by the extension

Chapter 9

Experimental Setup

In this section, we describe the mechanism used for classification of phishing tweets. Our aim is to detect phishing tweets in realtime. In order to build such a mechanism, we need to identify the correct and most efficient classification methodology for which we setup the experiment. In this section, we explain the experimental setup for our study. We describe various machine learning techniques we use for phishing tweets classification. Machine learning techniques involve classification of an unseen data point using a classification model built on a pre-labeled (already classified) dataset. Hence, our experiment involves three stages. In the first stage, to create a labelled dataset, we collect tweets with URLs and label these tweets as 'phishing' or 'safe.' In the second stage, we train a classifier model using a classification algorithm. In the third stage, whenever we obtain a tweet with URL, we use the trained model to make a classification decision for this newly appeared URL. Now, we describe the machine learning algorithms we use for our study and the evaluation metrics which indicate the quality and the accuracy of our classification task.

9.1 Machine Learning Classification

To evaluate the most effective technique for phishing detection on Twitter, we investigate the use of multiple classification algorithms. This section explains these algorithms in brief and details on how we use them for our phishing detection task.

9.1.1 Naive Bayes

This is a probabilistic classifier and is based on the Naive Bayes' theorem. It works efficiently when the dimensionality of the input feature vector is high and each feature is independent of each other. Based on each feature, the Naive Bayes classifier computes the likelihood of the data point to be classified into each possible category. The data point is then classified into the category for which the likelihood (probability) is the highest. For this study, we use the *naivebayes* module of Python NLTK *classify* package. [1]

[1] `http://nltk.org/api/nltk.classify.html#module-nltk.classify.naivebayes`

9.1.2 Decision Trees

This is a widely used machine learning technique. It is based on a predictive model which creates a classification tree. Decision tree algorithm creates a model that predicts the category of the target data point by learning simple decision rules inferred from the data features. We use '*DecisionTreeClassifier*' module provided by '*scikit*' library. [2]

9.1.3 Random Forest

Random Forest is one of the most accurate classifiers and it works efficiently for large databases. For each data point to be classified, this technique randomly chooses a subset of features which are used for classification. It selects the most important features of the data point hence improves the predictive accuracy and controls over-fitting. We use '*RandomForestClassifier*' module provided by '*scikit*' library for this study.

9.2 Training and Testing Data

We perform a 5 fold cross-validation for computing the classification results. The labeled dataset is partitioned into 5 subsets. In each test run, 4 subsets are used for training and the remaining subset is used as test data. Hence, we classify using 5 test run which ensures that each set has been used for training as well as testing. The final classification result is the average of results from the 5 classification runs.

[2] `http://scikit-learn.org/`

Chapter 10

Results

As stated earlier, our study consists of two parts. In the first part, we develop a classification model based on various features like URL based and Twitter based features and classify tweets accordingly as phishing or safe. This forms our PhishAri API which uses a trained model and classifies incoming tweets based on the described features. In the next step, we create an end-user solution by deploying a Chrome extension which makes a call to the above API and public blacklists and then marks each tweet as phishing or safe with the help of a color-coded marker.

In this section, we elaborate the results of the first part of our study, i.e., the results and observations based on the classification mechanism using the four set of feature sets described in Chapter 7.

10.1 Evaluation Metrics

In order to evaluate the effectiveness of our classification method based on the features described, we use the standard information retrieval metrics viz. accuracy, precision and recall. Precision of a class is the proportion of predicted positives in that class that are actually positive. Recall of a class is the proportion of the actual positives in that class which are predicted positive. To explain this further, we use the 'confusion matrix' described in Table 10.1.

Table 10.1: Confusion matrix for classification.

		Predicted	
		Phishing	**Safe**
Actual	**Phishing**	TP	FN
	Safe	FP	TN

Each entry in the table indicates the number of elements of a class and how they were classified by our classification method. For example, 'TP' is the number of phishing tweets which were correctly classified as phishing. Using this confusion matrix, we can compute the precision (Equation 10.1) and recall (Equation 10.2) for both 'phishing' and 'safe' classes. We also use the confusion matrix to compute the overall 'accuracy' (Equation 10.3) of the classifier. It is the ratio of the correctly classified elements of either class to the total number of elements.

$$Precision_{phishing} = TP/(TP + FP) \tag{10.1}$$

$$Recall_{phishing} = TP/(TP + FN) \tag{10.2}$$

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \tag{10.3}$$

10.2 Classification Results

We now describe the results of our classification experiment as described in Chapter 9. We use three classification methods for our study viz. Naive Bayes, Decision Trees and Random Forest. We present the results of classification task using all these methods.

From the 1,589 phishing tweets, we found that 1,473 tweets had unique text. Therefore, is our true positive dataset, we consider these 1,473 phishing tweets and 1,500 safe tweets chosen randomly from the tweets marked as 'safe' during our data collection process. We use this dataset for the rest of our classification experiments. We found that Random Forest classifier works best for phishing tweet detection on our dataset with a high accuracy of 92.52%. We also obtain a recall of 92.21% for phishing class and 96.82% for safe class. The results from the three classification techniques are described in the table 10.2. It is observed that when we used Random Forest classifier, we also achieved a high recall and precision for both 'phishing' and 'safe' classes. It is important in our study to achieve a good precision of both classes to reduce the number of false negatives and false positives. Precision-accuracy balance is hard to achieve and we notice that the precision of phishing class drops but accuracy increases when we move from Naive Bayes classifier to Decision Tree classifier. However, we finally achieved a desirable precision and accuracy when we used Random Forest classifier. Random Forest reduces false positives and hence the precision of both the classes increased significantly.

Table 10.2: Results of classification experiments. We observe that Random Forest performs the best with an accuracy of 92.52%

Evaluation metric	**Naive Bayes**	**Decision Tree**	**Random Forest**
Accuracy	87.02%	89.28%	92.52%
Precision (phishing)	89.21%	88.05%	95.24%
Precision (safe)	92.12%	94.15%	97.23%
Recall (phishing)	68.32%	74.51%	92.21%
Recall (safe)	85.67%	89.20%	95.54%

Previous studies show that Random Forest outperforms all classifiers for phishing email detection with an error rate of 7.72% [1]. We find that the superior performance of Random Forest for phishing detection on Twitter also holds true with a high accuracy. We further investigate the performance of Random Forest classification method by using the confusion matrix described in Table 10.3. We show that we could detect 92.31% phishing tweets correctly. However, we misclassified 9.6% of legitimate tweets as phishing tweets. The false negative percentage is low indicating that we classified only 7.78% phishing tweets as legitimate. The misclassification of phishing tweets as legitimate tweets happens because some phishing tweets exhibit similar features as legitimate tweets. We manually observed a sample of such misclassified tweets and found that there are Twitter accounts which often exhibit dual behavior by sometimes posting legitimate tweets and sometimes phishing tweets. These users are either already compromised or due to negligence, retweet a phishing tweet. Hence, tweets from such users are misclassified, as their behavior and attributes are very similar to both legitimate users and phishers. Since our classification methodology takes into account Twitter based features, with the evolution of phishing techniques on Twitter, if a malicious user makes the phishing tweet look like a legitimate tweet and has Twitter network features as that of a legitimate user, our classification

method may misjudge the phishing tweet as legitimate.

Table 10.3: Precision and Recall for phishing detection using Random Forest based on all four feature sets.

		Predicted	
		Phishing	**Safe**
Actual	**Phishing**	**92.31%**	7.78%
	Safe	9.60%	**94.41%**

10.3 Evaluation of various Feature Sets

Most of the previous studies to detect phishing have used features based on the URL of the suspicious page and the HTML source of the landing page. In this study, we propose to use Twitter based features along with URL based features to quickly detect phishing on Twitter at zero-hour. To evaluate the performance of detection using these additional set of features based on Twitter properties, we present feature-set wise performance of the classification technique we use.

Table 10.4: Feature set wise performance of classification of Phishing Tweets.

Feature Sets	**Precision (Phishing)**	**Precision (Safe)**	**Recall (Phishing)**	**Recall (Safe)**	**Accuracy**
F1	81.27%	88.21%	79.25%	91.34%	82.22%
F1 + F2	86.11%	89.92%	85.21%	92.21%	87.31%
F1 + F2 + F3	91.10%	94.66%	88.32%	92.88%	90.03%
F1 + F2 + F3 + F4	95.24%	97.23%	92.21%	95.54%	92.52%

As described in Table 7.1, we have used four sets of features in this study. To evaluate the impact of each feature set, we performed classification task by taking one feature set at a time and then added the other one in the next iteration. Table 10.4 presents our experiment results by using different set of features using Random Forest classification method which gives us the overall highest accuracy of 92.52%. We observe that when we use only URL based features, we get an overall accuracy of 82.22% and a low precision and recall for 'phishing' class. The addition of Twitter based feature sets, user based features and network based features significantly improve the performance of phishing detection and boost the precision of identifying phishing tweets significantly. Hence, Twitter based features are helpful in increasing the performance of classifying phishing tweets.

10.4 Most Informative Features

We now evaluate the most important features which help to decide whether a tweet is phishing or not. We use 'scikit' library to find out the most informative features. Random Forests deploy ensemble learning to evaluate the feature importance. After each random tree is constructed using a set of features, its performance (misclassification rate) is calculated. Then the values of each feature is randomly permuted (for each feature) and the new misclassification rate is evaluated. The best performing features are then chosen as the most informative features. The

most informative features which we found for phishing tweet detection using Random Forest classification are described in Table 10.5.

Ownership period is one of the most important features in phishing detection. The domains of malicious and phishing URLs tend to be short lived when compared to the domains of legitimate URLs in order to avoid detection. Similarly the age of Twitter account of the user posting phishing tweets is also generally less. Such users are often detected by Twitter and their accounts are suspended. However, using PhishAri API, we could detect a large number of phishing tweets by such users before they were suspended by Twitter.

Another important feature is the presence of conditional redirects. Many phishing websites redirect the user to a legitimate website instead of the phishing landing page if the page is being accessed via an automated script or bot. In our experiment, we compare the landing URL when the suspected URL is accessed by the browser simulation and bot simulation. In case the landing URLs are different, there is a high possibility that the website is malicious. The redirection to a legitimate website when accessed by an automated script is to avoid detection by bots such as googlebots traversing through the Internet.

We also find that presence of trending #tags in a tweet is an important feature for phishing detection. Phishers often hijack trending topics and start posting unrelated content in their tweets with the trending #tag appended. This increases the visibility of their tweet as trending topics specific to a location are always displayed on the homepage of a Twitter user.

Phishers usually have more number of followees than followers. Since relationships on Twitter are unidirectional, a Twitter user needs trust to be followed by another user. Since phishers do not often post legitimate content, very few Twitter users tend to follow phishers. However, phishers follow a lot of users in the hope of being followed back. Hence the ratio of Follower-Followee is very skewed in case of phishing tweets.

Another technique used by phishers to gain visibility is to directly mention other Twitter users in their tweets. Phishers tend to have a lot of @tags in their tweets so that their tweet is directly visible to the mentioned users. Since the mentioned users receive these tweets in their timeline, there is a high chance that the target users click on the links and fall victim to phishing attacks.

Table 10.5: Most informative features for detecting phishing tweets.

Ranking	**Feature**
1	Ownership period
2	Age of account
3	Presence of conditional redirects
4	Presence of trending #tags
5	Number of Redirections
6	Follower-Followee Ratio
7	Number of @tags

10.5 Comparison of PhishAri with Blacklists

The inherent problem of the blacklists is that they are slow to catch phishing URLs. Since Twitter provides a realtime stream of tweets to a user, it is important that the tweets are detected as phishing as soon as they appear to the user. Blacklists in such cases prove to be ineffective. To support our claim, we compare the performance of PhishAri with two public blacklists, Google Safebrowsing and PhishTank.

At the time of data collection, we collected realtime stream of tweets from Twitter and immediately look up the URLs present in the tweet in these blacklists. Since blacklists take some time to add newly created phishing URLs, we wait for 3 days and again lookup the URLs collected 3 days ago in the Google safebrowsing and PhishTank blacklists. We also use PhishAri to classify each of these tweets as phishing or safe.

We found that 80.6% unique phishing tweets were detected as phishing at zero-hour by PhishAri which were caught by the blacklists only later when we checked after 3 days. Public blacklists are often based on crowdsourcing (like PhishTank) or use URL based or landing page based features. However, phishers often keep changing their strategies and hence these detection mechanisms by blacklists often fail. We couple these features along with other features for a better phishing detection to obtain efficient realtime detection. This shows that PhishAri can complement the blacklisting mechanism for Twitter to detect more phishing URLs in realtime.

10.6 Comparison of PhishAri with Twitter

Twitter has its own detection mechanism for catching malicious, spam and phishing tweets. In case a URL in a tweet is not safe, Twitter shows a warning page to the user when one tries to navigate to that URL from Twitter. However, we found that Twitter's mechanism was not as quick and was unable to catch a large fraction of phishing URLs appearing in tweets in realtime.

To compare the performance of PhishAri API with Twitter's detection mechanism, we check whether Twitter marks a URL as safe or not at the time it is submitted to Twitter stream. Then, we again check the status of the URL after 3 days. Out of 3,09,321 tweets with URLs, we found that 492 tweets were undetected by Twitter at the time of data collection, however they were marked as 'suspicious' URLs only later when we checked after 3 days. However, PhishAri was able to detect 84.6% of these phishing tweets at zero-hour which were blacklisted by Twitter later. This shows that PhishAri if implemented along with Twitter's malicious tweets detection mechanism, can help boost the performance of realtime detection of phishing on Twitter.

(a) (b)

Figure 10.1: Figure 10.1(a) shows the most frequent words of phishing tweets in our dataset. Figure 10.1(b) shows the frequent words occurring in a sample of legitimate tweets from our dataset. Both the tagclouds have random 50 tweets. In case of phishing tweets there is a dominance of certain words which signify the spam campaign promoted at that time, however, the legitimate tweets have almost all words occurring with equal probability.

10.7 Time Evaluation

One of the major aims of this study is realtime detection of phishing tweets. Hence our mechanism needs to be robust enough to quickly classify a phishing tweet. We now evaluate how much time PhishAri takes to classify a URL. As mentioned before, a classifier model is preloaded on our server which is used to make decisions about a tweet. The PhishAri API is written using multiprocessing modules so that it can extract independent features simultaneously, hence increasing the speed of computation. We find that the time required for the feature extraction and classification of a tweet is a maximum of 0.522 seconds (Min: 0.167 sec, Avg: 0.425 sec, Median 0.384 sec). This time was taken when we ran our experiments on an Intel Xeon 16 core Ubuntu server with 2.67 GHz processor and 32 GB RAM. However, we must note that the speed of classification is also dependent on the response times of the Twitter API, the WHOIS repository and also the Internet bandwidth.

10.8 Characteristics of Phishing Tweets

We also found that the words used in case of phishing tweets are different from those used in legitimate tweets. Phishing tweets often have keywords which are specifically used to lure the unsuspecting Twitter user into clicking the URL. The content of the tweet is often appealing enough and promises some kind of benefit to the user if one visits the URL. Figure 10.1(a) shows the most popular words which appear in phishing tweets. We see that 'product,' 'allow,' etc. are the most popular words. They appear repeatedly because of a phishing campaign which asks Twitter details in return for more Twitter followers.

The text of phishing tweets is considerably different from that of legitimate tweets, where people usually talk about general topics and use a variety of words unlike phishing tweets which use a limited set of words. Figure 10.1(b) shows the word tag cloud of a sample of legitimate tweets. The words occurring in legitimate may also depend on the trending hashtags at the time tweets were posted. However, the text for phishing tweets remains relatively the same for a particular phishing campaign irrespective of the trending topic. However, phishing tweets contain the hastags which are trending at the time they were posted to gain visibility.

We also try to ascertain the country of the origin of phishing tweets in our dataset. We find that USA has maximum number of users posting phishing URLs followed by Brazil. The geomap in Figure 10.2 shows the concentration of phishing URLs originating from various countries across the world on Twitter. Manual evaluation shows that many of the phishing accounts were indeed from USA. However, it must be noted that the phishers could've falsely selected the country as USA in their Twitter bio page. Also, more than 25% of all Twitter users are from USA, thus, it might seem natural that there are more phishing tweets originating from there. [1]

10.9 PhishAri Extension for Chrome Browser

In this section, we evaluate the realtime browser extension we built for phishing detection. The extension works for Chrome browser and has currently more than 70 active users. We evaluate user experience of our extension and present statistics about how our extension is being used by Twitter users.

[1] http://venturebeat.com/2012/07/30/twitter-reaches-500-million-users-140-million-in-the-u-s/

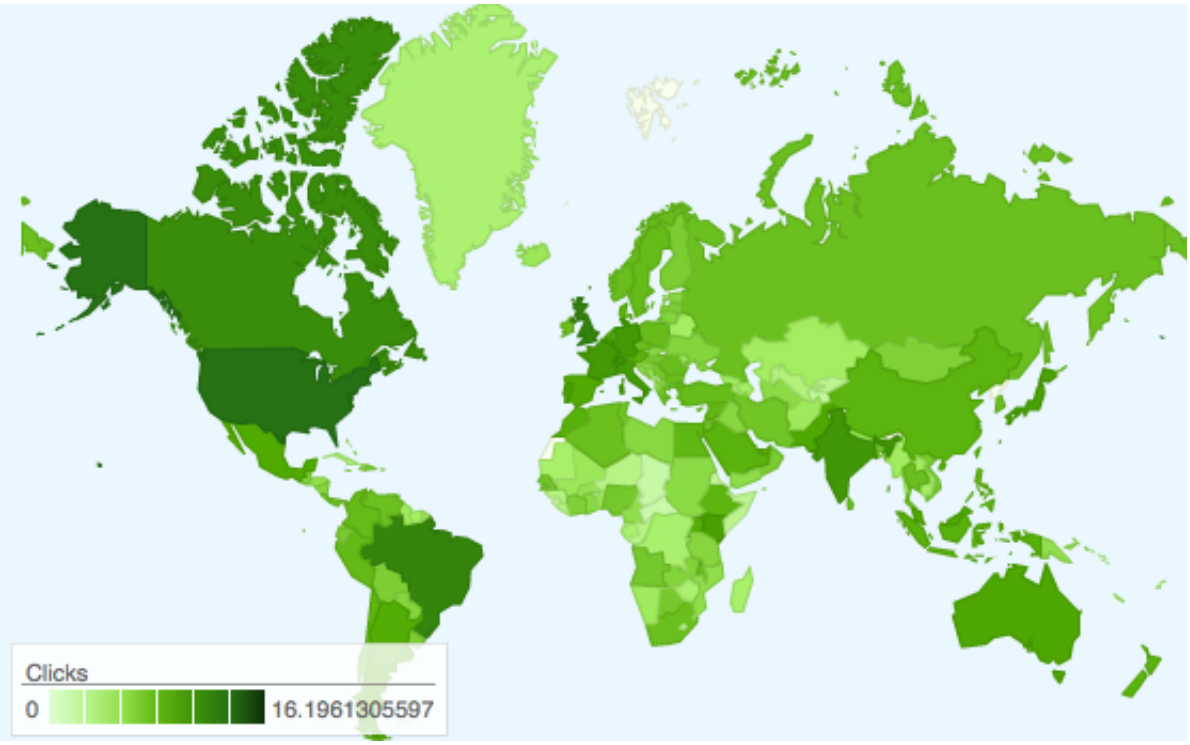

Figure 10.2: Countries from where phishing tweets originate.

10.9.1 User Experience

There have been user studies to evaluate and assess the user-experience of browser based tools [9]. The users of the study are asked to use the tool and give feedback about the system. We performed a lab study to find out the user experience with PhishAri and whether users find the extension effective and useful. The lab study consisted of 10 users out of which 7 were males and 3 females. All users were active on Twitter. They were given the download link of the PhishAri extension page which also gave details about the extension and described how it worked. Each user was asked to browse Twitter after installing the PhisAri extension and were asked if they found the plugin effective and easy to use. All users said that the extension was very easy to use and the indicator displayed with every tweet did not adversely affect their Twitter experience. However, 4 users commented that they prefer using Twitter clients over browser. Currently, PhishAri provides support only for browser based Twitter access. For all 10 users, the color coded indicators appeared as soon as the tweet loaded without any visible delay. However, 5 users observed a time lag in the appearance of the indicators when they were browsing tweet stream of a trending topic on Twitter. In future, we will try to make PhishAri faster to fill this gap. Since phishing tweets are not abundantly present as compared to other tweets, and we conducted a time-limited lab study, we created a dummy Twitter account which had a mix of phishing and legitimate tweets. Users were asked to go to the dummy Twitter account to check if a red indicator appears for phishing tweets or not. Two users commented that whenever there is a red indicator for a tweet, they would like to see a preview of the landing page (as in web-based systems likes *PhishTank*) when they hover over the indicator. We think that adding this feature in future would be useful to gain the confidence of user.

Users were asked if they would be interested to use the PhishAri extension daily in regular use. Except the 4 users who prefer using Twitter client over browser, all other users said that PhishAri seems to be a useful tool. These users also said that they would like to have a similar

spam detection tool for Twitter which indicates whether a tweet (irrespective of the presence of a URL) is spam or legitimate. However, the scope of PhishAri is currently to indicate whether a tweet which has a URL is 'phishing' or 'safe.' The lab study showed that PhishAri works with ease and is non-intrusive; the indicators do not distract the users attention while browsing Twitter. The color coded indicators are effective in indicating the status of a tweet but could be improved by showing an optional preview of the landing page when the cursor is hovered over the indicator.

10.9.2 Statistics

We present some statistics about PhishAri browser extension. We have Google Analytics [2] enabled for our extension which helps us track the user details like the country and the active time of user using the extension. We found that we have a wide diversity of users from various countries with highest traffic from the US and India. Table 10.6 shows the percentage of users from various countries who use PhishAri.

Table 10.6: PhishAri Chrome extension users from various countries across the world.

Country / Territory	**Users**
United States	32.59%
India	28.09%
Germany	8.20%
Saudi Arabia	6.90%
United Kingdom	3.62%
Greece	3.35%
France	2.93%
Russia	2.70%
Slovakia	2.25%
Egypt	2.09%
Singapore	1.41%
Morocco	1.29%

[2] `http://www.google.com/analytics`

Chapter 11

Limitations, Conclusion and Future Work

As a first step, we analyzed and discussed phishing attacks on online social media using URL shorteners. A shortened URL makes the length of actual URL shorter and hides the long URL behind it. We found that space gain for half of phishing URLs in our dataset was 37% which was significantly less than space gain in general URLs on the Internet. Also the tweets which quoted these URLs had length more than the general. This points to malicious intent of phisher for hiding their URLs behind bland hashes and entice viewers through words. The new identity in form of bit.ly domain name gives a sense of trust to the viewer and takes him / her to the trap page. We also found that online social media brands namely Facebook, Habbo and Orkut were amongst the top targets competing with e-commerce services. This shift in phishers' approach can be attributed to rapid growth, wider and gullible audience and open platform of online social media. We also found that geographically, USA, India and Brazil were the target countries of phishers. Phishers have realized that developing countries are fast catching up with the developed and are untapped "markets" for luring victims. Finally, we traced the footprints of phishers. Most referrals for these phishing URLs arose from online social media. Facebook, Orkut and Twitter combined accounted for 2/3rd of all referrals. Most of the Twitter accounts publishing these tweets were *inorganic* (automated). Third party applications were extensively used for this purpose.

In the next step to detect phishing on Twitter, we built PhishAri. Our methodology exploits not just the traditional phishing detection features which are based on the URL and the suspicious landing page, but also Twitter specific and WHOIS based features. We use a combination of URL based and Twitter based features which help in an effective and realtime detection of phishing on Twitter. As a proof of concept, we also develop a RESTful API which can be accessed using an HTTP POST method. We also implement a Chrome browser extension which makes a call to this API and accordingly shows an indicator next to each tweet indicating whether the tweet is phishing or not. We also show that our methodology works faster than standard blacklisting mechanism and Twitter's own defense mechanism. We were able to detect 80.6% more URLs than popular blacklists like PhishTank and Google Safebrowsing at zero-hour with an accuracy of 92.52%. Similarly, our detection mechanism also works better than Twitter's defense system by 84.6% at zero-hour. Since we do not achieve a 100% accuracy, there is always a possibility of false negatives. However, our method can be coupled with blacklisting and Twitter's defense mechanism for a better, more accurate realtime detection of phishing on Twitter.

Now we discuss how we can further improve PhishAri for more efficient and robust phishing

detection.

Backend database for faster lookup In future, we can maintain a cache backend database to capture tweets which have already been marked as either phishing or safe on Twitter. So, if a tweet with same URL appears on Twitter, then we can skip the entire process of feature extraction and classification and lookup in our dataset of phishing URLs and safe URLs. This will also help us increase our own database of phishing tweets.

Increase the scope of PhishAri from public to all tweets Currently, PhishAri can detect whether a tweet is phishing or not only if the source Twitter user of that tweet is a public user. Otherwise PhishAri is unable to extract the user specific information. In future, we will implement oauth integration of Twitter with PhishAri so that it can detect a wider range of phishing tweets. However, this is just a proof of concept and does not affect our methodology in any way.

Bibliography

[1] Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. A comparison of machine learning techniques for phishing detection. In *Proceedings of eCrime researchers summit* (2007), ACM, pp. 60–69.

[2] Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E., and Karagiannis, T. we. b: The web of short urls. In *Proceedings of the 20th international conference on World wide web* (2011), ACM, pp. 715–724.

[3] Barracuda. Warning: New facebook phishing via facebook chat and note. `http://www.barracudalabs.com/wordpress/index.php/2012/04/06/warning-new-facebook-phishing-via-facebook-chat-and-note/`, April 2012.

[4] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)* (2010).

[5] Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J., and Gonçalves, M. Detecting spammers and content promoters in online video social networks. In *ACM SIGIR conference on Research and development in information retrieval* (2009), ACM, pp. 620–627.

[6] Chandrasekaran, M., Narayanan, K., and Upadhyaya, S. Phishing email detection based on structural properties. In *NYS Cyber Security Conference* (2006).

[7] Chhabra, S., Aggarwal, A., Benevenuto, F., and Kumaraguru, P. Phi. sh/social: the phishing landscape through short urls. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (2011), ACM, pp. 92–101.

[8] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference* (2010), ACM, pp. 21–30.

[9] Cranor, L., Arjula, M., and Guduru, P. Use of a p3p user agent by early adopters. In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society* (2002), ACM, pp. 1–10.

[10] Fette, I., Sadeh, N., and Tomasic, A. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 649–656.

[11] Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., and Zhao, B. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th annual conference on Internet measurement* (2010), ACM, pp. 35–47.

[12] Grier, C., Thomas, K., Paxson, V., and Zhang, M. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security* (2010), ACM, pp. 27–37.

[13] Jagatic, T., Johnson, N., Jakobsson, M., and Menczer, F. Social phishing. *Communications of the ACM 50*, 10 (October 2007), 94–100.

[14] Jakobsson, M., and Myers, S., Eds. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley-Interscience, 2006.

[15] Krackhardt, D. Simmelian ties: Super strong and sticky. *Power and influence in organizations 21* (1998), 38.

[16] Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L., Hong, J., and Nunge, E. Protecting people from phishing: the design and evaluation of an embedded training email system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 905–914.

[17] Kwak, H., Lee, C., Park, H., and Moon, S. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 591–600.

[18] Lee, K., Eoff, B., and Caverlee, J. Seven months with the devils: A long-term study of content polluters on twitter. In *Int'l AAAI Conference on Weblogs and Social Media (ICWSM)* (2011).

[19] Lee, S., and Kim, J. Warningbird: Detecting suspicious urls in twitter stream. *NDSS 2012* (2012).

[20] MarkMonitor. Fraud intelligence report. `https://www.markmonitor.com/mmblog/q1-2012-fraud-intelligence-report/`, June 2012.

[21] Mashable. Warning: Twitter phishing scam spreads by direct messages. `http://mashable.com/2011/10/26/warning-twitter-spam/`, October 2011.

[22] Moore, T., and Clayton, R. An empirical analysis of the current state of phishing attack and defence. In *Workshop on the Economics of Information Security* (2007).

[23] Moore, T., and Clayton, R. Examining the impact of website takedown on phishing. In *Proceedings of eCrime researchers summit* (2007), ACM, pp. 1–13.

[24] RSA. Cyber security awareness month fails to deter phishers. http://www.rsa.com/solutions/consumer_authentication/intelreport/11541_Online_Fraud_report_1011.pdf, October 2011.

[25] Securelist. Spam report. http://www.securelist.com/en/analysis/204792234/Spam_report_May_2012, May 2012.

[26] Sheng, S., Magnien, B., Kumaraguru, P., Acquisti, A., Cranor, L. F., Hong, J., and Nunge, E. Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security* (2007), pp. 88–99.

[27] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., and Zhang, C. An empirical analysis of phishing blacklists. In *Sixth Conference on Email and Anti-Spam (CEAS)* (2009).

[28] Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D. Design and evaluation of a real-time url spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on* (2011), IEEE, pp. 447–462.

[29] Wang, A. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on* (2010), IEEE, pp. 1–10.

[30] Xiang, G., Hong, J., Rose, C., and Cranor, L. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC) 14*, 2 (2011), 21.

[31] Zhang, C., and Paxson, V. Detecting and analyzing automated activity on twitter. In *Passive and Active Measurement* (2011), Springer, pp. 102–111.

[32] Zhang, Y., Egelman, S., Cranor, L., and Hong, J. Phinding phish: Evaluating anti-phishing tools. In *In Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS 2007* (2007).

[33] Zhang, Y., Hong, J., and Cranor, L. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web* (2007), ACM, pp. 639–648.

[34] ZoneAlarm. The dark side of social media, how phishing hooks users. http://www.infographicsarchive.com/social-media/the-dark-side-of-social-media-how-phishing-hooks-users/, 2012.