Phishing Webpage Detection

Liu Wenyin[1], Guanglin Huang[1], Liu Xiaoyue[2], Xiaotie Deng[1] and Zhang Min[3]
[1]Dept. of Computer Science, [2]Dept. of Chinese, Translation & Linguistics
City University of Hong Kong, Tat Chee Avenue, Hong Kong, China
[3]State Key Lab of Intelligent Tech. & Sys., Tsinghua Univ., Beijing 100084, China
{csliuwy@, hwanggl@cs., xyliu0@, csdeng@}cityu.edu.hk, z-m@tsinghua.edu.cn

Abstract

An approach to detection of phishing webpages based on visual similarity is proposed, which can be utilized as a part of an enterprise solution to anti-phishing. A legitimate webpage owner can use this approach to search the Web for suspicious webpages which are visually similar to the true webpage. The approach first decomposes the webpages into salient (visually distinguishable) block regions. The visual similarity between two webpages is then evaluated in three metrics: block level similarity, layout similarity, and overall style similarity. A webpage is reported as a phishing suspect if any of them (with regards to the true one) is higher than its corresponding preset threshold. Preliminary experiments show that the approach can successfully detect those phishing webpages with few false alarms at a speed adequate for online application.

Keywords

Anti-Phishing, Web document analysis, Information filtering

1. Introduction

Phishing is a criminal trick of stealing victims' personal information by sending them spoofed emails urging them to visit a forged webpage that looks like a true one of a legitimate company and asks the recipients to enter personal information such as credit card number, password and etc. The victims may finally suffer losses of money or other kinds. According to the reports of Anti-Phishing Working Group [1], the number of phishing attacks is increasing by 50% monthly and they can usually convince 5% of the phishing email recipients to respond to them. By providing Internet transaction operations, it is the obligation of the companies to keep it safe. The companies may be expected to shoulder the responsibility, take the initiatives to go out to actively detect those phishing emails and phishing websites, and then prevent potential phishing attacks.

In this paper, we propose an approach to detection of phishing webpages based on visual similarity. An important feature of phishing webpages is that they look like the true ones in the aspects of the webpages and (optionally) their URLs. Otherwise, the victims would not believe them. Hence, a legitimate webpage owner can search for all suspicious URLs and compare the corresponding webpages with the true one in visual aspects. If the visual similarity of a webpage to the true webpage is higher than a threshold, the owner will be alerted and can then take whatever actions to immediately prevent potential phishing attacks.

In our approach, the visual similarity between two webpages is measured in three metrics: block level similarity, layout similarity, and overall style similarity, since a phishing webpage usually mimic the true one by using similar key regions/blocks, similar page layout, and similar styles (e.g., font family, size, decoration, and even spacing). All these three visual similarity metrics are defined based on webpage decomposition into a set of salient blocks [2].

In our experiments, we have collected 8 phishing webpages targeting at 6 true official webpages, together with other 320 normal webpages of commercial banks as the test dataset of suspicious webpages. We use each target webpage as a query and try to detect the phishing webpages targeting at it from the 320+8 webpages in the dataset. For each suspicious webpage, if its similarity to the query in any of the three aspects is higher than a specific threshold, a phishing case is reported. The experiment results show that our proposed approach can successfully detect the phishing webpages with few false alarms. We also test the pure-text features on this dataset but obtain a less clear boundary between the real phishing webpages and other non-phishing webpages.

The rest of this paper is organized as follows. Section 2 presents a brief survey of related works. Section 3 describes the system architecture of the proposed approach and some modules. Section 4 focuses on the method of visual similarity assessment, including the features and the details of the three visual similarity metrics. We present experiments and results in Section 5. Finally, we present concluding remarks and future works in Section 6.

2. Related Works

The problem of phishing webpage detection is in nature similar to the problem of duplicate document detection and plagiarism detection as well. Both problems are to detect or find similar documents or subdocuments. However, duplicate document detection focuses on text-based

1520-5263/05 $20.00 © 2005 IEEE

features in similarity measurement while phishing webpage detection should focus more on visual aspects of the documents. An important feature of the phishing webpages is that they use similar or exact visual effect to cheat the victims' eyes, make a phishing webpage look like the true one, and make the victims believe it is the true one.

The problem of detection of duplicate or near-duplicate documents is not new. Many approaches have been proposed, especially after the World Wide Web was invented. The Web is a tremendous but chaotic (non-structured) source (or database) of information. Many duplicate or near-duplicate versions exist for the same documents. It is necessary to detect and remove these duplications for efficient storage, indexing, and user satisfaction. Chowdhury [3] has presented a very good overview of the approaches to this problem. One of existing approaches is to create a unique hash value for each document. The hash value calculated by certain hash function can be used as the signature or fingerprint of that document. However, it is not so robust since slight changes in the document, even formatting changes, may result in a significantly different signature. Another group of approaches calculate the similarity of a pair of documents based on certain text features in order to reduce the complexity of document comparisons. The similarity model and features used can be different. Two well-known similarity models are resemblance [6] and cosine [7]. The text features are defined at various lexical unit (or granularity) levels, including general terms (word), shingles, and super shingles. These text features are further selected or weighted for efficiency reason. This is actually the IDF (Inverse Document Frequency) idea. The idea can also be applied to each lexical level, e.g., terms [4]. Simple sampling is also tried on shingles but yields poorer accuracy [6]. Usually, higher level lexical units (e.g., super shingle) do not work well for short documents [6], probably because the features can be used or selected are not rich. Hold and Zobel [5] extend the above similarity measures by developing a new identity measure based on the intuition that similar documents should contain similar numbers of occurrences of words.

All these duplicate data detection efforts are mainly for text documents, or at least, they use only text features for similarity measurement. They are not strong enough for the phishing webpage detection problem, in which the webpages are mainly similar in visual aspects and usually do not contain rich text.

3. The Approach

Figure 1 illustrates the system architecture of our approach, which mainly consists of five modules: True Webpage Processing, Suspicious URL Detection/Generation, Suspicious Webpage Processing, Visual Similarity Assessment, and Phishing Report. The true webpage is processed by the True Webpage Processing Module to obtain an intermediate representation. Given the true webpage's URL, the Suspicious URL Detection/Generation Module detects/generates all suspicious URLs. For each webpage at a suspicious URL, the Suspicious Webpage Processing Module fetches the webpage at that suspicious URL if it is available and generates its representation (in terms of blocks and features). The Visual Similarity Assessment Module compares the true webpage and each suspicious webpage and calculates their visual similarity based on their intermediate representations. If the visual similarity between a suspicious webpage and the true one exceeds a threshold, the Phishing Report Module is called.

The main idea of the approach is to obtain the webpages' visual features and assess their visual similarity based on the features. Since the objective of the system is to find those phishing webpages which mimic a given true webpage, which are known in advance, it is possible to run the True Webpage Processing Module in an offline mode as pre-processing.

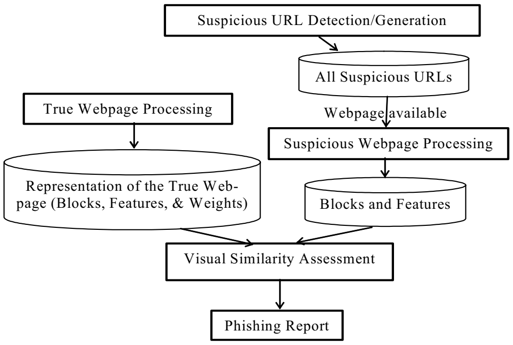
Suspicious URL Detection/Generation
All Suspicious URLs
True Webpage Processing
Webpage available
Suspicious Webpage Processing
Representation of the True Web-page (Blocks, Features, & Weights)
Blocks and Features
Visual Similarity Assessment
Phishing Report

Figure 1. System architecture of the proposed approach

The Suspicious URL Detection/Generation Module can be implemented by many methods. However, it is not our focus in this paper. Some heuristic rules can be applied to generate all suspicious URLs to a given URL by replacing some characters with other similar ones, e.g., 'o' with '0', and/or adding some prefix/suffix, e.g., "online" and "card". An alternative way is to monitor the email servers and analyze each message received. Since most phishing crimes are initiated by means of email, email is the most important source to detect potential phishing URLs. If a message contains a key word (e.g., the name of the customer company which requests this phishing-detection service), all the URLs embedded in the message are extracted as the output result of the Suspicious URL Generation Module.

The processing modules for the true webpage and a suspicious webpage are similar in the steps. Both contain the following two:

a) Page Segmentation
b) Feature Extraction

In our approach, the visual features of the webpages are obtained based on the result of webpage segmentation. In both the True Webpage Processing Module and the Suspicious Webpage Processing Module, the webpage is first decomposed into a set of salient blocks defined in [2], which are visually (in terms of visual features) and semantically (in terms of content relevancy) consistent within each block but distinguishable between adjacent blocks. We simply employ the method in [2] but do not group the salient blocks to higher levels.

After page segmentation, the visual features are extracted at the block level and then integrated into the page level. The details of feature definition and extraction will be presented in Section 4, where the Similarity Assessment Module is presented in detail. As we will see soon in Section 4, each block plays different roles and is weighted differently in the Visual Similarity Assessment Module. Key blocks are matched first prior to other normal blocks (see Section 4.1 and 4.2) and higher weight blocks contribute more to the webpage similarity than lower weight blocks. Hence, the true webpage should undergo the following additional step:

c) Key Block Labeling and Weighting
This step is used to assign the blocks with different roles and weights. It can also be done offline in advance, either automatically or manually. The automatic way can assign the weights of blocks using their areas or other heuristic rules. For the case of phishing, the blocks of company logo, user ID, and password are usually key blocks and should also be assigned with higher weights. A bigger area usually means a higher weight as well. In the manual way, the administrator of the system can designate several key blocks and assign them with some higher weights. The remaining blocks, however, can be weighted automatically, either uniformly or based on their areas.

Next, we discuss the Similarity Assessment Module in detail in Section 4.

4. Similarity Assessment

The Visual Similarity Assessment Module measures the visual similarity between two webpages in three aspects: block level similarity, layout similarity, and style similarity. All these three aspects are defined based on the blocks resulted from webpage segmentation. Next, we define these three similarity metrics in the following three subsections respectively.

4.1 Block Level Similarity

The block level similarity measures the visual similarity of two pages at the level of individual blocks. It is defined as the weighted average of the visual similarities of all matched block pairs between two pages. Basically, the content of a block can be categorized as either text or image. We use different features to represent text blocks and image blocks. The features for text blocks include text content, color, font, boundary, navigation, etc., and the features for image blocks include alt tag, block color, size, source, navigation, etc. The features for each block are extracted in the feature extraction step in the corresponding webpage processing module and form the feature set of the block.

If the two comparing blocks are of different types, their similarity is simply set to 0 in our current implementation. However, we are considering in the future that we can extract some textural information from the image block using an OCR module. In this case, the image block is converted to a text block and their similarity can be calculated using the text block similarity measurement method. Alternatively, the text block can be converted to an image block and the image block similarity measurement method can be applied accordingly.

If the two blocks are of the same type, we first calculate their similarity in terms of each feature in the feature set and then use a weighted sum of the individual feature similarities as the total similarity of the two blocks. The weight of each feature means its importance to the total similarity and can be assigned empirically. In out implementation, we focus more on color related features. The feature similarity of each different feature type is defined differently. If the possible values of a feature are enumerative or discrete (e.g., the font family), the similarity value for that feature is binary. If the possible values of a feature are relatively continuous (e.g., font size or color), the similarity value for that feature is simply defined as 1 minus the normalized difference of the feature values.

Two blocks are considered as matched if their similarity is higher than a threshold. After we obtain the similarity values of all pairs of possible matching blocks, we try to find a matching scheme between the two webpages' blocks. This is actually a bipartite graph matching problem, and a greedy-like matching algorithm is employed to find the matching blocks for the true webpage blocks in the descending order of their weights. The weight of each block in the true webpage is generated in the Key Block Labeling and Weighting step in the True Webpage Processing module which is discussed in Section 3. The key blocks are matched first. When trying to find a matching for a remaining true webpage block, its most similar suspicious webpage block in the remaining candidate list is selected as its matching block and then removed from the remaining candidate list.

The block level similarity of the true webpage and a suspicious webpage is defined as the weighted average of the visual similarities of all matched block pairs between the two webpages, as shown in Eq. (1).

$$Sim_B(P_t, P_f) = \sum_i w(B_i) Sim(B_i, B_{m(i)}), \qquad (1)$$

where, B_i is a block on the true webpage P_t , and $B_{m(i)}$ is the matching block on the suspicious webpage P_f for B_i, $w(B_i)$ is the weight of B_i. $Sim(B_i, B_{m(i)})$ denotes the block similarity between B_i and $B_{m(i)}$. $Sim_B(P_t, P_f)$ denotes the block level similarity between P_t and P_f.

4.2 Layout Similarity

Usually, it takes some efforts to make a brand new webpage mimicking a true webpage. A very convenient way is to copy the source file of the true one and modify it a little bit for this purpose. In this case, the main webpage structure is kept and the two webpages look very similar in their page layouts. Hence, we define the layout similarity as the ratio of the weighted number of matched blocks to the number of total blocks in the true webpage. We simply employ the method in [2], namely, the Neighborhood Relationship Model (NRM), for layout matching.

The method NRM in [2] is based on an assumption that there are several identical blocks between the two compared webpages for initial matching. However, this assumption is not always held in our situation. In our application, we use the matching method presented in 4.1 to find the matching blocks for the key blocks obtained in the Key Block Labeling and Weighting step in the True Webpage Processing Module (Section 3). These matched pairs are used as initial matching such that we can continue the other steps in the layout matching method in [2].

For layout similarity, we binarize the block similarity value of each true block (obtained in Section 4.1) to either 1 for matched or 0 for unmatched, that is,

$$Sim(B_i) = \begin{cases} 1 & B_i \ \ is \ \ matched \\ 0 & otherwise \end{cases}, \qquad (2)$$

We then define the layout similarity of two webpages as:

$$Sim_L(P_t, P_f) = \sum_i w(B_i) Sim(B_i), \qquad (3)$$

It can be considered as the ratio of the weighted number of matched blocks to the number of total blocks in the true webpage, and the weight of each block $w(B_i)$ is the same as in 4.1.

4.3 Overall Style Similarity

In addition to the webpage content, the style consistency is another important feature which can easily cheat the victims' eyes. Generally, all webpages owned by one company would keep the style consistent. Experience shows that a webpage viewer usually ignores the detail (textual or graphical) differences between the phishing webpage and the true one. If their styles are similar, it is usually hard for a common viewer to tell whether the webpage is from the authenticated company or not.

The overall style similarity focuses on the visual style of a webpage, which can be represented by several format definitions, e.g., the font family, background color, text alignment, line spacing, etc. These style features are used to define the overall style similarity metrics. We first obtain the histogram (distribution) of the style feature values for each webpage. For each discrete feature value (or bin) ϖ_{jk} for Feature j, we use the blocks and their weights as the unit to count the times, namely, the distribution value $\Gamma(\varpi_{jk})$ of that value/bin, as follows.

$$\Gamma(\varpi_{jk}) = \sum_{i, where \ \ f_j(B_i) = \varpi_{jk}} w(B_i), \qquad (4)$$

where, $f_i(B_i)$ means the value of Feature j in Block B_i. The weight of each block $w(B_i)$ is the same as in 4.1. The overall style similarity of two webpages is defined as the correlation coefficient (normalized to the range of [0,1]) of the two webpages' histograms.

5. Experiments

We have implemented our approach using the three similarity metrics defined in Section 4 for experiments. From the recent phishing attacks reported by the Anti-Phishing Working Group [1], we have collected 8 phishing webpages to evaluate our approach. Among them, three aimed at eBay and one at EarthLink, ICBC, Wells Fargo, US Bank, and Washington Mutual Bank, respectively. The six true target webpages are also collected for comparison. In the remaining part of this section, we denote a true webpage by adding the prefix "t-", e.g., t-eBay stands for the true webpage of eBay, and denote a phishing webpage by adding the prefix "f-", e.g., "f-ICBC" refers to the phishing webpage targeting at ICBC. The three phishing webpages targeting at eBay are denoted as "f-eBay1", "f-eBay2", and "f-eBay3", respectively.

Table 1. Block level similarities between test webpages.

	t-eBay	t-EarthLink	t-ICBC	t-Wells Fargo	t-US Bank	t-Washington
f-eBay1	**0.912**	0.680	0.569	0.683	***0.802***	0.675
f-eBay2	**0.968**	0.685	0.568	0.684	***0.795***	0.667
f-eBay3	**0.974**	0.688	0.539	0.703	0.672	0.669
f-EarthLink	0.613	**0.771**	0.586	0.628	0.583	0.615
f-ICBC	0.507	0.569	**0.929**	0.527	0.583	0.568
f-Wells Fargo	0.667	0.662	0.536	**0.982**	0.583	0.689
f-US Bank	***0.785***	0.649	0.538	0.645	**0.969**	0.633
f-Washington	0.676	0.695	0.578	0.645	0.640	**0.926**

Table 2. Overall style similarities between test webpages

	t-eBay	t-EarthLink	t-ICBC	t-Wells Fargo	t-US Bank	t-Washington
f-eBay1	**0.943**	0.503	0.423	0.573	0.545	0.487
f-eBay2	**0.951**	0.500	0.419	0.578	0.550	0.491
f-eBay3	**1**	0.460	0.496	***0.626***	0.607	0.528
f-EarthLink	0.462	**0.615**	0.326	0.528	0.481	0.483
f-ICBC	0.341	0.418	**0.994**	0.326	0.481	0.340
f-Wells Fargo	***0.617***	0.385	0.423	**0.775**	0.481	0.432
f-US Bank	0.514	0.397	0.415	0.519	**0.971**	0.675
f-Washington	0.436	0.326	0.315	0.519	***0.702***	**0.986**

Error! Reference source not found. shows the block level similarity values of the test webpages. The layout similarity values for all real pairs are 1.0 except for the "EarthLink" case, which is 0.445. The layout similarity values for other pairs are 0. **Error! Reference source not found.** shows the overall style similarity values. From these tables we can see that, for most cases the real pairs of phishing webpages and their targets result in significantly higher similarity values than other pairs. This indicates that our similarity assessment metrics are suitably defined and compatible with human eyes' visual perception.

Furthermore, to test our approach's ability to avoid false alarms, we have also collected a set of 320 normal webpages, which are index pages from the official websites of 320 commercial banks. These 320 normal webpages together with the 8 phishing pages are used to form the test dataset. All these test webpages can be downloaded at [8]. The 6 true pages are regarded as query to search for visually similar webpages in the test dataset. Once the similarity in term of any of the three metrics between a webpage and the query is larger than a threshold *t*, the system will report that this webpage is probably a phishing page. Actually, this is the way how our approach to phishing webpage detection works.

Table 3. Similarities between the true and phishing webpages.

Query	Reported page	Block level	Layout similarity	Overall style
t-eBay	f-eBay1	0.912	1	0.943
t-eBay	f-eBay2	0.968	1	0.951
t-eBay	f-eBay3	0.974	1	1
t-EarthLink	f-EarthLink	0.771	0.445	0.615
t-ICBC	f-ICBC	0.929	1	0.994
t-Wells Fargo	f-Wells Fargo	0.982	1	0.775
t-US Bank	f-US Bank	0.969	1	0.971
t-Washington…	f-Washington..	0.926	1	0.986

Table 3 list the similarity values of three metrics for 8 pairs of true and phishing webpages. Table 4 and Table 5 show the phishing detection result with the similarity threshold *t*=0.9 and *t*=0.7, respectively. We can see from the result that the false alarm rate is not high. We can also adjust the threshold further to guarantee no missing but with less false alarms.

Table 4. Phishing detection result *t*=0.9

Query	#Report Phishing	#Actual Phishing	#Missing	#False alarm
t-eBay	3	3	0	0
t-EarthLink	0	1	1	0
t-ICBC	1	1	0	0
t-Wells Fargo	1	1	0	0
t-US Bank	1	1	0	0
t-Washington Mutual	1	1	0	0

Table 5. Phishing detection result *t*=0.7

Query	#Report Phishing	#Actual Phishing	#Miss	#False alarm
t-eBay	4	3	0	1
t-EarthLink	1	1	0	0
t-ICBC	1	1	0	0
t-Wells Fargo	1	1	0	0
t-US Bank	4	1	0	3
t-Washington Mutual	1	1	0	0

We also record the time cost for similarity calculation for each pair of pages and the average is around 0.8s. We think the speed of our approach should be fast enough for online detection of the phishing webpages, since the bottleneck for such application is usually at the crawler, which takes time to download the webpage due to network latency.

6. Conclusions and Future Works

In this paper, we propose a novel approach to detect phishing webpages based on visual similarity. The approach first decomposes the webpages into salient blocks according to visual cues. The visual similarity between two webpages is then measured in three aspects: block level similarity, layout similarity, and overall style similarity. A webpage is reported as a phishing suspect if any of these similarities to the true webpage is higher than a threshold.

We have done experiments on a test dataset of 328 suspicious webpages. The 6 true webpages attacked by these 8 phishing webpages are used as query to retrieve those phishing webpages from the test dataset. Preliminary results show that our approach can successfully detect the phishing webpages with few false alarms. The speed is also fine for practical use.

We believe that the approach can be used as a part of an anti-phishing strategy in an enterprise solution. A true webpage owner can use this approach to detect phishing webpages. Furthermore, the application of the proposed approach is not limited to detection of this kind of phishing attacks. It can be used for detection of all maliciously forged webpages mimicking the webpages of any organization and person. We also believe that the overall style similarity metrics proposed in this paper can be applied to other applications. More experiments will be conducted to investigate these applications in the future.

7. Acknowledgement

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7001462).

8. References

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

[2] Liu Y., Liu W., and Jiang C. User interest detection on webpages for building personalized information agent. In Proc. of the 5th International Conference on Web-Age Information Management (WAIM 2004), Dalian, China. LNCS, Vol. 3129, pp. 280–287, 2004.

[3] Chowdhury A. Duplicate data detection (an overview). AOL. http://ir.iit.edu/~abdur/Research/Duplicate.html

[4] Chowdhury A., Frieder O., Grossman D., and McCabe. M. Collection statistics for fast duplicate document detection. ACM Trans. on Information Systems (TOIS) 20(2): 171–191, 2002.

[5] Hoad T.C. and Zobel J. Methods for identifying versioned and plagiarised documents. Journal of the American Society for Information Science 54(3): 203–215, 2003.

[6] Broder A., Glassman S., Manasse M., and Zweig G. Syntactic clustering of the web. Proc. of WWW97, pp.391–404.

[7] Salton G., Wong A., and Yang C.S. A vector space model for information retrieval. Journal of the American Society for Information Science 18(11), pp. 613–620, 1975.

[8] http://www.cs.cityu.edu.hk/~liuwy/phishing/testdata.zip