

An efficacious method for detecting phishing webpages through target domain identification



Gowtham Ramesh ^{a,*}, Ilango Krishnamurthi ^b, K. Sampath Sree Kumar ^a

^a Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, Tamilnadu, India

^b Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Kuniamuthur, Coimbatore, Tamilnadu, India

ARTICLE INFO

Article history:

Received 26 February 2013

Received in revised form 1 December 2013

Accepted 7 January 2014

Available online 17 January 2014

Keywords:

Phishing

Anti-phishing

E-commerce security

Target domain detection

ABSTRACT

Phishing is a fraudulent act to acquire sensitive information from unsuspecting users by masking as a trustworthy entity in an electronic commerce. Several mechanisms such as spoofed e-mails, DNS spoofing and chat rooms which contain links to phishing websites are used to trick the victims. Though there are many existing anti-phishing solutions, phishers continue to lure the victims. In this paper, we present a novel approach that not only overcomes many of the difficulties in detecting phishing websites but also identifies the phishing target that is being mimicked. We have proposed an anti-phishing technique that groups the domains from hyperlinks having direct or indirect association with the given suspicious webpage. The domains gathered from the directly associated webpages are compared with the domains gathered from the indirectly associated webpages to arrive at a target domain set. On applying Target Identification (TID) algorithm on this set, we zero-in the target domain. We then perform third-party DNS lookup of the suspicious domain and the target domain and on comparison we identify the legitimacy of the suspicious page.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Phishing encourages end users to visit fake webpages which have similar look and feel of a webpage with the malicious intention to steal user credentials and identities. This identity theft is used for many illegal activities like online money laundering. The losses created as a result of these activities run into billions of dollars [1]. According to the RSA's online fraud report, in the year 2012 there is 59% increase in phishing attacks as compared to 2011 and an estimated loss of more than \$1.5 billion due to phishing attacks on the global organizations in the same period, which is 22% higher than in 2011 [2]. This results in people losing faith over the e-commerce industry and leads to significant loss in their market value [3]. Thereby there is a strong demand for an effective measure to curb such phishing attacks.

Any phishing attack usually involves first, creation of a fake website which looks similar to a legitimate website and then lures the users to these fake websites instead of the legitimate webpage, to provide the required authentication and other personal details. These details are extracted from the user without his knowledge. To mitigate this attack many possible counter measures have been developed by the researchers such as white-list based methods, blacklist based methods, heuristic approaches, hybrid approaches or multifaceted mechanisms.

All the aforesaid anti-phishing methods attempt to identify the phishing webpage, but lack techniques to identify the legitimate webpage that the phishing webpage mimics (phishing target [4]).

However, any anti-phishing technique becomes incomplete without identification of the phishing target, as it plays a vital role in confirming that there is a phishing attack on a legitimate webpage. Unfortunately, finding the target webpage can be tedious at times when phishers attack the less popular or new webpages. Sometimes, it is also difficult to identify the target because of the masquerading techniques used by the phishers. For example, if the phishers create the webpage using only embedded objects like images and scripts then identifying the target becomes tedious with the existing methods.

Hence, there is a need for a holistic approach that can identify the right phishing target even when attackers use any masquerading techniques. Such a method would gain significant importance among anti-phishing techniques as it alerts the target owners to take necessary counter measures and enhance security.

In this paper, we propose a novel approach to detect the phishing webpages. In this process, we take the webpage under scrutiny and identify all the direct and indirect links associated with the page and generate domain group sets S1 and S2 respectively. From these sets we identify the target domain set, which is given as input to Target Identification (TID) algorithm to identify the phishing target. Using DNS lookup, we map the domains of suspicious webpage and phishing target to corresponding IP addresses. On comparing both the IP addresses, we conclude the authenticity of the suspicious webpage. As our approach depends only on content of the suspicious webpage it requires neither a prior knowledge about the site nor requires the training data.

An overview of literature review and related work is presented in Section 2. Section 3 covers the system overview. In Section 4, we have explained the target domain set construction followed by the target

* Corresponding author.

E-mail address: rameshgowtham@gmail.com (G. Ramesh).

domain identification using TID-algorithm and phishing detection procedure in Sections 5 and 6 respectively. The implementation details, evaluation methodology and experimental results are discussed in Section 7. We conclude by highlighting the key features of our technique and its limitations in Section 8.

2. Related work

Various solutions to phishing have been developed during the past years. In this section, we briefly review some of the notable anti-phishing works and empirical studies based on it. The studies of different approaches have motivated us to propose a method that overcomes these limitations of the existing schemes.

The white-list approach maintains a list of all safe websites and their associated information. Any website that does not appear in the list is treated as a suspicious website. The current white-list tools usually use a universal white-list of all legitimate websites that need to be constantly updated. In order to simplify this, Han et al. [7] developed an approach to maintain an individual white-list which records the well-known legitimate websites of the user rather than maintaining a universal legitimate site list. In this approach, the Automated Individual White-List (AIWL) records every URL along with its LUI (Login User Interface) information and the legitimate IP addresses mapping to these URLs. Here the AIWL warns the user when the account information submitted to the website does not match with the entry in the white-list. This helps the user to distinguish a phishing website. This technique is adopted and suitably improvised for our work.

The blacklist approach maintains a list of known phishing sites to check the currently visiting website against the list. This blacklist is usually gathered from multiple data sources like spam traps or spam filters, user posts (e.g. phishtank) or verified phish compiled by third parties such as takedown vendors or financial institutions. Prakash et al. [8] used an approximate matching algorithm that divides a URL into multiple components that are matched individually against entries in the blacklist. Zhang et al. [9] proposed a system where customized blacklists are provided for the individuals who choose to contribute data to a centralized log-sharing infrastructure. This individual blacklist is generated by combining relevance ranking score and the severity score generated for each contributor. But the blacklist needs frequent updates from their sources and the exponential growth of the list demands great deal of system resources.

The heuristic-based approaches extract one or more features from a webpage to detect phishing instead of depending on any of precompiled lists. Most of these features are extracted from URL and HTML DOM (Document Object Model) of the suspicious webpage. Zhang et al. [10] proposed a content-based approach CANTINA, based on the tf-idf (term frequency and inverse document frequency) algorithm to identify top ranking keywords from the page content and meta keywords/description tags. These keywords are searched through a trusted search engine such as Google. Here, a webpage is considered legitimate if the page domain appears in the top N search results. CANTINA+ is an upgraded version of CANTINA proposed by Xiang et al. [11], where new components are included to achieve better results. Particularly, they have included ten other features along with four of the CANTINA features and one extended feature. In our approach we have used tf-idf similar to CANTINA to extract keywords from the webpage.

Another heuristic based approach exploring HTML DOM is “Phishark” developed by Prevost et al. [12]. In this research they have analyzed and studied the characteristics of phishing attack and have defined twenty heuristics to detect phishing webpages. These twenty heuristics were then checked for the effectiveness to decide as to which of these heuristics would play a major role in identifying both the phishing and the legitimate webpages. Since, these approaches do not require any pre-compiled lists, they are capable of detecting new phishing webpages by identifying anomalies in it, but legitimate sites also may have such anomalies when it is developed by novice

developers. These methods fall short in detecting a phishing webpage made up of only embedded objects like images and scripts.

The other area of research focuses on detecting phishing by comparing visual and image similarities between webpages. Fu et al. [13] proposed an approach which uses the Earth Mover's Distance (EMD) to measure webpage visual similarity. In this approach they first convert the webpage into low resolution images and then use color and coordinate features to represent the image signatures. EMD is used to calculate the signature distances of the images of the webpages. They used trained EMD threshold vector for classifying a webpage as a phishing or legitimate. Medvet et al. [14] proposed an approach which identifies phishing webpages, by considering text pieces and their style, images embedded in the page and the overall visual appearance features of the webpage. Chen et al. [15] present an image based anti-phishing system, which is built on discriminative key point features in webpages. Their invariant content descriptor and the Contrast Context Histogram (CCH) compute the similarity degree between suspicious and legitimate pages. Chen et al. [16] also proposed an approach which uses gestalt theory for detecting visual similarity between two webpages. They used the concept of super-signals to treat the webpage as indivisible unite; these indivisible super-signals are compared using the algorithmic complexity theory. But these techniques may result in false positive when a legitimate page crosses the similarity threshold value and also fails to identify the targeted page.

Multifaceted approaches use any combination of techniques in computational science to detect phishing websites. Joshi et al. [17] developed the PhishGuard tool that identifies phishing websites by submitting actual credentials after the bogus credentials during the login process of a website. They have also proposed architecture for analyzing the responses from server against the submission of all those credentials to determine if the website is legitimate or a phished one. Yue and Wang [18] designed a BogusBiter tool that submits a large number of bogus credentials along with the actual credential of users to nullify the attack. A similar approach has been applied by Joshi et al. [17] but BogusBiter is triggered only when a login page is classified as a phishing page by a browser's built-in detection component.

Shahriar and Zulkernine [19] proposed a model to test trustworthiness to suspected phishing websites. In a trustworthiness testing, they check if the behavior (response) of websites matches with the known behavior of a phishing or legitimate website to decide whether a website is phishing or legitimate. The model is explained using the notion of Finite State Machine (FSM) that captures the submission of forms with random inputs and the corresponding responses to describe the website's behavior. This approach can detect advanced XSS-based attacks that many contemporary tools currently fail to detect.

A category of research focuses on experimental studies to comprehend the significance of implementing anti-phishing strategies. Bose and Leung [5] demonstrated an experimental study showing that the firms that invest in adopting advanced phishing countermeasures earn trust of the customers which in turn reflects as encouraging return in their market value. Lai et al.'s [6] study on identity theft through coping perspective creates awareness for consumers, government agencies and e-commerce industries to counteract against such threats. Chen et al. [3] proposed a method to assess the possible financial loss of phishing targets. In this method key phrase extraction technique is used to discover the reports of phishing attack on firms. To estimate the potential financial loss of firms an event study was conducted to determine the change in market value after the release of phishing attack report. These studies clearly reveal the severity of phishing attacks and requirement of an effective anti-phishing method to protect firms and consumers.

Our work is also motivated by two multifaceted approaches that detect phishing targets along with the phishing webpage. These approaches are discussed in brief below.

Wenyin et al. [20] proposed to identify legitimacy of a given suspicious webpage and discovering its phishing target by calculating and reasoning defined association relations on its Semantic Link Network (SLN). This approach first finds the given webpage's associated pages and then

constructs a SLN from those webpages. They exploited a mechanism of reasoning on the SLN to identify whether the given webpage was a phishing page, and discovered its target if so. Although this method can detect a phishing webpage and find its target as well, there are cases where the system fails when phishing webpage contains few hyperlinks or the keywords extracted from the phishing webpage do not match with the keywords of the target webpage. If a legitimate webpage is not easily discovered by a search engine, it is likely to be identified as phishing. Moreover, the computational cost of building a semantic link network is expensive.

Wenyin et al. [21] also proposed an anti-phishing approach that identifies phishing targets using suspicious webpage's associated relationships. A webgraph is constructed from the associated pages and further partition of the graph results a web community (the parasitic community) for the given webpage. Parasitic coefficient is used to measure the parasitic relationship's strength from the given page to each page in the community. The page with the strongest parasitic relationship to the given suspicious webpage is regarded as the phishing target. If such a target is found, they identify the given suspicious webpage as a phishing webpage. Otherwise, they considered it legitimate. The construction of directly associated link set and indirectly associated link set in our method is adopted from this paper with additions and modifications.

Table 1 provides a brief summary of related works in comparison to our work with respect to five features. These includes efficiency of the methods in detection of phishing websites hosted in any language, detection of new phishing websites that have not yet been identified, detection of phishing webpages designed in embedded objects, detection of pharming based attacks and detection of phishing target.

3. System overview

Fig. 1 shows the overall system design. Our system identifies phishing websites based on the following certainty that for a phishing website, the target will be a legitimate site, whereas for a genuine website, the system will point to the genuine site itself as its own target. On this stand we identify the phishing webpage by comparing the suspicious webpage with its target.

For a given suspicious page, our method first identifies all the direct and indirect links associated with that page. The links which are directly associated with the webpage are extracted from the HTML source of the page and grouped based on their domains, as a set of domain S1. The indirectly associated links of the page are then retrieved by first extracting the keywords in the webpage and feeding these keywords to a search engine. We retrieve the first n links returned by the search engine as indirectly associated links and group them as a second set of domain S2. A reduced domain set S3 is constructed by extracting only the common domains present in both S1 and S2. This set S3 is fed as an input to a

TID algorithm, to identify the phishing target domain. We use DNS lookup to map the domain of the identified phishing target to its corresponding IP address. Similarly, we also map the domain of the suspicious webpage to its corresponding IP address. On comparing the two IP addresses we conclude the authenticity of the suspicious webpage.

4. Target domain set discovery

In this section, we discuss our approach for identifying the target domain set, consisting of domains which are directly and indirectly associated to the suspicious webpage. This set possibly contains a domain which is either the victim domain if the suspicious page is a phish or the domain of a legitimate page. In this approach we consider only domains of the associated webpage instead of considering the individual links. Our experiments have proved that this approach not only improves our prediction accuracy but also minimizes the computation overhead. We further discuss these merits in Section 7.

4.1. Keyword extraction

Keywords are important words in the webpage associated with a product or service. These keywords determine the web identity of a page. We apply term frequency-inverse document frequency (tf-idf) scheme [22] to extract the keywords set from various portions of a suspicious webpage e.g. title tag, meta tag (meta description tag, meta keywords tag), alt attribute of tags and body tag and create a document. This document is tokenized and pre-processed to eliminate the most commonly occurring words, stop words and words of size less than or equal to two before applying the tf-idf algorithm. The tf-idf algorithm assigns weight to every word in the document based on its importance to the document. A word that appears often in the document but rarely in the corpus will score high, while the other combinations will not. Finally we retrieve up to seven keywords from the resultant keywords set based on its tf-idf score [10]. These keywords are further used in identifying the indirectly associated links of the webpage as explained in the following section.

4.2. Extracting link sets

The connection between any phishing page and the webpage it imitates is the hyperlinks from the phishing webpage. Hence, analyzing the links in suspicious webpage is important to identify its target and to verify its legitimacy. Therefore, we build two link sets L1 and L2 from the suspicious webpage.

The links in directly associated link set (L1) are extracted from DOM object's properties of the suspicious webpage P, which includes href, src, and alt attribute of anchor, image, area, script and link tags. Here, we

Table 1
Summary of related work in comparison with our work.

Work	Language independence	Zero day protection	Embedded objects	Pharming based attack	Phishing target detection
Han et al. [7]	Yes	No	No	Yes	No
Prakash et al. [8]	Yes	No	No	No	No
Zhang et al. [9]	Yes	No	No	No	No
Zhang et al. [10]	No	Yes	No	No	No
Xiang et al. [11]	No	Yes	No	No	No
Prevost et al. [12]	No	Yes	No	No	No
Fu et al. [13]	Yes	No	Yes	Yes	No
Medvet et al. [14]	Yes	No	No	No	No
Chen et al. [15]	Yes	No	Yes	No	No
Chen et al. [16]	Yes	No	Yes	No	No
Joshi et al. [17]	Yes	Yes	No	Yes	No
Yue and Wang [18]	Yes	Yes	No	Yes	No
Shahriar and Zulkernine [19]	Yes	Yes	No	Yes	No
Wenyin et al. [20]	Yes	Yes	No	No	Yes
Wenyin et al. [21]	Yes	Yes	No	No	Yes
Our work	Yes	Yes	Yes	Yes	Yes

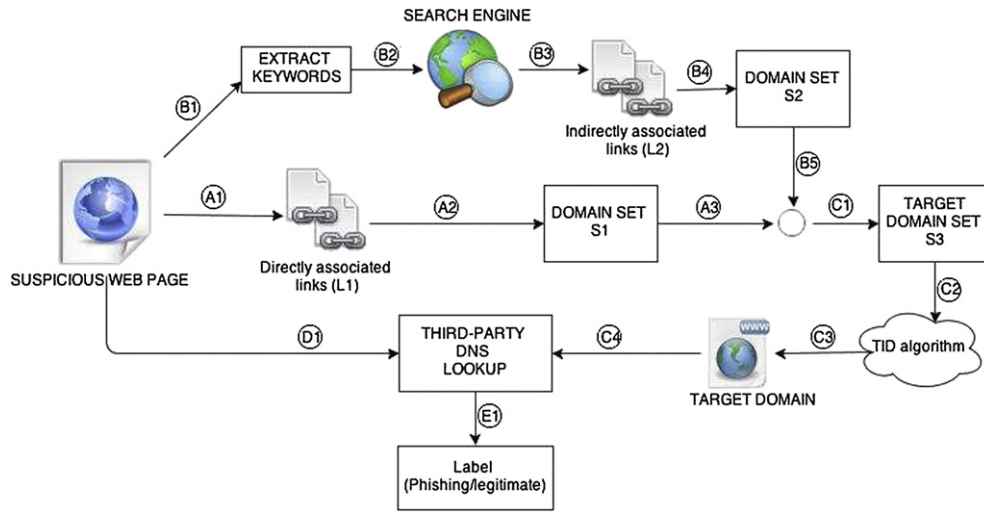


Fig. 1. System design (A1–A3: Extract links present in webpage; group links according to domains; domain set S1 given for set comparison; B1–B5: Extract keywords; keywords feed to search engine; extract the results; group links according to domains; domain set S2 given for set comparison; C1–C4: Identified target domain set; input target domain set to TID algorithm; identify the target domain; supply domain name of the target domain to third-party DNS server; D1: Supply domain name of the suspicious webpage to third-party DNS server; E1: Label generation based on DNS comparison (phishing = 0, legitimate = 1).

replace the relative links by their hierarchically associated known absolute link.

The links in indirectly associated link set (L2) are obtained from search engine results, by supplying keywords extracted as given in Section 4.1. Search engines take keywords of a web document as a query and crawls the web to identify the webpages which closely match the description given. When a suspicious page is phished, no matter what keywords we add on the query string, search engines will not return the phished page in top ranked results [10]. Therefore, we have considered this as the most reliable way of identifying a genuine webpage. The links extracted from the search engine result (L2) are used to narrow down the domains in set L1 for the target domain identification which is explained in the following sections. Here, as it would be impractical to consider all the links returned by the search engine, we take only a finite number of these links. In our system, we have chosen the first ten links as the associated links because we have observed through experiments that the target domain always appears among the highest ranked.

4.3. Identifying the target domain set

The links in set L1 are grouped by domains which results in domain set S1. Similarly we find the domain set S2 from the set L2. The target domain set S3 is constructed by considering common domains present in both S1 and S2. This intersection operation narrows down the possible target domains by eliminating irrelevant domains linked from suspicious page. Let us consider the case of www.abc.com/index.html being a suspicious webpage whose target webpage is www.xyz.com. But not all the links from www.abc.com/index.html are directed to www.xyz.com, few of the links might be directed to some irrelevant domains. This reduced domain set S3 will be given as an input to TID algorithm, which analyzes all domains in the target domain set and predicts the target domain.

5. Identifying the target domain

In this section we have discussed about how the target domain is identified from the target domain set (S3) also we check the authenticity of the suspicious webpage. The set S3 contains the predicted target domains and depending on the number of domains in it two scenarios are possible. In the following subsections we would discuss the target domain discovery in each of these scenarios and the corresponding algorithm is explained in Fig. 2.

5.1. The intersecting set S3 is a nonempty set

For each of the domains in the target domain set, we calculate the cost it takes to reach the domain from the current suspicious webpage P. The cost of reaching a domain from the webpage is the number of links that are directly and indirectly associated with the webpage. Consider a domain 'i' in the target domain set, let the number of directly associated links be D_i and the number of indirectly associated links be G_i , then the cost, X_i is calculated as Eq. (1).

$$X_i = D_i + G_i \quad (1)$$

Let us consider the number of domains in the target domain set to be n . The set L3 contains all the links from P to the domain set S3. Taking one domain at a time in S3, we find links in set L3 which points to that domain. And by visiting each of those pages, we sort the links in the webpage according to the domains in the target domain set S3, and we maintain the count of number of links, which are pointing from current visiting domain to domains in the S3. We represent this count as N_{ij} where i is a visiting domain and j is a domain in set S3.

In finding phishing target we use the count for each domain in S3. To this count N_{ij} (number of links from domain i to each domain in S3), we also multiply the corresponding weight of domain j in order to minimize the effect of heavily populated links from domains i to j . While considering the count N_{ij} alone, there is a possibility that a skew is created. In other words, we mean to say that there can be a situation where the number of links pointing from domain i to off-target domain j is very large in comparison to the number of links pointing from domain i to a target domain in S3, which may lead to erroneous results in finding the phishing target. To overcome this drawback, we introduce the concept of weights. For every domain i in the target domain set, we find its corresponding weight. The weight of a domain is calculated by taking the membership of the domain in the set L3. For example, let us consider that there are three domains in the target domain set S3 and 44 links in L3. Out of 44 links in L3, let 10 links point to a domain-1, 31 links point to domain-2 and 3 links point to domain-3. Then, the membership of domain-1, domain-2 and domain-3 would be 0.227, 0.705 and 0.068 respectively. For this calculation, we have considered both crawlable links and non-crawlable links (links that point to image, external objects, etc.) of the webpages. Mathematically, this is expressed as shown in Eq. (2).

$$W_i = \frac{X_i}{\sum_{i=1}^n X_i} \quad (2)$$

Input: Target Domain set (S3)

Output: Target domain

1. If S3 is nonempty set
 - (a) Calculate the cost of reaching a domain from webpage.
 - (b) Calculate the cost of reaching from one domain to other domain, for all domains in S3.
 - (c) Construct the cost matrix.
 - (d) Identify the target domain.
2. If S3 is a null set
 - (a) If S1 and S2 are nonempty sets
 - i. When the ratio of active to inactive links in the suspicious page less than 1, consider S2 as target domain set S3 and go to step 1.
 - ii. Otherwise, consider target domain set S3 as union of S1 and S2 and go to step 1.
 - (b) If S1 is a null set and S2 is nonempty set
 - i. Consider S2 as target domain set S3 and go to step 1.
 - (c) S2 is null set
 - i. Declare as phishing page, phishing target is the domain having highest number of occurrences in set L1.

Fig. 2. TID algorithm.

Finally, to identify the target domain, we perform the following steps. In order to find the target domain, we need to calculate the cost of reaching from one domain to other domains in the target domain set. It is calculated as Eq. (3).

$$X_{i,j} = N_{i,j} * W_j \quad (3)$$

where i and j are the domains in the target domain set. The corresponding graph is shown in Fig. 3.

We need also to construct a square matrix that maps the cost taken to reach a domain from every single domain including itself. The domains in this matrix include not just the target domains but the domain of P as well. We calculate the sum of each column, which indicates the cost of reaching the domain from all domains in S3 as shown in Fig. 4. The domain corresponding to the column with the highest sum is our target domain.

5.2. The intersecting set S3 is a null set

This case occurs when none of the elements in sets S1 and S2 matches or either of the sets S1 or S2 is a null set. Set S1 would be a null set when we couldn't extract links from webpage. Set S2 would be a null set when

we could not extract keywords from the webpage. This happens in situations when a suspicious webpage is only made up of any combinations like images, scripts or encoded content. In the following subsections we explain phishing target detection when S3 is a null set.

5.2.1. S1 and S2 are nonempty sets

Here, we check the ratio of active to inactive links in the suspicious webpage to narrow down the possible target domains. If the ratio is less than 1, it shows that the attacker is trying to manipulate the links in a webpage to bypass or mislead the phishing detection via fake links to a non-existing webpage. In this case we treat the set S2 as the target domain set S3 and proceed to construct the cost matrix as discussed in Section 5.1.

In the case of a legitimate webpage most of the links are active and the ratio of active links to inactive links is always greater than 1. Here, we consider the case where all links of P are directed to its own domain or the case where webpage is having a sizable number of links pointing to the external domains. In the abovementioned cases we take every domain that is present in S1 and S2 to be in S3 and proceed further using the method discussed in Section 5.1 to construct the cost matrix. Here,

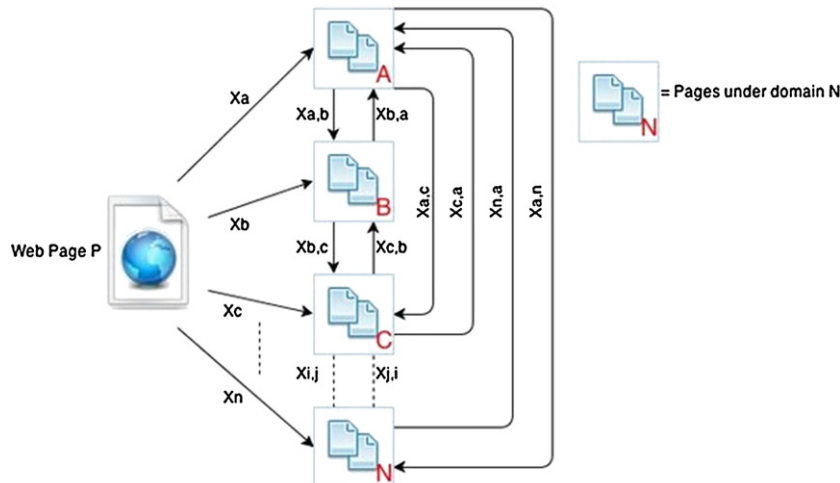


Fig. 3. Graph indicating the target domain set and link costs.

	P	A	B	..i..	N
P	0	X_a	X_b	$..X_i..$	X_n
A	0	$X_{a,a}$	$X_{a,b}$	$..X_{a,i}..$	$X_{a,n}$
B	0	$X_{b,a}$	$X_{b,b}$	$..X_{b,i}..$	$X_{b,n}$
..i..	0	$X_{i,a}$	$X_{i,b}$	$..X_{i,i}..$	$X_{i,n}$
N	0	$X_{n,a}$	$X_{n,b}$	$..X_{n,i}..$	$X_{n,n}$

$$\sum_{j=p}^N X_{j,a} \quad \sum_{j=p}^N X_{j,b} \quad \sum_{j=p}^N X_{j,i} \quad \sum_{j=p}^N X_{j,n}$$

Fig. 4. Cost matrix for target identification.

we cannot identify the target by the column sum of the matrix alone, since the sets S1 and S2 are distinct. To reduce this uncertainty in the target detection, we go for post processing.

In the post processing, we consider each of domains corresponding to the column in the matrix starting from the domain having the highest cost. The domain name is then compared with the keywords extracted from the suspicious page. If matched, we declare it as a target domain since the URL address of most of the genuine websites is related to its content. This characteristic of the URL address clearly shows that there is a relationship between base domain of the webpage and the keywords extracted from the webpage. For example, the co-operative bank's URL is <https://personal.co-operativebank.co.uk>. The keyword identity set for this webpage is {co-operative, internet, digit, bank, visa} where co-operative is part of the base domain.

5.2.2. S1 is a null set and S2 is a nonempty set

In this case, we treat set S2 as the target domain set S3 and proceed further using the method discussed in Section 5.1.

5.2.3. S2 is a null set

This case occurs when we cannot extract keywords from a suspicious webpage and when the attacker explicitly tries to hide page identities from the anti-phishing tools. In this case, we identify the target domain by taking the foreign domain which is having highest number of occurrences in set L1; also we conclude the page as phishing. In the legitimate webpages we can extract keywords at least from the title and meta tags present in page source.

6. Phishing detection using DNS lookup

In the previous sections we have identified the target domain of the suspicious webpage. Here we take the target domain and the domain of the suspicious webpage P, and perform third-party DNS lookup. As a result we get the corresponding IP addresses for both the domains. On comparing these two sets of IP addresses we draw a conclusion on the legitimacy of P. If the IP addresses of the domain P are matched with those retrieved for the target domain we declare P to be a legitimate webpage. Otherwise, we conclude it to be a phishing webpage. We have used third-party DNS lookup to avoid pharming attack (The user is redirected to a phished page even though he enters a correct URL. Attackers carry out this by exploiting the vulnerability in DNS server software).

In identifying the legitimacy of a webpage we have used IP address comparison instead of domain names, to overcome the discrepancies in domain names. For example we consider a case where, www.gmail.com may be having different aliases like mail.google.com, googlemail.l.google.com; here domain comparison leads to false positive.

7. Implementation and evaluation

7.1. Implementation

Our anti-phishing system is implemented in Java platform standard edition 7. It takes URL of the suspicious webpage as input and evaluates its legitimacy. The directly associated links (L1) are extracted from the

webpage using Jsoup HTML parser [24]. It provides a convenient way to access links from HTML of the page. Along with Jsoup, we have also used pattern matching which helps in extracting links from webpages that are not well formed.

The indirectly associated links (L2) of a webpage are extracted in three step process; 1) keywords are extracted from a suspicious webpage using the tf-idf method. Here, we retrieve up to 7 terms from resultant keyword list whose tf-idf values are ranked at the top. These keywords are used to frame the search query. 2) The search query is given to search engine to retrieve the top 10 search results, for which we have used Google Custom Search API [25]. 3) The links in set L2 are extracted by parsing Google's search result.

The links in sets L1 and L2 are grouped by domain name which results in sets S1 and S2 respectively. We have used guava-libraries [26] to extract parent level domain from each of the links. The target domain set S3 is constructed by performing intersection operation between sets S1 and S2, which is fed as an input to TID algorithm to identify the target domain. Finally, IP address of the target domain is compared with IP of suspicious webpage using Google Public DNS (8.8.8.8 and 8.8.4.4) to verify its genuineness. For the runtime analysis we have used hrtlib.jar [27] timing library. This library uses Java Native Interface (JNI) implementations to return even a submillisecond timing spent by our system.

7.2. Metrics used in evaluation

We have used three metrics to evaluate the performance of system, which are true positive rate (TPR), false positive rate (FPR) and accuracy (ACC).

The true positive rate measures the percentage of correctly classified phishing sites. The TPR is computed using Eq. (4).

$$TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)} \quad (4)$$

where TP is the number of correctly classified phishing pages. P is the number of phishing pages, which is equivalent to the sum of correctly classified phishes (TP) and missed phishes (FN).

The false positive rate measures the percentage of legitimate sites wrongly classified as phishing. The FPR is computed using Eq. (5).

$$FPR = \frac{FP}{L} = \frac{FP}{(FP + TN)} \quad (5)$$

Here FP is the number of legitimate pages which are wrongly classified as phishing, L is the number of legitimate pages which is equivalent to the sum of falsely classified legitimate pages (FP) and correctly classified legitimate pages (TN).

The accuracy measures the degree of closeness between measurements of classified sites and sum of actual phishing sites and legitimate sites. The ACC is computed using Eq. (6).

$$ACC = \frac{(TP + TN)}{(P + L)} \quad (6)$$

Here accuracy value will be close to 1 for any ideal anti-phishing system. Accuracy of the system can be improved by having higher TP value and lower FP value.

7.3. Description of data

We have collected a real world dataset of 4574 live phishing and legitimate websites over a period of 3 months from November 2012 to January 2013. Specifically, our dataset consists of 1200 legitimate pages and 3374 unique phishing pages.

Legitimate pages in our dataset are obtained from three sources as shown in Table 2. Legitimate pages in our dataset mainly focus on the popular and most targeted websites published in the sources.

Each entry in our phishing dataset is unique which are downloaded from two sources as shown in Table 3.

7.4. Detection accuracy

The experiment results are shown in Table 4. The true positive rate of this method is 99.67%, false positive rate is 0.5% and accuracy is 99.62% as shown Fig. 5. This statistics clearly shows that this system detects phishes with less false positives and high accuracy rate. Moreover, for all the successfully classified pages we have identified its target also.

The key reason for false positive is the absence of the target domain in the target domain set S3. This occurs because of two reasons; (1) when a legitimate webpage's domain is not listed in the top 10 search engine results and (2) when we could not extract keywords from a page. Similarly the false negative occurs when a phishing page is hosted on the compromised domain.

The test results of our method are compared to those of CANTINA [10], CANTINA + [11], and Wenyin's methods [20,21]. As Table 5 shows our method is more advantageous over other methods as our method has a low false positive rate and higher accuracy. The results of other methods (except Wenyin's method [21]) are collected from the respective papers and the testing dataset for each method is different.

The accuracy of phishing target detection of our method is 99.85% which shows that our target detection method has more merits than other methods that have been discussed in the related work section. Earlier target detection approaches most of the times identify cluster of target pages for a suspicious webpage and leaves the decision to the user to select right target page. But, our approach eliminates this problem by detecting a target domain instead of target pages.

In Appendix-A, we have included the screenshot of our system output and phishtank's (www.phishtank.com) user review on a webpage. In Appendix-B, we have shown a brief comparison of our system's output with sitewatcher [21] results and phishtank's user reviews.

7.5. Runtime analysis

All our experiments were carried out on a computer with a 2.4 GHz processor and 4 GB RAM. We have observed that our method takes average run time of 20,806 ms \pm 27,272 ms to decide the legitimacy and identify target of the webpage. Table 6 shows the average runtime of four modules. In this, average runtime of domain set S1 generation includes time taken for the extraction of links from HTML DOM of the suspicious webpage and time taken for grouping it by domain. Similarly, average runtime of domain set S2 generation includes time taken for extraction of keywords from suspicious webpage, querying the search engine, parsing the search result and grouping it by domain. Among these four modules the TID algorithm has a wide standard deviation in the runtime. This is because, for some of the webpages the set S3 is a null, but S1 and S2 are not null sets. In this case, TID algorithm constructs cost matrix for every domain in sets S1 and S2 for the target detection which leads to increase in computation time.

Table 2
Legitimate data source.

Source	Sites	Link
Google's top 1000 most-visited sites	675	http://www.google.com/adplanner/static/top1000/
Alexa's top sites	340	http://www.alexa.com/topsites
Netcraft's most visited sites	135	http://toolbar.netcraft.com/stats/topsites/
Millersmiles' top targeted sites	50	http://www.millersmiles.co.uk

Table 3
Phishing data source.

Source	Sites	Link
Phishtank's open database	2589	http://www.phishtank.com/developer_info.php
Reasonable-phishing webpage list	785	http://antiphishing.reasonables.com/BlackList.aspx

Table 4

Experiment results: N is the total number of pages, n is the number of correctly classified pages.

	Phishing pages	Legitimate pages	Total
N	3374	1200	4574
n	3363	1194	4557

7.6. Detection accuracy of system with different search engines

7.6.1. Metrics used in search engine evaluation

In this section we have evaluated the effectiveness of various search engines in order to find the right one that helps to enhance the accuracy of our anti-phishing system. As discussed in Section 4.2 the indirectly associated link set (L2) is constructed from search engine results by supplying keywords obtained from suspicious webpage. These links in set L2 are further grouped by domains resulting in set S2. The scope of detecting right target domain increases when the number of entries in S2 is less, also it significantly reduces the process of constructing cost matrix for detecting target domain. Thus in our method, the search engine that returns more pages (in top ten ranks) from same domain is considered the most effective.

Here, we have used discounted cumulative gain (DCG) based evaluation [23] to measure the effectiveness of search engines. DCG examines the ranked list of search result for a given query based on two assumptions. 1) Highly relevant documents are more valuable when they attain the high ranking positions in the search engine result list. 2) Highly relevant documents are less useful when they appear in the lesser ranked positions of search engine result list, as it gains less focus of the user.

Each search engine's average performance is evaluated by four step process. In the first step, the relevance score G' (gain vector) is constructed by examining each document from ranked positions 1 to 10 in the search result of a query. In gain vector, if the base domain of the search result is a target domain it is represented as 1 otherwise 0. For example, $G' = \langle 1, 1, 1, 1, 0, 0, 1, 0, 1, 0 \rangle$.

In the second step, the discounted cumulative gain accumulated at a particular rank position i is calculated for G' . This discounting function is required to gradually reduce the document's relevance score as its rank increases.

$$DCG[i] = \begin{cases} G[i] & \text{if } i=b \\ DCG[i-1] + \frac{G[i]}{\log_b i} & \text{if } i \geq b \end{cases} \quad (7)$$

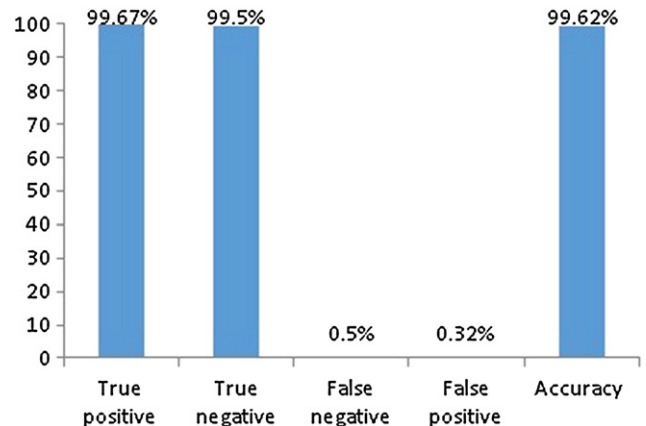


Fig. 5. Assessment of experiment.

Table 5
Comparison among anti-phishing methods.

Anti-phishing methods	No. of legitimate pages (L)	No. of phishing pages (P)	Total pages in dataset (L + P)	TPR	FPR	Accuracy
CANTINA [10]	100	100	200	89%	1%	94%
CANTINA+ [11]	4780	8118	12,898	99.63%	0.4%	99.6%
Semantic link network method [20]	1000	1000	2000	83.4%	13.8%	84.8%
Antiphishing through phishing target discovery [21]	1200	2000	3200	93.25%	4.16%	98.73%
Our method	1200	3374	4574	99.67%	0.5%	99.62%

Table 6
Average runtime and standard deviation of four modules.

Modules	Average runtime (milliseconds)
Domain set S1 generation	3192 ± 3443
Domain set S2 generation	2290 ± 894
TID algorithm	13,123 ± 24,399
DNS lookup and IP comparison	670 ± 703

where the base value b is considered as 2 and the $G[i]$ denotes i th position in the gain vector G' .

In the third step DCG vectors are normalized. As, comparing a search engine's performance from one search result to the next cannot be consistently achieved using DCG alone, the cumulative gain at each position for a chosen value of i should be normalized across search queries. The normalized DCG vectors (nDCG) are obtained by dividing DCG by the corresponding ideal DCG vectors as shown in Eq. (8). In our system the ideal vector I' is selected as all 1s since, first n positions are expected to get pages from same target domain, that is $I' = \langle 1, 1, 1, 1, 1, 1, 1, 1, 1 \rangle$. The ideal DCG I' vector is obtained by applying Eq. (7) on the selected ideal gain vector I' .

We obtain following DCG I' by applying Eq. (7) on ideal gain vector I' .

$$DCG'_I = \langle 1, 2, 2.63, 3.13, 3.56, 3.95, 4.30, 4.64, 4.95, 5.25 \rangle$$

Normalized discounted cumulative gain is computed as:

$$nDCG'_I = \langle \frac{DCG'_1}{DCG'_I1}, \frac{DCG'_2}{DCG'_I2}, \dots, \frac{DCG'_n}{DCG'_In} \rangle. \quad (8)$$

Finally, the average of nDCG up to position K is calculated as shown in Eq. (9).

$$Avg-pos(nDCG, K) = K - 1 * \sum_{i=1..K} nDCG[i] \quad (9)$$

The average value 1 shows that ideal result has been returned by the search engine. Here, the average value indicates the accuracy of current search for a given query.

7.6.2. Detection accuracy

Using the aforementioned method, we have assessed the performance of five different search engines and phishing detection accuracy of our system, the test results are as shown in Table 7. This experiment was conducted with 2000 URLs randomly selected from dataset stated in Section 7.3. The experiment results show that, with 76% of average performance google.com performs better by returning more number

of documents from the same target domain. Though, with google.com we could successfully detect the legitimacy of 1993 webpages, there is no significant difference in the system's prediction accuracy even when using other search engines. For example, with excite.com also our system could successfully classify 1989 webpages. This is achieved because on average excite.com returns 4 documents that are from target domain in the search result which gives room to have multiple domains other than the target domain in set S2. But these non-target domains in set S2 are removed or minimized when we compare it with set S1 (explained in Section 4.3). Thus, this operation helps our system to sustain the accuracy with any search engine.

7.7. Limitations of our approach

In this approach we cannot detect phishing webpages hosted on the compromised domains. This is because, the set S1 contains the phishing target. When intersecting it with the domain set S2 the target domain would be eliminated from the target domain set S3. This may result in the false prediction of target domain.

If we cannot extract links or keywords from the suspicious webpage, sets S1 and S2 will become null. As our approach requires either links or keywords of suspicious page to proceed in detection of target domain, the absence of both will lead to an undesirable situation which in turn results in false prediction.

In our method the accuracy of prediction depends on the keywords that are extracted from the suspicious page. Wrong keywords extracted will lead to irrelevant search results which may in turn lead to erroneous prediction and classification. This demands the extraction of effective keywords that uniquely identify the document.

Our method has the advantage of detecting phishing webpages of any language but, it requires an effective language independent keyword extraction method to find a right target domain. In the tf-idf method to retrieve document frequency value of a term, we use a ready-made frequency list compiled by <http://www.wordfrequency.info>, but the bottleneck is that the frequency list is available only for few languages. For other language websites which do not have a ready-made frequency list we extract keywords only from title and meta tags of it with size greater than or equal to two instead of depending on the tf-idf method. This may affect the accuracy of target detection.

Our approach uses search engine results and DNS lookup to identify target and phishiness of webpages. Therefore, the prediction time of our system depends upon the speed of the search engine and DNS lookup time. The delay caused by any of these external sources would increase the target prediction time proportionally. But with, today's high speed internet and availability of alternate sources this bottleneck problem is eliminated.

Table 7
Accuracy of different search engines on same dataset.

S. No.	Search engine	Average performance on our dataset(in %)	TPR (in %)	FPR (in %)	Accuracy (in %)
1	google.com	76.76	99.8	0.5	99.65
2	aol.com	71.1	99.7	0.5	99.6
3	hotbot.com	65.5	99.8	0.7	99.55
4	bing.com	59.48	99.7	0.8	99.45
5	excite.com	57.94	99.8	0.9	99.45

8. Conclusion

Our system identifies phishing websites along with its victimized domain that most of the anti-phishing methods lack. Also, our approach detects newest phishing websites hosted in any language. We have convincing results which show that our system has 99.62% of accuracy out of which 99.85% target domains were rightly

identified. This high detection accuracy is possible because of the methods we have adopted to narrow down the possible target domains from initial sets and cost matrix construction for target detection. Though we have got impressive runtime performance from our system, it can be further improved by devising a target identification strategy without typically depending on external information repositories in the web.

Appendix A

PhishTank is operated by [OpenDNS](#), a free service that makes your Internet safer, faster, and smarter. [Get started today!](#)



Signed in: [rameshgowtham](#) | [My Account](#) | [Sign Out](#)

[Home](#)
[Add A Phish](#)
[Verify A Phish](#)
[Phish Search](#)
[Stats](#)
[FAQ](#)
[Developers](#)
[Mailing Lists](#)
[My Account](#)

Submission #1932226 is currently offline

Submitted Jul 14th 2013 7:44 PM by [leofelix](#) (Current time: Jan 21st 2014 7:32 AM UTC)

<http://50.63.131.249/primarylogin.php>



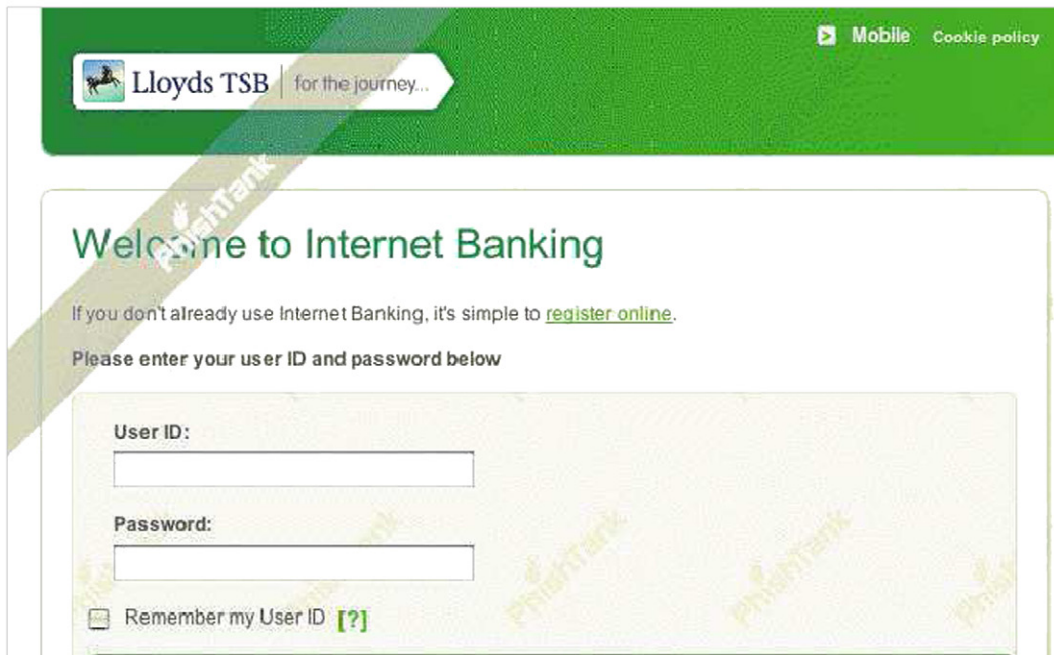
Verified: **Is a phish** [Next unverified phish >](#)

As verified by [NotBuyingIt](#) [alsf78](#) [codpiece](#) [knack](#)

Is a phish 100%

Is NOT a phish 0%

[Screenshot of site new window](#)
[View site in frame](#)
[View technical details](#)
[View site in](#)
[Something wrong with this submission?](#)



[Friends of PhishTank](#) | [Terms of Use](#) | [Privacy](#) | [Contact](#)

PhishTank is operated by [OpenDNS](#). Learn more about [PhishTank](#) or [OpenDNS](#).

Server: pt8.phishtank.com

Fig. 6. Phishtank — Details on suspected phish (1932226).

```

<terminated> CopyOfProject [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe
Please enter the url:
http://50.63.131.249/primarylogin.php
Extracting Links from the webpage...

The domain under DAL (S1) :
50.63.131.249 lloydstsb.com lloydstsb.co.uk
lloydsbankinggroup.com icra.org rsac.org

Extracting keywords and querying search Engine...
Keywords = lloyds tsb welcome internet
Language = en
The domain under IAL (S2) :
lloydstsb.com lloydstsb.co.uk lloydstsb-usa.com
lloydstsbbusiness.com lloydstsb-offshore.com yimg.com
lloydsbankinggroup.com
S3 is = [lloydstsb.com, lloydstsb.co.uk, lloydsbankinggroup.com]
size of S3 is = 3
Now calculating inward links and weightage ....

Number of links from webpage to domain 0 = 10
Number of links from webpage to domain 1 = 31
Number of links from webpage to domain 2 = 3
The total weight from webpage to S3 = 44
Weightage of domain 0 = 0.227
Weightage of domain 1 = 0.705
Weightage of domain 2 = 0.068
The number of crawlable links(L3) : 18
Crawling and finding the Target domain...

The target domain is : lloydstsb.com
50.63.131.249
[50.63.131.249]
lloydstsb.com
[141.92.130.226]
The Webpage is Phishing!!!
  
```

Fig. 7. System output.

References

- [1] Proof point: security, compliance and the cloud, <http://blog.proofpoint.com/2012/11/spear-phishing-attack-cause-of-massive-south-carolina-data-breach.html> November 27 2012 (Visited: June 2013).
- [2] RSA Anti-Fraud Command Center, RSA monthly online fraud report, <http://www.emc.com/collateral/fraud-report/online-rsa-fraud-report-012013.pdf> January 2013 (Visited: June 2013).
- [3] Xi Chen, Indranil Bose, Alvin Chung Man Leung, Chenhui Guo, Assessing the severity of phishing attacks: a hybrid data mining approach, *Decision Support Systems* 50 (4) (2011) 662–672.
- [4] Guang Xiang, Jason I. Hong, A hybrid phish detection approach by identity discovery and keywords retrieval, *Proceedings of the 18th International Conference on World Wide Web*, ACM, 2009, pp. 571–580.
- [5] Indranil Bose, Alvin Chung Man Leung, The impact of adoption of identity theft countermeasures on firm value, *Decision Support Systems* (2013) 753–763, <http://dx.doi.org/10.1016/j.dss.2013.03.001>.
- [6] Fujun Lai, Dahui Li, Chang-Tseh Hsieh, Fighting identity theft: the coping perspective, *Decision Support Systems* 52 (2) (2012) 353–363.
- [7] Weili Han, Ye Cao, Elisa Bertino, Jianming Yong, Using automated individual white-list to protect web digital identities, *Expert Systems with Applications* 39 (15) (2012) 11861–11869.
- [8] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, Minaxi Gupta, PhishNet: predictive blacklisting to detect phishing attacks, *INFOCOM*, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–5.
- [9] Jian Zhang, Phillip Porras, Johannes Ullrich, Highly predictive blacklisting, *Proc. of the 17th Conference on Security Symposium*, USENIX Association, Berkeley, CA, USA, 2008, pp. 107–122.
- [10] Yue Zhang, Jason I. Hong, Lorrie F. Cranor, CANTINA — a content-based approach to detecting phishing web sites, *Proc. of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, May 08–12 2007, pp. 639–648.
- [11] G. Xiang, J. Hong, C.P. Rose, L. Cranor, CANTINA+: a feature-rich machine learning framework for detecting phishing web sites, *ACM Transactions on Information and System Security* 14 (2) (September 2011) (Article 21, 28 pp.).
- [12] Prevost, Gustavo Gonzalez Granadillo, Maryline Laurent, Decisive heuristics to differentiate legitimate from phishing sites, *Proc. of Conference on Network and Information Systems Security (SAR-SSI)*, La Rochelle, France, May 2011, pp. 1–9.
- [13] Anthony Y. Fu, Liu Wenyin, Xiaotie Deng, Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD), *IEEE Transactions on Dependable and Secure Computing* 3 (4) (2006) 301–311.
- [14] E. Medvet, E. Kirda, C. Kruegel, Visual-similarity-based phishing detection, *IEEE International Conference on Security and Privacy in Communication Networks*, Istanbul, Turkey, IEEE Computer Society Press, September 2008.
- [15] K.T. Chen, J.Y. Chen, C.R. Huang, C.-S. Chen, Fighting phishing with discriminative keypoint features, *IEEE Internet Computing* 13 (3) (2009) 56–63.
- [16] T.C. Chen, S. Dick, J. Miller, Detecting visually similar web pages: application to phishing detection, *ACM Transactions on Internet Technology (TOIT)* 10 (2) (May 2010) (Article 5, 38 pp.).
- [17] Yogesh Joshi, Samir Saklikar, Debabrata Das, Subir Saha, PhishGuard: a browser plug-in for protection from phishing, *Internet Multimedia Services Architecture and Applications*, 2008. 2nd International Conference on IMSAA 2008, IEEE, 2008, pp. 1–6.
- [18] Chuan Yue, Haining Wang, BogusBiter: a transparent protection against phishing attacks, *ACM Transactions on Internet Technology (TOIT)* 10 (2) (2010) 6.
- [19] Hossain Shahriar, Mohammad Zulkernine, Trustworthiness testing of phishing websites: a behavior model-based approach, *Future Generation Computer Systems* 28 (8) (2012) 1258–1271.
- [20] Liu Wenyin, Ning Fang, Xiaojun Quan, Bite Qiu, Gang Liu, Discovering phishing target based on semantic link network, *Future Generation Computer Systems* 26 (3) (2010) 381–388.
- [21] Liu Wenyin, Gang Liu, Bite Qiu, Xiaojun Quan, Antiphishing through phishing target discovery, *IEEE Internet Computing* 16 (2) (2012) 52–61.

Appendix B

A brief comparison of our system output with phishtank (www.phishtank.com) user reviews and SiteWatcher [21] output by supplying random URLs that are selected from phishtank's (Valid phishes, Invalid and Unknown) URL databases.

URL	Language	Phishtank user review		Our output			Sitewatcher output	
		Is a phish	Is not a phish	Set S3	Prediction	Target domain	Prediction	Number of targets
http://serviciowebinternetbod.ekiwi.es/	EN	100	0	bodmillenium.com	Phishing	www.bodmillenium.com	Unknown	Unknown
http://kiwi6.com/file/	EN	43	57	kiwi6.com	Legitimate	www.kiwi6.com	Unknown	Unknown
http://umzuegweien.at/ubersiedlung-wien/	EN	50	50	umzuegweien.at	Legitimate	www.umzuegweien.at	Unknown	Unknown
http://aquiorestaurante.com	EN	100	0	paypal.co.uk	Phishing	www.paypal.co.uk	Unknown	Unknown
http://php-developers.co.za/templates	EN	100	0	paypal.com	Phishing	www.paypal.com	Phishing	15
http://specialneedsok.org/drupal/	FR	80	20	paypal.com	Phishing	www.paypal.com	Unknown	Unknown
http://mobile365.mk/	EN	40	60	mobile365.mk	Legitimate	www.mobile365.mk	Legitimate	1
http://www.ff-winners.com/wp-includes/SimplePie/Cache/index.htm	EN	100	0	llyodstsb.com llyodstsb.co.uk llyodsbankinggroup.com	Phishing	www.llyodstsb.co.uk	Unknown	Unknown
http://www.fizakikas.com/index1099.php	EN	100	0	fizakikas.com facebook.com	Phishing	www.facebook.com	Unknown	Unknown

- [22] Gerard Salton, Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.
- [23] Kalervo Jarvelin, Jaana Kekalainen, Cumulated gain-based evaluation of IR techniques, *ACM Transactions on Information Systems (TOIS)* 20 (4) (2002) 422–446.
- [24] Jsoup: Java HTML parser, <http://jsoup.org> (Visited: October 2013).
- [25] Google custom search APIs and tools, <https://developers.google.com/custom-search/> (Visited: October 2013).
- [26] Guava: Google core libraries for Java, <http://code.google.com/p/guava-libraries/> (Visited: September 2013).
- [27] Vladimir Roubtsov, My kingdom for a good timer—reach submillisecond timing precision in Java, <http://www.javaworld.com/javaqa/2003-01/01-qa-0110-timing.html> (Visited: September 2013).

Mr. R. Gowtham is an assistant professor in the Department of Computer Science & Engineering at Amrita University, India. He has obtained his B.E. degree from Periyar University and M.E. degree from Anna University. He is currently pursuing Ph.D. under Anna University. His research interests are in the areas of information security and data mining.

Dr. Ilango Krishnamurthi graduated from BITS, Pilani and then Iowa State University, USA both in the field of Computer Science & Engineering. He then earned his doctorate degree in Computer Science & Engineering from the Indian Institute of Technology, Chennai. He has been teaching Computer Science & Engineering since the year 1988. He spent 15 years at NIT, Trichirapalli in the capacities of Lecturer, Assistant Professor and Coordinator of the part time B.Tech programme. Since October 2005, he is with SKCET as professor and then HOD since July 2006. Since June 2008 he has been promoted as Dean, CSE Department. He has published several research papers in National, International journals and conferences. His current research interests are in the areas of semantic web and data mining. Dr. Ilango is currently guiding ten students towards their doctorate degrees.

K. Sampath Sree Kumar is currently pursuing his B.E. degree in Computer Science and Engineering at Amrita University, Coimbatore. His research interests include semantic web, Web security and Information retrieval.