Automatic Detection of Phishing Target from Phishing Webpage[*]

Gang Liu, Bite Qiu, Liu Wenyin
Department of Computer Science, City University of Hong Kong,
83 Tat Chee Ave., HKSAR, China
gangliu@student.cityu.edu.hk, biteqiu@cityu.edu.hk, csliuwy@cityu.edu.hk

Abstract

An approach to identification of the phishing target of a given (suspicious) webpage is proposed by clustering the webpage set consisting of its all associated webpages and the given webpage itself. We first find its associated webpages, and then explore their relationships to the given webpage as their features for clustering. Such relationships include link relationship, ranking relationship, text similarity, and webpage layout similarity relationship. A DBSCAN clustering method is employed to find if there is a cluster around the given webpage. If such cluster exists, we claim the given webpage is a phishing webpage and then find its phishing target (i.e., the legitimate webpage it is attacking) from this cluster. Otherwise, we identify it as a legitimate webpage. Our test dataset consists of 8745 phishing pages (targeting at 76 well-known websites) selected from PhishTank and preliminary experiments show that the approach can successfully identify 91.44% of their phishing targets. Another dataset of 1000 legitimate webpages is collected to test our method's false alarm rate, which is 3.40%.

1. Introduction

Phishing is a kind of online attack widely used by phishers to steal users' accounts and passwords, and other personal information for illegal appropriation. In recent years, phishing attacks have become more and more sophisticated and their volume expands dramatically. According to the Anti-phishing Working Group [1], 211,271 unique phishing websites were reported in the first half of 2009 and about 279 reputable brands were hijacked every month.

In order to direct users to fraudulent web sites and steal their money, phishing patterns evolve constantly by phishers. However, the essence of phishing has no change. Generally, most phishing webpages use links pointed to legitimate webpages and visually similar content to lure visitors to enter their sensitive information. In this sense, phishing webpages are not isolated from their targets but have strong relationships with them, which can be used as clues to find their targets.

In this paper, we propose an approach to automatic identification of the phishing target of a given webpage by clustering the webpage set consisting of all its associated webpages and the given webpage itself. The associated webpages are those which are pointed by forward links of the given webpage and webpages returned by a powerful search engine with certain representative keywords (e.g. brand, title and keywords of content) in the given webpage as queries. This approach first finds its associated webpages and then mines the features such as links relationship, ranking relationship, webpage text similarity and webpage layout similarity relationship between the given webpage and its associated webpages. Finally, a DBSCAN [2] clustering method is employed to find if there is a cluster around the given webpage. If such cluster is found, the given webpage is regarded as a phishing webpage and its target is identified as the legitimate webpage in the cluster which is closest to the given webpage. Otherwise, we identify it as a legitimate webpage.

2. Related Work

Existing anti-phishing methods can be mainly classified into the following categories.

[*] The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 117907].

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.1010

IEEE computer society

(1) User interface. Dhamija et al. [3] and Wu et al. [4] proposed methods that need webpage creators to follow certain rules to create webpages, either by adding dynamic skin to webpages or adding sensitive information location attributes to HTML code.

(2) Visual similarity approaches. Liu et al. [5] proposed a visual similarity based strategy for detection of phishing webpages. They employed an algorithm to compute visual similarity, including block similarity, layout similarity and overall style similarity between a suspicious webpage and a protected webpage. Fu et al. [6] used Earth Mover's Distance (EMD) to measure webpage visual similarity and calculate the signature distance between suspicious webpages and a protected webpage for phishing detection.

(3) Hybrid approaches. Zhang et al. [7] implemented a content-based approach to detecting phishing websites, based on the TF-IDF information retrieval algorithm. Five terms from the given webpage with the highest TF-IDF scores were calculated and fed into the Google to get top N search results. If the domain name of the given webpage did not fall into the search results, they considered it as a phishing website. Pan and Ding [8] proposed an approach which employed a SVM-based page classifier to analyze the consistency between the identity claimed by and those features of the given webpage to determine whether it is phishing or not. Xiang and Hong [9] proposed a phish detection method by discovering whether the identity of the given webpage itself is consistent with the identity shown before cyber users.

(4) SLN-based approach. Liu et al. [10] used the Semantic Link Networks (SLN) to automatically identify the phishing target of a given webpage. They first found the associated webpages of the given webpage and then constructed a SLN from all those webpages. A mechanism of reasoning on the SLN was exploited to identify whether it was phishing or not and discovered its target if it was phishing.

3. The Approach

Our method consists of the following steps.

3.1. Finding the associated webpage set

The associated webpages are mainly from two sources.

(1) *Directly associated pages.* They are the webpages that are directly linked by the given webpage *P*. They can be found by examining the HTML source of page *P* and extracting all hyperlinks in it.

(2) *Indirectly associated pages.* They are the pages which share the same or similar text/visual information with *P*. For example, we can mine such indirectly associated pages of *P* by searching the Web with the representative keywords found in *P* as queries. The representative keywords are those found in the title, words in Meta tag, keywords in the body tag and organization name of *P*.

3.2. Representing webpages in feature vectors

The strength values of the association relationships are defined and measured as follows.

Feature 1: link relationship

We use L_{ij} as the metrics for the link relationship from $page_i$ to $page_j$.

$$L_{ij} = \frac{NL_{ij}}{NL_i}, \tag{1}$$

where, NL_{ij} is the number of forward links from $page_i$ to any page on the website of $page_j$; NL_i is the total number of forward links from $page_i$.

Feature 2: ranking relationship

We define the ranking association relationship from $page_i$ to $page_j$ based on the rank of $page_j$ in the search result using the representative keywords of $page_i$ as the query. The strength (degree) of the ranking relationship is measured by the rank of the domain name of $page_j$ in the search results. R_{ij} is used as the metrics for ranking relationship from $page_i$ to $page_j$.

$$R_{ij} = \frac{N_r - (R_s - 1)}{N_r}, \tag{2}$$

where N_r is the total number of search results with which we are concerned and R_s is the rank of the domain name of $page_j$ in the results. If it cannot be found in the result, its rank value is set as *zero*.

Feature 3: text similarity relationship

We measure text similarity relationship TS_{ij} from $page_i$ to $page_j$ as follows.

$$TS_{ij} = \cos(x, y) = \frac{x \cdot y}{\|x\| \, \|y\|}, \tag{3}$$

where, x is the term vector extracted from $page_i$; y is the term vector extracted from $page_j$; $\|x\|$ is the length of vector x, $\|y\|$ is the length of vector y.

Feature 4: layout similarity relationship

We define the layout similarity LS_{ij} from $page_i$ to $page_j$ as the ratio of the weighted number of matched blocks to the total number of blocks in $page_j$. Two blocks are considered matched if they both exhibit high visual similarity and satisfy the same constraints with corresponding matched blocks.

$$LS_{ij} = \frac{\left|T_i(blocks) \cap T_j(blocks)\right|}{\left|T_j(blocks)\right|}, \qquad (4)$$

where, *T_i(blocks)* and *T_j(blocks)* denote the blocks set of *$page_i$* and *$page_j$* respectively; |*T_j(blocks)*| is the total number of blocks in *$page_j$*; |*T_i(blocks)* ∩ *T_j(block)*| is the number of the blocks they share.

We quantify those association relationships from the given webpage *P* to any page (*$page_i$*) in the associated webpage set and use them as features in vector $V_i=\{L_i, R_i, TS_i, LS_{ij}\}$. Since our intention is to check whether certain pages in the associated page set can form a cluster with *P*, we also add *P* into the associated webpage set before clustering. With *P* being identical to itself, we define $V_p=\{ 1,1,1,1 \}$.

3.3. Clustering the associated webpage set

In the associated webpage set, the page has stronger association relationship with *P* is closer to *P* in the data space. The data points are not uniformly distributed. The closer to *P*, the greater the data point's density is. Hence, we employ the well-known density-based DBSCAN algorithm. In addition, it can select any data point as the start point for clustering. We select the given webpage as the start point for our purpose. The concepts used in our approach are described as follows.

Eps: Maximum radius of the neighbourhood of the cluster.

MinPts: Minimum number of points in an Eps-neighbourhood of that point.

core point(CO): Point is in the interior of a density-based cluster.

border point: A border point is not a core point, but falls within the neighbourhood of a core point.

directly-density-reachable (DDR): If point *x* is *CO*, point *y* is in x's Eps-neighbourhood.

density-reachable: There exists a chain of *DDR* objects from point *x* to point y.

Based on the above concepts, we present our clustering-based method as follows:

1) Find the associated webpage set of *P* and also add *P* into the set.

2) Quantify the association relationships from *P* to each webpage in the set and represent each webpage in a feature vector.

3) Select the given webpage *P* as the start point and retrieve all points density-reachable from *P* with respect to *Eps* and *MinPts*.

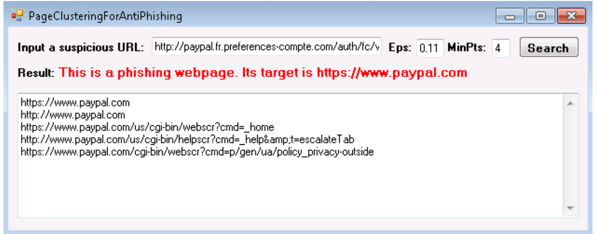
5) If *P* is a core point, a cluster is formed and the process stops.

6) If *P* is a border point and no point is density-reachable from *P*, no cluster is formed and the process stops.

If we can find a cluster including *P*, we claim that *P* is a phishing webpage and the page which has the strongest association relationship with *P* is its phishing target.

4. Experiments and Evaluation

We have implemented our method as a windows application accepting a URL as the given page. The user interface of the application is shown in Figure 1. *Eps* and *MinPts* are two parameters of the DBSCAN algorithms. If the given page is phishing, all the webpages it is potentially attacking are listed.



Input a suspicious URL: http://paypal.fr.preferences-compte.com/auth/fc/v Eps: 0.11 MinPts: 4

Result: This is a phishing webpage. Its target is https://www.paypal.com

```
https://www.paypal.com
http://www.paypal.com
https://www.paypal.com/us/cgi-bin/webscr?cmd=_home
http://www.paypal.com/us/cgi-bin/helpscr?cmd=_help&amp;t=escalateTab
https://www.paypal.com/cgi-bin/webscr?cmd=p/gen/ua/policy_privacy-outside
```

Figure 1. Interface of our windows application: Automatic Identification of Phishing Target

4.1. Datasets and experiment results

In our experiment, we selected 8745 phishing URLs targeting at 76 well-known companies/brands from PhishTank [11] to test the phishing target identification accuracy of our method. In our experiment, an identified phishing target is considered as correct for the given webpage if their domain name matches. Our method's accuracy rate of identification is 91.44%. Another testing dataset consists of 1000 legitimate pages which were randomly obtained from Random Yahoo Link [12]. The false alarm rate is 3.40%.

4.2. Empirical parameters setting

In this experiment, we tune the two parameters, *Eps* and *MinPts* to find their optimal values such that we can achieve the maximum average accuracy rate in detecting phishing targets and minimum average false alarm rate. Figure 2 and figure 3 depict the results using different *Eps* and *MinPts*.

As the results show, when the value of *MinPts* increases, bigger clusters are more likely to be labeled as noise and the target has less possibility to be identified. Hence the accuracy rate of phishing target detection deceases and the false alarm rate is also decreased. As for the parameter *Eps*, when it increases, it is more likely to form a cluster. Thus the accuracy rate of phishing target detection is improved and the false alarm rate should also be increased. After our observation on the real data from these two figures, we conclude that our method provides the best performance with *Eps* = 0.11 and *MinPts* = 4, where the accuracy rate is 91.44% and the false alarm rate is 3.40 %.

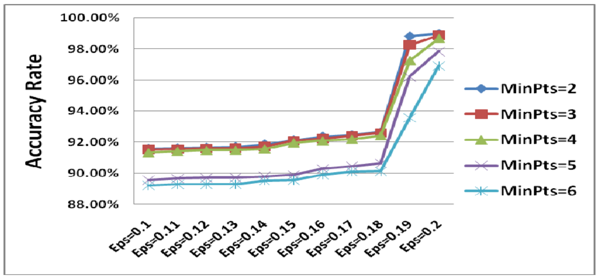


Figure 2. Accuracy rate with different *Eps* and *MinPts*

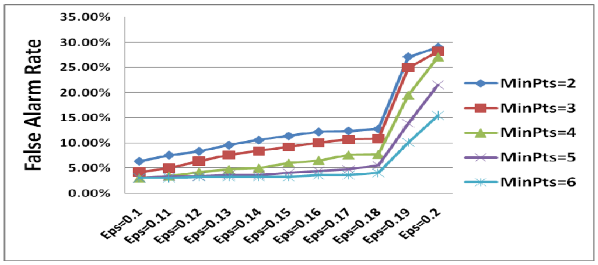


Figure 3. False alarm rate with different *Eps* and *MinPts*

4.3. Comparison with SLN-based method

We compare our method with a SLN-based method [10] in terms of performance. The detailed description of the comparison is as follows: (1) phishing target detection accuracy and (2) false alarm rate test.

Table 1. Performance comparison

Anti-phishing methods	Phishing target detection		False alarm rate test	
	Accuracy rate	# of Dataset	False alarm rate	# of Dataset
SLN method	83.40%	1000 pages	15.90%	1000 pages
Our method	91.44%	8745 pages	3.40%	1000 pages

Based on the comparison in Table 1, the accuracy of our method outperforms the SLN-based method by 8.04% for target detection while its false alarm rate is much lower than the SLN-based method.

5. Conclusion and Future Work

In this paper, we present a novel approach to identifying the potential phishing target of a given webpage. Our approach first mines its associated webpage set of the given webpage and systematically inspects those features of the link relation, ranking relation, similarity relation from the given webpage to its associated webpages. A DBSCAN clustering method is employed to find if there exists a cluster of webpages attacked by the given webpage potentially attacking. Experiments show that the accuracy rate is 91.44% and the false alarm rate is 3.40 %

In our future work, we plan to integrate this solution into web browsers as light-weight plug-ins to provide alerts for phishing attacks. We also plan to implement a class library (APIs) for enterprise users to build their own application system to check suspicious websites.

References

[1] Anti-Phishing Working Group, http://www.antiphishing.org/.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial database with noise. *Proc. KDD 1996*, pp. 226–231, 1996.

[3] R. Dhamija and J. D. Tygar. The battle against phishing: dynamic security skins. *Proc. SOUPS* 2005.

[4] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. *Proc. SOUPS*, 2006.

[5] W. Liu, X. T. Deng, G. L. Huang, and A. Y. Fu. An anti-phishing strategy based on visual similarity assessment. *IEEE Internet Computing* 10(2): 58-65, 2006.

[6] A. Y. Fu, W. Liu, X. T. 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):301-311, 2006.

[7] Y. Zhang, J. I. Hong and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. *Proc. WWW 2007*, pp. 639-648, 2007.

[8] Y. Pan and X. H. Ding. Anomaly based Web phishing page detection. *Proc. 22nd Annual Computer Security Applications Conference*, pp. 381-392, 2006.

[9] G. Xiang and J. I. Hong. A hybrid phish detection approach by identity discovery and keywords retrieval. *Proc. WWW 2009*, pp. 571-580, 2009.

[10] Liu W., N. Fang, X. J. Quan, B. T. Qiu and G. Liu. Discovering phishing target based on semantic link network, *Future Generation Computer Systems* 26(3):381-388, 2010.

[11] PhishTank, http://www.phishtank.com/.

[12] Random Yahoo Link, http://random.yahoo.com/bin/ryl.